



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS  
AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

UNIDAD ZACATENCO  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA  
SECCIÓN COMUNICACIONES

**“Algoritmos para la evaluación del desempeño del  
protocolo TCP NewReno”**

T E S I S

Que presenta

M. en C. ROCÍO IDALIA ROMERO ÁNGELES

Para obtener el grado de

DOCTORA EN CIENCIAS  
EN LA ESPECIALIDAD DE INGENIERÍA ELÉCTRICA

Directores de la Tesis:

Dr. DOMINGO LARA RODRÍGUEZ

Dr. RAÚL GARCÍA RUIZ

México, D.F.

MAYO, 2015.



# Índice general

<b>Resumen</b>	<b>5</b>
<b>Abstract</b>	<b>7</b>
<b>Introducción</b>	<b>9</b>
<b>Objetivos</b>	<b>11</b>
<b>1. Protocolo de control de transporte, TCP.</b>	<b>13</b>
1.1. Operación de TCP. . . . .	13
1.2. TCP Tahoe. . . . .	19
1.3. TCP Reno. . . . .	21
1.4. TCP NewReno. . . . .	23
<b>2. Modelado del desempeño de TCP en redes cableadas</b>	<b>27</b>
2.1. Contexto histórico. . . . .	29
2.2. El modelo PFTK. . . . .	38
2.2.1. Suposiciones generales. . . . .	38
2.2.2. Suposiciones particulares del modelo. . . . .	40
2.2.3. Desarrollo del modelo: primera etapa. . . . .	41
2.2.4. Desarrollo del modelo: segunda etapa. . . . .	44
2.3. Modelos simplificados para TCP NewReno “lento pero firme” e “impaciente” desarrollados en [48], [49] y [50]. . . . .	47
2.3.1. Suposiciones particulares comunes en los modelos. . . . .	48
2.3.2. Suposiciones particulares del modelo presentado en [48]. . . . .	48
2.3.3. Suposiciones particulares de los modelos desarrollados en [49] y [50].	49
2.3.4. Desarrollo del modelo reportado en [48]. . . . .	49
2.3.5. Desarrollo de los modelos presentados en [49] y [50]. . . . .	53
2.3.6. Conclusión de los trabajos reportados en [48], [49] y [50]. . . . .	56
2.4. Modelo para TCP NewReno propuesto en [37]. . . . .	57
2.4.1. Suposiciones generales del modelo presentado en [37]. . . . .	57
2.4.2. Suposiciones particulares del modelo reportado en [37]. . . . .	57
2.4.3. Desarrollo de la primera etapa del modelo presentado en [37]. . . . .	58
2.4.4. Desarrollo de la segunda etapa del modelo desarrollado en [37]. . . . .	62
2.4.5. Validación del modelo reportado en [37]. . . . .	65

2.4.6.	Conclusiones del modelo reportado en [37]. . . . .	75
<b>3.</b>	<b>Algoritmo para la evaluación del desempeño en estado estable de una conexión TCP NewReno de larga duración.</b>	<b>77</b>
3.1.	Suposiciones sobre las que se desarrolla el algoritmo. . . . .	79
3.1.1.	Suposiciones generales. . . . .	79
3.1.2.	Suposiciones que se derivan del modelo presentado en [37]. . . . .	79
3.1.3.	Suposiciones particulares usadas para este algoritmo. . . . .	80
3.2.	Descripción del algoritmo. . . . .	80
3.2.1.	Fase CA. . . . .	82
3.2.2.	Fase TO. . . . .	85
3.2.3.	Fase SS. . . . .	87
3.2.4.	Fase FR. . . . .	91
3.3.	Modelo de pérdidas de la red. . . . .	95
3.3.1.	Modelo de Bernoulli. . . . .	95
3.3.2.	Modelo de Gilbert simplificado. . . . .	96
3.4.	Algoritmo. . . . .	98
3.5.	Validación del algoritmo. . . . .	101
3.5.1.	Módulo de pérdidas con distribución Bernoulli. . . . .	103
3.5.2.	Módulo de pérdidas con distribución Gilbert Simplificado. . . . .	103
3.6.	Conclusiones del algoritmo. . . . .	107
<b>4.</b>	<b>Algoritmo para la evaluación del desempeño de una conexión TCP NewReno.</b>	<b>111</b>
4.1.	Suposiciones sobre las que se desarrolla el algoritmo. . . . .	112
4.1.1.	Suposiciones generales. . . . .	112
4.1.2.	Suposiciones que se derivan del modelo en [37]. . . . .	112
4.1.3.	Suposiciones particulares que se derivan del algoritmo del capítulo 3. . . . .	112
4.1.4.	Suposiciones particulares usadas para este algoritmo. . . . .	112
4.2.	Descripción del algoritmo. . . . .	114
4.2.1.	Fase SS. . . . .	116
4.2.2.	Fase CA . . . . .	131
4.2.3.	Fase FR . . . . .	153
4.2.4.	Fase TO . . . . .	166
4.3.	Algoritmo . . . . .	176
4.4.	Validación del algoritmo. . . . .	182
4.5.	Conclusiones del algoritmo. . . . .	189
	<b>Conclusiones generales y trabajo a futuro.</b>	<b>193</b>
	<b>Glosario.</b>	<b>197</b>
	<b>Referencias.</b>	<b>199</b>

# Resumen

En la presente tesis doctoral se desarrolla una herramienta de evaluación del desempeño del protocolo TCP NewReno en redes cableadas, que incluye una representación completa y precisa de sus mecanismos y que es aplicable para una gran variedad de redes. Como primera propuesta de solución se desarrolla un algoritmo de evaluación de la tasa de transmisión en estado estable de una conexión TCP NewReno de larga duración. El algoritmo usa como parámetros de entrada el RTT promedio, la duración promedio del temporizador de retransmisión, la duración típica de una conexión TCP de larga duración y los parámetros del modelo de pérdidas de la red. Este algoritmo corresponde a una forma novedosa de evaluar el desempeño de TCP sobre una red cuyo proceso de pérdidas extremo a extremo pueda caracterizarse por un modelo analítico o por cualquier otro método que resulte en la secuencia de paquetes perdidos a lo largo de la transmisión. El algoritmo ofrece resultados más precisos que los obtenidos con los modelos analíticos existentes. Este incremento en la precisión se debe principalmente a dos factores: 1) la incorporación de una representación más cercana y detallada de los mecanismos del protocolo (en particular del Comienzo Lento y de la Evitación de la Congestión); 2) la posibilidad de usar un modelo de pérdidas adecuado para caracterizar distintos escenarios de operación en la red. Además, el tiempo de procesamiento que requiere y su complejidad son significativamente menores que los involucrados en una simulación. El algoritmo se valida a través de su comparación contra un conjunto de resultados de pruebas de simulación ns-2, obteniendo un error promedio del 1.4 % cuando el proceso de pérdidas en la red es modelado a través de un modelo de Bernoulli (pérdidas independientes), y de un 9 % cuando se usa el modelo de Gilbert (pérdidas correlacionadas).

Como propuesta de solución final se generaliza el algoritmo anterior para ser aplicable a flujos de datos de cualquier tamaño. El objetivo del segundo algoritmo es el de evaluar el valor esperado de la tasa de transmisión de una conexión TCP NewReno en una red cableada. La diferencia con el algoritmo anterior radica en los siguientes puntos: a) Usa el tamaño del flujo TCP como parámetro de entrada. b) Incluye una representación exacta de la operación del mecanismo de Recuperación Rápida de TCP NewReno. c) Usa una aproximación simple, pero adecuada, de la duración y de la cantidad de segmentos enviados durante el mecanismo de recuperación de pérdidas activado por la expiración del temporizador. d) Permite calcular las tasas de transmisión normal y efectiva de una conexión TCP NewReno particular de cierto tamaño. Los promedios de ambas medidas de desempeño se obtienen usando una muestra de ejecuciones del algoritmo. La validación de este algoritmo se realiza mediante su comparación contra un conjunto de resultados de simulación ns-2. Se consideran varios escenarios de pérdidas, todos ellos representados a través del modelo de Gilbert. El error promedio correspondiente al total de los casos presentados es del 2.15 %.



# Abstract

In this thesis, a new method to evaluate the throughput of a TCP NewReno connection is developed. The protocol mechanisms are completely and accurately represented in this method. In addition, it can be applied to any network whose end-to-end packet-loss process can be described by an analytical model or by any other method that results in the sequence of lost packets along the transmission. As a first proposal, this work presents the development of an algorithm to evaluate the steady-state throughput of a TCP NewReno bulk data flow as a function of the round-trip time, the retransmission timer value, the typical duration of a TCP NewReno connection, and the loss behavior of the network. This proposal achieves more accurate results than those from existing analytical models, with greater simplicity and less resource consumption than those necessary in a simulation. The improvement in accuracy accomplished with this algorithm, compared with the current analytical models, is due to the incorporation of a closer representation of the protocol mechanisms (specifically, the Slow-Start and Congestion Avoidance mechanisms), combined with its applicability to characterize the protocol operation on any network loss behavior. Such algorithm is validated by its comparison with a set of ns-2 simulation results, obtaining an average error of 1.4 % when the loss behavior of the network is modeled by a Bernoulli process (independent network losses) and of 9 % when the Gilbert loss model (correlated network losses) is used. Finally, the proposed solution consists of a generalization of the previous algorithm to be applicable to data flows of any size. The aim of this algorithm is to evaluate the expected value of the throughput of a TCP NewReno connection. This proposal differs from the previous one in several ways: 1) It uses the size data flow as an input parameter. 2) It includes an accurate representation of the TCP NewReno Fast Recovery mechanism. 3) It includes a simple and accurate approximation to the duration and the number of sent segments along the loss recovery mechanism triggered by a timeout. 4) It allows the evaluation of TCP performance through the average throughput or the average goodput. Both of these performance measures are obtained by averaging a sample of executions of the algorithm. Such algorithm is validated by its comparison with a set of ns-2 simulation results. The loss behavior of the network is modeled by the Gilbert loss model. An average error of 2.15 % is obtained.



# Introducción

El presente trabajo de tesis tiene como tema de investigación estudiar el desempeño del protocolo de control de transporte (TCP: *Transport Control Protocol*) en redes cableadas.

TCP fue diseñado originalmente para su operación en redes cableadas. Estas redes se caracterizan por que la pérdida de los datos se debe principalmente a los problemas de congestión en sus nodos “cuellos de botella”. Adicionalmente, y en comparación con una red inalámbrica o móvil, en una red cableada el tiempo “de ida y vuelta” en una conexión varía lentamente, esto debido a la relativa “estabilidad” de los elementos en la red. Además, la pérdida de los reconocimientos es relativamente poco probable ya que su tamaño es mucho menor que el de un paquete de datos. TCP ha incorporado una dinámica de transmisión que intenta aliviar situaciones de congestión en la red para evitar un colapso de la misma.

En las últimas dos décadas se ha incrementado considerablemente la diversificación de los escenarios de comunicación sobre redes de computadoras e Internet. El desempeño de TCP para todos y cada uno de estos tipos de escenarios se ha convertido en objeto de estudio. La mayoría de las principales variantes de TCP actuales están basadas en los estándares TCP Reno, TCP NewReno y TCP SACK, los cuales fueron pensados para operar sobre redes cableadas. Por ello, el análisis del desempeño de TCP en redes cableadas debe ser un paso ineludible en el estudio de cualquier variante de TCP sobre otro tipo de escenario de comunicaciones. Este trabajo se enfoca al análisis de TCP NewReno, por ser una de las versiones de TCP más ampliamente usada actualmente.

El objetivo de esta tesis es desarrollar una herramienta de evaluación del desempeño de TCP NewReno en redes cableadas para flujos de datos de cualquier tamaño, que incluya una representación completa y precisa de sus mecanismos y que permita usar diferentes modelos de pérdidas para representar una gran variedad de redes. La necesidad de cubrir dicho objetivo surge de la revisión de las herramientas de evaluación del desempeño de TCP NewReno actuales, que consisten en modelos analíticos desarrollados para evaluar la tasa de transmisión

en estado estable de este protocolo con flujos de larga duración. Usualmente, una de las formas de validación de los modelos analíticos es a través de la comparación de sus resultados contra los obtenidos de experimentos de simulación, usando para ello una plataforma universal de simulación de redes, como ns-2. Debido al alto grado de complejidad involucrado en la dinámica de la transmisión en una conexión TCP, todos los modelos recurren al uso de simplificaciones tanto del protocolo como de la red para poder desarrollar un modelo adecuado pero con un nivel de complejidad manejable. Entre estas simplificaciones es común encontrar que se consideran redes con modelos de pérdidas de paquetes independientes y una serie de restricciones en las condiciones de operación de los mecanismos del protocolo, principalmente de los mecanismos asociados con la recuperación de segmentos perdidos. Como consecuencia, el error obtenido de la estimación de la tasa de transmisión, en comparación con la tasa de transmisión resultante de la simulación, puede llegar a ser inaceptable en escenarios que se salen de las suposiciones sobre las que los modelos se desarrollaron. Un porcentaje importante de los flujos de datos transitando en Internet consiste en flujos de datos cortos. No obstante, los modelos analíticos existentes se enfocan únicamente a flujos de datos de larga duración. Por lo tanto, es clara la ausencia de una herramienta de evaluación del desempeño de este protocolo que pueda aplicarse también a flujos de datos cortos y que incluya un modelo más completo del protocolo así como distintos modelos de pérdidas de paquetes, no sólo independientes.

El contenido de esta tesis incluye un repaso del estándar TCP NewReno, la revisión de las propuestas de análisis de su desempeño sobre redes cableadas que se han desarrollado hasta la fecha y dos propuestas de evaluación de su desempeño bajo distintas restricciones. En el primer capítulo se incluye una breve descripción del protocolo de control de transporte, en particular de las versiones estandarizadas TCP Tahoe, TCP Reno y TCP NewReno. En el segundo capítulo se presenta la revisión de los trabajos más importantes en el área de modelado de TCP Reno y NewReno, con el señalamiento de sus alcances y limitaciones. En el tercer capítulo se desarrolla una propuesta de evaluación del desempeño de TCP NewReno con un flujo de datos de larga duración sobre una red caracterizada por su modelo de pérdidas, que brinda una solución de algunos de los problemas abiertos detectados en el estudio sobre este tema. En el cuarto capítulo se desarrolla una generalización de la propuesta de evaluación de TCP NewReno presentada en el capítulo tres, de forma que sea aplicable a un flujo de datos de cualquier tamaño. Finalmente, se presentan las conclusiones de la tesis y el trabajo a futuro.

# Objetivos

El objetivo general del presente trabajo doctoral es el de desarrollar una herramienta de evaluación del desempeño del protocolo TCP NewReno en redes cableadas para flujos de datos de cualquier tamaño, que incluya la representación completa y precisa de sus mecanismos y que sea aplicable para una gran variedad de redes.

Para su consecución, se establecen los objetivos particulares siguientes:

- Realizar un estudio detallado de los mecanismos del protocolo TCP NewReno y sus predecesores.
- Estudiar los modelos analíticos desarrollados para evaluar el desempeño de TCP.
- Realizar una revisión detallada de los modelos analíticos del desempeño de TCP NewReno, detectando sus alcances y limitaciones.
- Desarrollar una primera propuesta de solución a los problemas abiertos en el área de evaluación del desempeño de TCP NewReno, específicamente, proponer una herramienta alternativa al modelo analítico que logre evaluar el desempeño de este protocolo con flujos de datos de larga duración en redes con pérdidas no independientes y que involucre una mejor aproximación de la operación de los mecanismos del protocolo.
- Generalizar la propuesta de solución anterior para crear un algoritmo de evaluación de TCP NewReno para flujos de cualquier tamaño, usando una representación más completa y precisa del protocolo.



# Capítulo 1

## Protocolo de control de transporte, TCP.

El protocolo de control de transporte, TCP, es un protocolo orientado a conexión que permite la comunicación de datos entre procesos de forma bidireccional y confiable. Fue diseñado para operar dentro de una arquitectura de protocolos (en la capa 4 del modelo OSI) sin suponer confiabilidad de los protocolos de las capas inferiores [1]. La definición de TCP y sus continuas actualizaciones y mejoras están documentadas en una gran cantidad de RFCs (*Request For Comments*) publicados por la IETF (*Internet Engineering Task Force*) [2 – 15] [23 - 25] [61]. En [16] se puede encontrar una buena guía para ubicar los RFCs relacionados con diversos aspectos de TCP. TCP cumple con las siguientes funciones: transferencia ordenada y segmentada de datos, confiabilidad (recupera datos dañados, perdidos, duplicados o entregados fuera de orden), control de flujo, multicanalización/ desmulticanalización y control de congestión (la cual no forma parte de su diseño original y fue incorporada posteriormente). En la tabla 1 se resumen los principales RFCs asociados con cada una de estas funciones [17].

### 1.1. Operación de TCP.

Una conexión TCP se establece para lograr una comunicación confiable entre dos aplicaciones. Se identifica por el par de *sockets* (dirección IP/puerto TCP) de las aplicaciones fuente y destino [1]. En el establecimiento de la conexión se acuerdan varios parámetros y mecanismos que serán vigentes durante la conexión, tales como: el tamaño máximo de segmento (MSS: *Maximum Segment Size*) [2], la opción de usar reconocimientos (ACKs: *ACKnowledgments*) selectivos como mecanismo de recuperación de segmentos dañados (SACK: *Selective ACK*) [8][9], la opción de escalar el tamaño de la ventana [7], la opción de manejar notificaciones explícitas de congestión (ECN: *Explicit Congestion Notification*) [10], etc.

Tabla 1.1. Resumen de las funciones principales de TCP y los estándares que las tratan [17].

<b>Función TCP</b>	<b>Implementación</b>	<b>RFCs</b>
Transferencia ordenada y segmentada de datos.	Establecimiento y terminación de la conexión. Opción MSS. Descubrimiento de MTU ( <i>Maximum Transmission Unit</i> ).	793, 1122. 879. 1191.
Multicanalización /desmulticanalización.	Número de puerto.	793, 1122.
Control de flujo.	Tamaño de la ventana receptora. Eliminación del síndrome de ventana "tonta" ( <i>silly window</i> ). Algoritmo Nagle. Opción de escalado de la ventana.	793, 1122. 813, 1122. 896, 1122. 1323.
Control de errores.	Cálculo para la detección de un segmento erróneo. Números de secuencia. Protección contra secuencias traslapadas. ACKs acumulativos y selectivos.  Retransmisión de segmentos fallidos y temporizador para la retransmisión.	793, 1071. 793. 1323. 1122, 2018, 2883. 1122, 2988.
Control de congestión.	Algoritmo Karn. Tamaño de la ventana inicial. Comienzo lento ( <i>Slow Start</i> ). Evasión de la congestión ( <i>Congestion Avoidance</i> ). Retransmisión rápida y recuperación rápida ( <i>Fast Retransmit and Fast Recovery</i> ). Soporte de notificación explícita de congestión (ENC).	2988. 3390. 5681. 5681. 5681, 3782, 3517, 6582. 3168.

Una vez establecida la conexión TCP, el transmisor TCP coloca los datos que recibe de la aplicación fuente en una cola de salida, enumerando cada byte.

La tasa de transmisión que rige la conexión TCP obedece a dos restricciones: la capacidad de procesamiento de la entidad TCP receptora (control de flujo) y el ancho de banda disponible en la red, en la ruta desde el transmisor TCP hasta su receptor (control de congestión).

La transmisión de los datos se hace en grupos de bytes llamados segmentos. El tamaño de los segmentos puede variar a lo largo de la conexión, pero sin sobrepasar el MSS. Cada segmento se identifica por el número de secuencia (SN: *Sequence Number*) del primer byte de datos que contiene. Existen algoritmos para evitar la transmisión de datos en segmentos muy pequeños que hagan ineficiente la transmisión, tanto acumulando los bytes de datos en la cola de salida como evitando actualizaciones de la ventana de recepción en porcentajes pequeños [3 - 5].

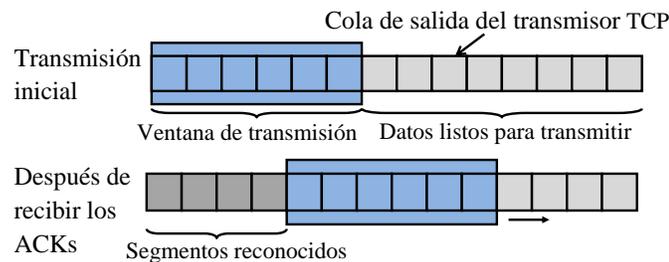


Figura 1.1: El concepto de ventana deslizante. Deslizamiento de la ventana del transmisor [19].

Se usa una ventana deslizante en el transmisor y otra en el receptor (*rwnd*) para establecer la dinámica de la transmisión de datos. Cuando el transmisor transmite los datos de su ventana, la cierra, para evitar una transmisión subsecuente. El receptor TCP verifica la integridad de cada segmento recibido. Si es correcta, lo almacena en su *rwnd* y envía un ACK al transmisor con el SN del siguiente byte que espera recibir, dando por hecho que todos los bytes anteriores a éste ya han sido correctamente recibidos (ACK acumulativo) y que están siendo procesados para su entrega a la aplicación destino. El receptor TCP usa los ACKs para informarle además al transmisor el tamaño de su *rwnd*. Cuando el transmisor TCP recibe la realimentación desde el receptor, mediante ACKs, que confirma la recepción adecuada de los datos previamente enviados, abre de nuevo la ventana de transmisión, y la desliza para permitir el envío de los siguientes datos en la cola de transmisión (figura 1.1). El transmisor TCP ajusta el tamaño de su ventana para no saturar al receptor, es decir, de forma que no exceda el de la *rwnd* (figura 1.2). Algunas versiones de TCP usan otro tipo de ACKs, llamados ACKs selectivos (SACKs), con los que, además de informar al transmisor sobre el siguiente SN que se espera para la recepción ordenada de los datos, también se le informa sobre los posibles huecos en la secuencia de numeración; esto con el fin de proporcionarle información más detallada sobre los segmentos que han llegado realmente al receptor y que pueda hacer más eficientemente sus retransmisiones.

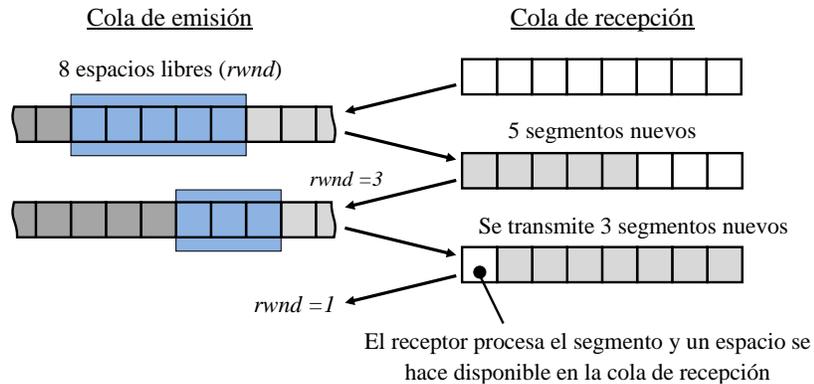


Figura 1.2: El concepto de control del flujo: el receptor informa el tamaño de su  $rwnd$  al transmisor, el cual ajusta el tamaño de su ventana para que no la exceda.

El transmisor TCP cuenta con un temporizador de retransmisión. El temporizador se activa con el envío del primer segmento de datos. Si se recibe un ACK que incluya el reconocimiento de este segmento, sucede una de dos cosas: si se ha enviado un siguiente segmento, aún no reconocido por el ACK recibido, entonces se actualiza el temporizador, asociándolo ahora con éste segmento en transmisión; si no hay más segmentos enviados sin reconocer, entonces se cancela el temporizador. El evento de expiración del temporizador (TO: *TimeOutevent*) se considera como una señal de que uno o más segmentos se perdieron por causa de la congestión de la red. Como consecuencia, se comienza con la retransmisión de los segmentos, a partir del segmento asociado con el temporizador, y se activa el mecanismo de control de congestión. La duración del temporizador de retransmisión (RTO: *Retransmission TimeOut duration*) se calcula dinámicamente durante la conexión como un intento de capturar las condiciones de operación de la red sin congestión, reflejadas en el tiempo de “ida y vuelta” (RTT: *Round-Trip Time*) [5] [11].

El mecanismo de control de congestión de TCP tiene como objetivo evitar o aliviar una condición de congestión de la red, pues ésta puede llevar a una reducción de la tasa efectiva de transmisión a prácticamente cero. Existe una gran cantidad de propuestas de mecanismos de control de congestión para TCP. La mayoría de ellas no están estandarizadas y sólo algunas cuentan con una implementación amplia en Internet [18] [19].

En principio, todo mecanismo de control de congestión tiene por objetivo encontrar un tamaño de la ventana de transmisión óptimo, con el cual se pueda evitar la congestión de la red, o aliviarla, si ya está presente. Así, el tamaño de la ventana, además de estar limitada

por la *rwnd*, está restringido por un estimado de la capacidad de la red, en la ruta desde el transmisor hacia el receptor; es decir, el tamaño de la ventana se ajusta a la cantidad de bytes que se estima pueda ser entregada a la red sin congestionarla. Debido a que en la mayoría de los casos, la tasa de procesamiento en los ordenadores supera por varios órdenes de magnitud a la tasa de transmisión que una red puede ofrecer, es probable que la *rwnd* no imponga una restricción mayor que la impuesta por las condiciones cambiantes de la red. Por lo anterior, es muy frecuente encontrar que la tasa de transmisión de TCP esté gobernada por la congestión de la red, cuya estimación se refleja en el tamaño de la ventana del transmisor, normalmente denominada ventana de congestión *-cwnd-* [19][37][48][49].

Todos los mecanismos de TCP estandarizados se basan en información de los extremos de la conexión (principio extremo-a-extremo), lo cual significa que no dependen de ningún tipo de señalización explícita de la red. Esto dificulta grandemente la estimación precisa del ancho de banda disponible en la red, por su naturaleza compleja y cambiante en espacio y tiempo. En consecuencia, es prácticamente imposible regular con exactitud la tasa de transmisión TCP de forma que se evite llevar a la red a un estado de congestión. Por ello, han proliferado propuestas de mecanismos de congestión para TCP que violan este principio y usan información directamente de la red, lo que permite saber con mayor claridad las condiciones actuales en ella. La ventaja de este tipo de propuestas, evidentemente, es que proporcionan mayor información de la red, y por ende, permiten el diseño de estrategias que se ajusten más adecuadamente a la misma. La desventaja es que la mayoría de las implementaciones soportan las versiones clásicas (principalmente TCP NewReno y TCP SACK [19] [21]), y que desplegar masivamente estas nuevas versiones suponen un largo proceso de estandarización e implementación. Debido a esto, ambos tipos de tendencias tienen permanencia en el trabajo sobre TCP [18 - 20].

Refiriéndonos sólo a las propuestas que respetan el principio extremo-a-extremo, una de las clasificaciones más generales que existen de las variantes de TCP se refiere al objetivo buscado por los mecanismos de control de congestión: evitar la congestión o aliviarla. Las primeras se denominan propuestas pro-activas, mientras que las segundas se conocen como propuestas reactivas [18] [19].

Las primeras propuestas que aparecieron para resolver el problema de la congestión en TCP son reactivas [8] [22 - 25], las cuales se han refinado con el tiempo [13 - 15] [61] y se han convertido en las más ampliamente implementadas en la Internet [18 - 21]. Éstas se caracterizan por comenzar la transmisión del flujo TCP con una tasa de transmisión pequeña, con

la finalidad de probar “poco a poco” la condición en la que se encuentra la red. Mientras no se tenga indicación de algún segmento perdido, se va incrementando exponencialmente la tasa de transmisión. En el momento en que se detecta la primera pérdida, ésta se interpreta como un estado de congestión de la red, a lo que se responde con una reducción a la mitad de la tasa de transmisión y se opta por un incremento lineal. A esta técnica se le conoce como incremento aditivo/decremento multiplicativo, (AIMD: *Additive Increase/Multiplicative Decrease*). La secuencia histórica de estas propuestas para el control de congestión de TCP es como sigue:

TCP Tahoe [22] [23] incorpora dos conjuntos de algoritmos: por un lado un conjunto de algoritmos relacionados con el control de congestión, llamados “comienzo lento” (SS: *Slow Start*), “evasión de la congestión” (CA: *Congestion Avoidance*) y “retransmisión rápida” (*fast retransmit*); por otro lado, incorpora otro conjunto de algoritmos que refinan el cálculo de la RTO [11].

Como una mejora sobre Tahoe, TCP Reno introduce el algoritmo de “recuperación rápida” (FR: *Fast Recovery*) [24] [15]. Su objetivo es que la respuesta del transmisor TCP ante un estado de congestión incipiente de la red sea distinta de la que tiene frente a un estado de congestión severa, de forma que se evite la disminución drástica de la tasa de transmisión como consecuencia de la detección de cualquier pérdida.

Mediante varios estudios se descubrió que el desempeño de TCP Reno se reduce significativamente en escenarios en donde aparecen más de un segmento perdido en la transmisión de una ventana de datos. Ésto debido a que la FR se activa por cada segmento perdido en la misma ventana transmitida (FRs sucesivos), reduciendo el tamaño de la *cwnd* a la mitad en cada ocasión [26] [15]. Para aliviar esta deficiencia de Reno aparecen dos propuestas, las cuales se han convertido en las más ampliamente implementadas en la Internet [18 - 21]: TCP NewReno [14] [25] [61] y TCP SACK [13]. Ambas permiten la recuperación de varios segmentos perdidos dentro de la misma ventana mediante una modificación del algoritmo FR de Reno, con lo que se impide que se reduzca la tasa de transmisión más allá de la mitad y se evita, en gran parte, la recuperación de segmentos perdidos a través de un TO.

## 1.2. TCP Tahoe.

En TCP Tahoe [22] se incluye un conjunto de algoritmos que refinan el cálculo de la RTO. La importancia de éste parámetro se debe a que su relación con el RTT, mediante el cociente  $RTO/RTT$ , define el carácter (conservador o ineficiente) del mecanismo de detección y recuperación de pérdidas detectadas por un TO. Si la relación  $RTO/RTT$  es muy grande, entonces se tiene un comportamiento muy conservador: la reacción del control de congestión de TCP se retrasa porque se considera que el TO no necesariamente indica congestión de la red. Como consecuencia, si en realidad existe un estado de congestión, ésta se ve agravada. Por el contrario, si la relación  $RTO/RTT$  es cercana a la unidad, entonces pueden generarse excesivas retransmisiones innecesarias al confundir una pérdida de segmento con el simple retardo en su transmisión a través red. Los algoritmos en [11] cumplen con varias funciones: definen la expresión para calcular dinámicamente la RTO, con el fin de seguir de manera cercana las variaciones de la red; identifican las muestras válidas del RTT usadas para dicho cálculo (algoritmo de Karn); y establecen el mecanismo de *back-off* de la RTO, que supone un incremento exponencial de la RTO ante TOs sucesivos. Es decir, con cada TO, la RTO se duplica, de manera que si la retransmisión generada por el TO falla, el TO siguiente aparecerá en el doble del tiempo que el anterior. Como máximo, la RTO se incrementará hasta un valor máximo definido en [11].

En el momento en que el receptor TCP Tahoe recibe correctamente un segmento, con el cual rellena parcial o totalmente la secuencia de datos en su *rwnd*, envía un ACK con el SN del siguiente byte que espera recibir en este orden. Si en lugar de recibir un segmento con el SN esperado, recibe otro fuera de orden, entonces emite de forma inmediata un ACK con el SN del siguiente byte en orden que espera recibir, lo que se traduce en un ACK duplicado (DA: *Duplicate ACK*). Por ello, la recepción de varios DAs puede interpretarse como la señal de un segmento perdido. El algoritmo de “retransmisión rápida” establece que el transmisor TCP interprete la recepción de una cantidad específica de DAs iguales como señal de la pérdida del segmento con el SN indicado por ellos.

El algoritmo “comienzo lento”, SS, de TCP Tahoe define el incremento de la *cwnd*, desde un valor inicial (*iwnd*) [12], con una tasa de crecimiento exponencial de base 2. Por cada ACK que reconoce nuevos datos (RA: *Regular ACK*), la *cwnd* se incrementa en un MSS (incremento multiplicativo). Este incremento continúa hasta que sucede una de dos cosas: se detecta la pérdida de un segmento o el tamaño de la *cwnd* alcanza (o rebasa) el del parámetro *ssthresh* (umbral del SS), el cual se inicializa con un valor arbitrariamente alto. Si sucede lo

segundo, se activa el algoritmo de “evasión de la congestión”, CA. Durante la CA, la  $cwnd$  se incrementa en aproximadamente un MSS cada RTT (incremento aditivo), o equivalentemente, con cada RA se usa la siguiente expresión:

$$cwnd = cwnd + MSS(MSS/cwnd) \quad (1.1)$$

La CA concluye cuando se detecta una pérdida o cuando se termina con la transmisión de los datos. Existen dos maneras de detectar la pérdida de un segmento: con el TO o mediante el algoritmo de “retransmisión rápida”. Independientemente de la forma en que se detecta una pérdida, TCP Tahoe reacciona disminuyendo su  $cwnd$  al valor de la ventana de pérdida ( $lwnd$ ), la cual normalmente equivale a la  $iwnd$ : Después se activa el mecanismo de SS, con el valor del  $ssthresh$  dado por:

$$ssthresh = \text{máx}(FlightSize/2, 2(MSS)) \quad (1.2)$$

donde  $FlightSize$  es la cantidad de datos en curso en la red, es decir, es la diferencia entre el número máximo de secuencia enviado y el número máximo de secuencia reconocido.

Considerando que las pérdidas de los segmentos se deben únicamente a la congestión en una red en estado estable y que las retransmisiones siempre llegan exitosamente, la dinámica de TCP Tahoe, una vez superada la fase de SS inicial, se refleja en una dinámica cíclica del tamaño de la  $cwnd$ . Como se muestra en la figura 1.3, una conexión TCP incrementa su  $cwnd$  durante el SS inicial hasta alcanzar el estado de congestión incipiente en la red, punto que se refleja en el valor al que se actualiza el  $ssthresh$ . A partir de ahí, el tamaño de la  $cwnd$  oscilará en ciclos SS - CA desde la  $iwnd$  hasta el  $ssthresh$ , cambiando de mecanismo cuando equivale al  $ssthresh$ .

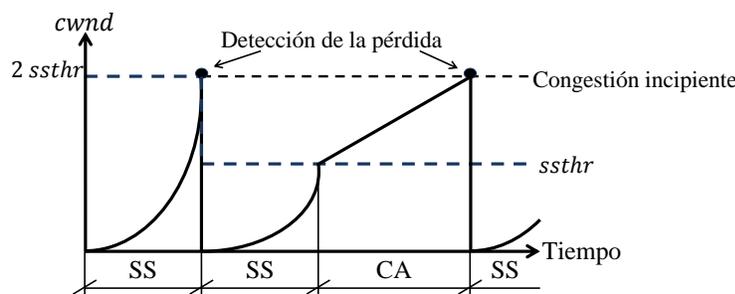


Figura 1.3: Dinámica de la  $cwnd$  en TCP Tahoe [19].

### 1.3. TCP Reno.

TCP Reno [15] mantiene los algoritmos SS, CA, “retransmisión rápida” y de recuperación de pérdidas detectadas por un TO, tal y como están definidos en Tahoe, excepto porque introduce una modificación en SS: por cada RA, la *cwnd* se incrementa por lo menos en la cantidad de bytes reconocidos por este RA y a lo más en un MSS. Como una mejora sobre Tahoe, TCP Reno introduce el algoritmo de “recuperación rápida”, FR.

El objetivo de la FR es el de evitar que la tasa de transmisión se disminuya drásticamente ante un evento de congestión ligera, o pasajera, de la red. La base para este algoritmo es la distinción entre los estados de congestión ligera y severa. Se considera como indicación de una congestión severa el que la pérdida de un segmento sea detectada mediante un TO. Se supone un estado de congestión ligera o pasajera cuando la detección de una pérdida es a través de DAs múltiples, por lo general tres (TD: *Triple Duplicated ACK*). Lo anterior se debe a que un TD representa no sólo que un segmento se ha perdido, sino también que los segmentos siguen llegando al receptor.

Al detectarse un TD, se retransmite el segmento perdido, se actualiza el valor del *ssthresh* según (2) y se le asigna este valor al tamaño de la *cwnd*, con lo que se inicia la fase FR, la cual terminará con la recepción de cualquier RA (figura 1.4, estado 2). Una vez dentro de la FR, el valor de la *cwnd* se incrementa momentáneamente en un MSS con cada nuevo DA recibido (así que en principio, la *cwnd* se incrementa en 3 MSS, por el TD) –figura 4, estado 3. Cada vez que el  $\min(cwnd, rwnd)$  lo permita, se envían nuevos segmentos. Eventualmente, es posible que si la cantidad de DAs es suficiente, se pueda continuar la transmisión de datos nuevos (figura 1.4, estado 4). Al llegar un RA, la ventana se decrementa y se recorre según el SN del RA (figura 1.4, estado 5). En el caso de un TO, se sigue con el mecanismo de recuperación de pérdidas detectadas por un TO.

Considerando que las pérdidas de los segmentos se deben únicamente a estados de congestión pasajera en una red en estado estable (es decir, que todas las pérdidas se detectan por TDs), que el número de pérdidas en una ventana de transmisión es pequeño y que las retransmisiones siempre llegan exitosamente, la dinámica de Reno, una vez superada la fase de SS inicial, se traduce en una dinámica cíclica del tamaño de la *cwnd* (ver figura 1.5). Una conexión TCP Reno incrementa su *cwnd* durante el SS inicial hasta alcanzar el estado de congestión incipiente en la red. Este punto se refleja en el valor al que se actualiza el *ssthresh*. A partir de ahí, el tamaño de la *cwnd* oscilará en ciclos FR - CA desde *ssthresh* hasta 2 veces el *ssthresh*.

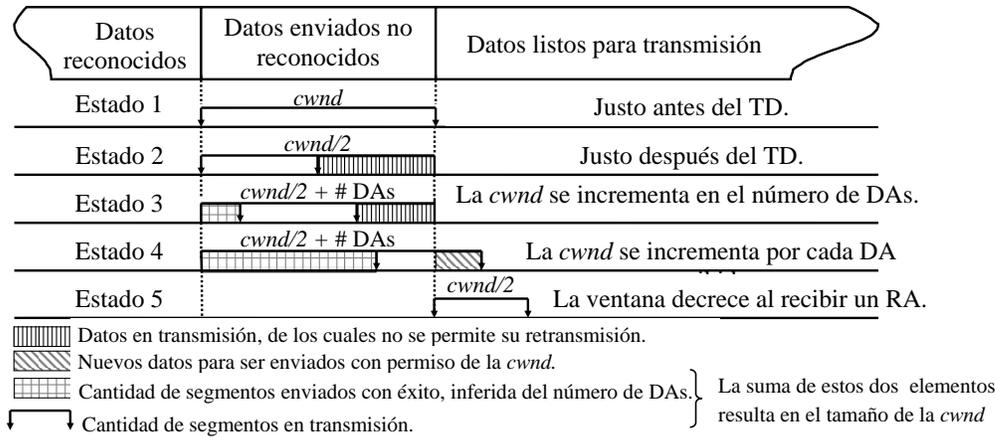


Figura 1.4: Algoritmo FR de TCP Reno [19].

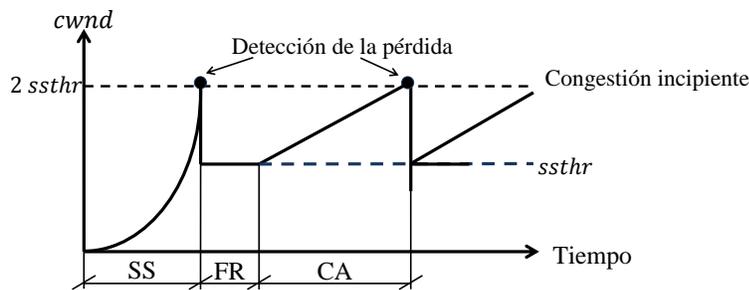


Figura 1.5: Dinámica del tamaño de la  $cwnd$  en TCP Reno [19].

Como se mencionó anteriormente, TCP Reno presenta el problema de tener un desempeño muy pobre en escenarios en donde aparecen más de un segmento perdido en una misma ventana de datos. Esto se debe a que la recuperación de todos los segmentos perdidos dentro de la misma ventana de transmisión supone, en el mejor de los casos, una sucesión de FRs (la  $cwnd$  se disminuye a la mitad, o menos, al entrar en cada nueva FR), y en el peor de los casos, un TO (la  $cwnd$  decrece hasta  $lwnd = 1 \text{ MSS}$  según [15]). En la figura 1.6 se muestra un caso que ejemplifica lo anterior. Suponiendo que la cantidad de segmentos en transmisión equivale a  $cwnd$ , que más de un segmento se pierde, que todas las pérdidas se detectan mediante TDs y que las retransmisiones siempre son exitosas, la  $cwnd$  se reduce a la mitad con la detección del primer segmento perdido (figura 1.6, estado 2). Se recibe el RA que responde a la retransmisión de éste y se presenta otro TD que indica la siguiente pérdida en la ventana, volviéndose a reducir la  $cwnd$  a la mitad (figura 1.6, estado 3). Bajo estas condiciones, este proceso sucede así tantas veces como pérdidas hubo en la ventana (figura 1.6, estado 4). De

manera que la recuperación de pérdidas múltiples en la misma ventana se traduce en una reducción drástica de la tasa de transmisión de TCP Reno.

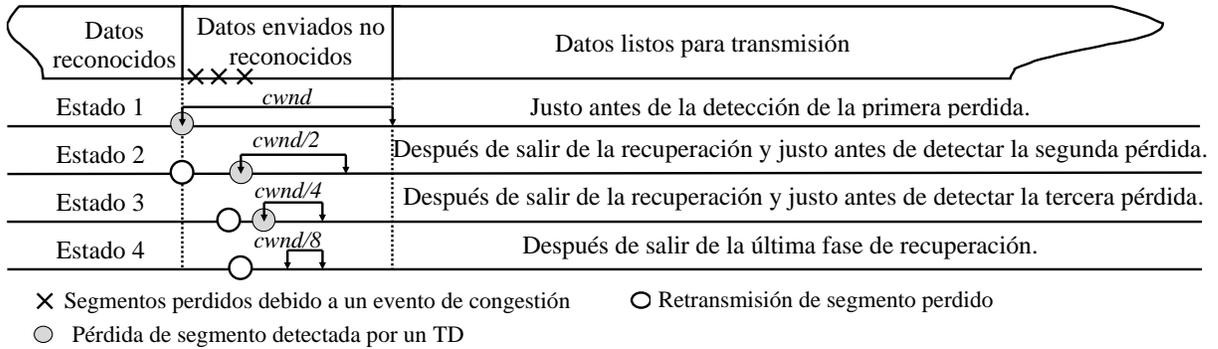


Figura 1.6: Operación ineficiente de Reno ante varias pérdidas por ventana [19].

Como se muestra en [26], en ausencia de SACK, TCP NewReno soluciona el problema de Reno ante múltiples segmentos perdidos por ventana. Por ello, en [15] se recomienda que para recuperar varias pérdidas en una ventana, junto con Reno se implementen los mecanismos de SACK o NewReno. Aunque la implementación de SACK ha seguido una tendencia creciente, de manera que se habla de una extensión de su implementación en Internet del 90% o más, no ha sucedido lo mismo con el porcentaje del uso real de SACK como mecanismo de recuperación de pérdidas, el cual se ha reportado muy por debajo de la mitad [35]. Ambos mecanismos permiten recuperar varias pérdidas por ventana con las siguientes ventajas y desventajas: SACK recupera más rápido que NewReno los segmentos perdidos, pero no todos los extremos TCP tienen configurado o usan adecuadamente esta opción; NewReno puede tardar más en la recuperación de los segmentos perdidos, sobre todo si son muchos, sin embargo es más común encontrar implementada esta versión de TCP en cualquier extremo TCP [21].

## 1.4. TCP NewReno.

TCP NewReno [61] permite la recuperación de varios segmentos perdidos dentro de la misma ventana de transmisión sin tener que recurrir a una secuencia de FRs consecutivos. Ésto mediante una modificación al algoritmo FR de Reno (ver figura 1.7), que consiste en lo siguiente:

- Se incluye un nuevo parámetro llamado *recover*, el cual almacena el valor del máximo SN que ha sido enviado hasta el momento de la detección de una pérdida, ya sea por un TD o por un TO.
- Un TD se considera como una indicación de pérdida sólo si el SN reportado por los DAs es mayor que *recover*. Si es así, se actualiza la variable *recover* y se comienza con el algoritmo de FR. Esta medida es la que permite evitar la ocurrencia de FRs consecutivos para recuperar varias pérdidas dentro de una misma ventana de transmisión.
- Adicionalmente, se distingue entre dos tipos de RA dentro de la FR: si el SN del RA es mayor que *recover* se le llama ACK total; en caso contrario, RA parcial (PA: *Partial ACK*).
- En el caso de recibir un PA, se infiere que el segmento con SN reportado por el PA también se ha perdido y se retransmite. La *cwnd* se decrementa en la cantidad de nuevos datos reconocida por el PA y se incrementa en un MSS, para reflejar la retransmisión exitosa (estados 4, 5 y 6 de la figura 1.7).
- El algoritmo FR termina sólo si se recibe un ACK total (estado 7 de la figura 1.7) o si el temporizador expira. Si se recibe un ACK total, la *cwnd* se actualiza como 1)  $cwnd = ssthresh$  ó 2)  $cwnd = \min(ssthresh, \max(MSS, FlightSize) + MSS)$ , con el valor actual del *FlightSize*.

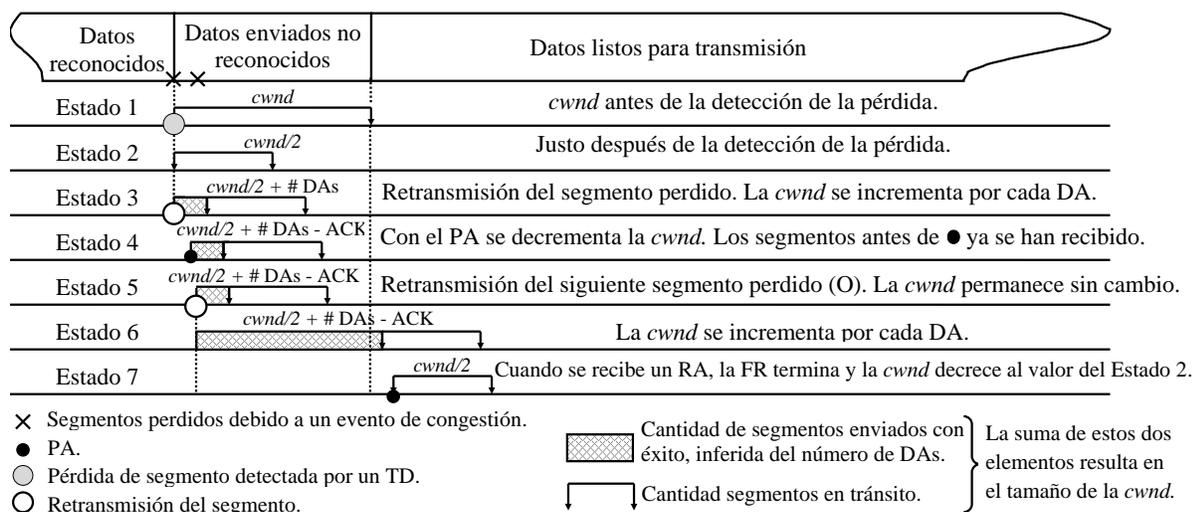


Figura 1.7: Algoritmo de FR en TCP NewReno [19].

Con estas modificaciones, ante segmentos múltiples perdidos en la misma ventana de transmisión, NewReno logra mejorar en gran medida el desempeño de Reno [26].

En [14] se definen dos variantes de TCP NewReno: TCP impaciente (*Impatient TCP*) y TCP “lento pero firme” (*Slow-but-Steady TCP*). La versión de TCP “lento pero firme” supone el restablecimiento del temporizador con cada PA que se reciba. Por su lado, TCP impaciente sólo restablece el temporizador con la llegada del primer PA. La diferencia se hace significativa en escenarios con una gran cantidad de segmentos perdidos dentro de la misma ventana de transmisión: con TCP impaciente seguramente se terminará por activar un TO para recuperar el total de las pérdidas; con TCP “lento pero firme” la FR durará tantos RTTs como sea el número de segmentos perdidos en la ventana. La recuperación será más rápida con alguna de las dos versiones según sea el tamaño del patrón de pérdidas en una misma ventana. Ninguna de las dos se reconoce como mejor, sino que dependiendo del escenario, en algunas ocasiones el desempeño de una sobrepasa el de la otra y en otras ocasiones sucede lo contrario [49] [14].

En [25] se define la modificación llamada *bugfix*, con la cual se asegura que no se pueda activar una FR sin haber terminado por completo la recuperación previa, activada por un TO. En su variante *Careful*, la condición que implica que la última recuperación de pérdidas activada por un TO se ha completado es que los DAs cubran más que *recover*, es decir, que al menos sea reconocido un byte más que el valor de *recover*. En su variante *Less careful*, la condición es que los DAs cubran por lo menos *recover*. En [61] la modificación *bugfix* en su opción *Careful* queda incluida en el estándar.

En la siguiente sección se desarrolla el contexto histórico del modelado de las versiones de TCP explicadas aquí, en particular de TCP NewReno, con una muestra de los trabajos más significativos en el área. Ya que NewReno es una modificación de Reno, y éste a su vez lo es de Tahoe, la mayoría de los trabajos fundamentales en el modelado de NewReno se derivan de propuestas realizadas para Reno y Tahoe. El enfoque se dirige hacia NewReno por ser una de las versiones de TCP más ampliamente implementadas en la Internet [18 - 21].



## Capítulo 2

# Modelado del desempeño de TCP en redes cableadas

La evaluación por simulación del desempeño de un protocolo es una herramienta muy útil y práctica de análisis y diseño, sin embargo, comúnmente requiere de un alto consumo de recursos computacionales y de tiempo, su complejidad es inversamente proporcional al nivel de abstracción del fenómeno y su uso se complica si se requiere abarcar una amplia variedad de escenarios. En contraste, un modelo analítico del desempeño de un protocolo presenta las ventajas de a) servir como referencia para la comparación de diferentes escenarios (representados por un conjunto de datos o parámetros), b) suele ser matemática o computacionalmente tratable y c) su evaluación es mucho más rápida que la de una simulación. Sin embargo, para condiciones sumamente cambiantes como las presentes en Internet [40], en donde el tráfico varía ampliamente a lo largo del tiempo y de un lugar a otro en la red, es común encontrar que un modelo analítico se puede obtener sólo bajo una serie de suposiciones y simplificaciones en la abstracción del fenómeno real. Adicionalmente, un modelo analítico normalmente no representa de manera exacta los diversos cambios abruptos en el comportamiento de un experimento simple, sino que representa sus cambios suaves [27].

La mayoría de las propuestas de modelado analítico del desempeño de TCP en redes cableadas comúnmente usan como medida de desempeño la tasa (o tasa efectiva) de transmisión (THR: *THRoughput*) promedio; algunas otras usan variables directamente relacionados con ésta, como el tamaño de la *cwnd* o la latencia. Todas ellas intentan obtener un modelo que muestre la tendencia del desempeño de TCP como función de diversos parámetros, tales como la probabilidad de pérdida de segmento, la probabilidad de evento de pérdida (que se refiere a uno o más segmentos perdidos dentro de la misma ventana de transmisión), el RTT, la RTO, el tamaño del flujo, el tamaño de la *rwnd*, el patrón de pérdidas (pérdidas uniformemente distribuidas, eventos de pérdida tipo ráfaga con distribución uniforme, eventos de

pérdida uniformes con pérdidas uniformemente distribuidas dentro de cada evento), etc. [17] [28 - 34] [36 - 37] [43 - 44] [48 - 53].

De acuerdo con varios estudios como [38], [39], [41], [42] y [47], aproximadamente el 90 % de los flujos en Internet son de un tamaño menor a los 4 kB. Este fenómeno se puede considerar como una propiedad “invariante” de Internet [17], refiriéndose por éste concepto a una propiedad de Internet que se ha mostrado relativamente constante a lo largo de una gran cantidad de experimentos [40]. Ya que la mayor parte del tráfico en Internet se debe a unos pocos flujos de gran tamaño (conocidos como “elefantes”), mientras que la menor parte se debe a la inmensa mayoría de flujos de corta duración (conocidos como “ratones”), el análisis del desempeño de TCP para flujos de ambos tipos ha sido objeto de gran atención. Mientras que algunos trabajos se han concentrado sólo en los flujos de corta duración, como [17], [29], [31] y [43], otros más, la mayor parte, se han enfocado hacia los flujos de gran tamaño y larga duración, tales como [28], [30], [32], [34], [36], [37], [48], [49], [50] y [52]. Sólo algunas pocas propuestas han incorporado ambos tipos de flujos en su análisis, agregando el tamaño del flujo como un parámetro en su modelo, como [33], [44] y [53]; sin embargo, ninguno de éstos análisis es sobre TCP NewReno.

Las primeras propuestas de modelos analíticos para evaluar el desempeño de TCP recurren a grandes simplificaciones relacionadas comúnmente con el patrón de pérdidas y la reacción de TCP ante ellas. De hecho, en los primeros trabajos no se toman en cuenta los escenarios con pérdidas o éstos se restringen a tener sólo una pérdida por ventana de transmisión, con lo que se evita incluir en el análisis los mecanismos de recuperación de pérdidas (FR y TO - SS). Posteriormente, en el desarrollo de los nuevos modelos se han ido considerando, cada vez con mayor precisión, estos mecanismos. Sin embargo, debido a la gran complejidad de la operación de TCP en un ambiente tan cambiante como Internet, aún se siguen teniendo una gran cantidad de simplificaciones en los modelos. Principalmente, estas simplificaciones son:

- La transmisión de datos es vista a nivel de segmentos de tamaño MSS y no de bytes.
- Los análisis suponen fuentes continuas de datos.
- Se considera que la pérdida de los ACKs es despreciable.
- El RTT se supone invariante a lo largo de la conexión e igual su promedio.
- El tiempo de transmisión de una ventana en ráfaga se considera mucho menor que el RTT.

- La probabilidad de cada evento de pérdida (uno o más segmentos perdidos dentro de la misma ventana) se supone independiente de una ventana de transmisión a otra.
- La probabilidad de evento de pérdida y la duración del RTT se toman como independientes del tamaño de la *cwnd*.
- No se modela el algoritmo Nagle ni los mecanismos para evitar el síndrome de ventana “tonta”.
- Se recurre a una simplificación de los mecanismos del protocolo TCP.
- No se toman en cuenta los retardos del procesamiento en los nodos ni el límite en la cola de salida en el nodo transmisor.

A continuación se presenta la revisión de algunos de los trabajos más representativos en el área.

## 2.1. Contexto histórico.

En [28] se considera que la dinámica de TCP Reno consiste en ciclos de fases CA. En cada ciclo el tamaño de la *cwnd* crece linealmente desde  $W/2$  a  $W$ , con una pendiente que depende de la demora que el receptor TCP impone a la emisión de los ACKs acumulados [5] [15]. La *cwnd* decrece debido a la detección de una pérdida. Las pérdidas se presentan con probabilidad  $p$ . Los autores derivan la expresión para el valor esperado del tamaño de la ventana al comienzo de la fase CA como:

$$W = \sqrt{\frac{8}{3 \cdot p}} \quad (2.1)$$

que coincide con lo reportado en otros trabajos como [30] y [32]. Así mismo, los autores derivan la expresión del ancho de banda usado por TCP, suponiendo que en cada ciclo CA se transmiten  $1/p$  segmentos. Este modelo supone: una tasa pequeña de pérdidas, una sola pérdida por ventana, las pérdidas siempre se detectan por el mecanismo de retransmisión rápida, el RTT es fijo e igual a su promedio y no toma en cuenta los mecanismos de recuperación de pérdidas de TCP. Debido a estas simplificaciones, sus resultados se acercan a los de simulación sólo en condiciones de pocas e infrecuentes pérdidas, mientras que su aproximación a los resultados de mediciones sobre Internet es muy burda. No obstante, se introduce una discusión sobre varios aspectos importantes a considerar en el trabajo de análisis de TCP,

tales como: que el RTT es función del retardo en las colas de los nodos “cuello de botella” del enlace TCP, y sobre el tamaño y duración del flujo simulado que se requiere para poder alcanzar un estado estable. El tipo de restricciones en [28] se puede hallar en muchas otras propuestas tales como [29 - 34].

En [29] se realiza el análisis del THR de flujos pequeños de tipo HTTP sobre distintos protocolos de transporte, entre ellos TCP. Su estudio se enfoca a comparar diversos protocolos sobre los que HTTP puede operar. En particular, para cada protocolo se evalúa el costo de la transmisión de tráfico HTTP sobre redes con distintas características de retardo y de ancho de banda. Es una referencia común en el área de modelado de TCP por ser uno de los primeros trabajos en donde se examinan y modelan aspectos de la fase SS inicial de una conexión TCP, específicamente: el crecimiento de la *cwnd*, la cantidad de segmentos enviados y su duración. El énfasis en esta fase es debido a que el tráfico HTTP se considera dominado por flujos de corta duración, los cuales transcurren, casi en su totalidad, en ella. En este trabajo se considera un ambiente libre de pérdidas. También se analiza el efecto del ACK demorado y de su temporizador (al expirar el temporizador del ACK se fuerza el envío de un ACK, como medida para conservar el reloj de la conexión, el cual está basado en la llegada de los ACKs [5]).

En [30] el THR se evalúa ante distintas relaciones del producto del ancho de banda por el retardo del enlace contra el tamaño de la cola en el nodo “cuello de botella”. El THR se calcula como el cociente del valor estimado de la cantidad de segmentos enviados sobre el valor estimado del tiempo transcurrido en ello. Se enfoca en TCP Tahoe y TCP Reno. Los autores consideran que el comportamiento de TCP es en ciclos compuestos por fases correspondientes con los mecanismos SS y CA, respectivamente. Revisan dos escenarios: sin pérdidas aleatorias y con ellas. Algunas de las suposiciones en las que se basa este modelo son: que el enlace de regreso, donde viajan los ACKs, no tiene pérdidas; que sólo hay una pérdida por ventana; no se consideran los mecanismos de recuperación de pérdidas de Reno; etc. Los autores realizan la comparación de la utilización del enlace en dos escenarios distintos: cuando las pérdidas de los segmentos se deben únicamente al desbordamiento del nodo “cuello de botella” y cuando las pérdidas son aleatorias e independientes. Sus resultados muestran, en particular, que si la probabilidad de pérdida es relativamente alta, la tasa de transmisión (en este caso, reflejada en el tamaño promedio de la *cwnd* al inicio de cada fase CA), es dominada por el efecto de las pérdidas. Es decir, en el segundo escenario de prueba se nota que la utilización del enlace se degrada mucho en tales casos. Los autores en [30] extienden el análisis del THR de TCP para escenarios con más de una conexión simultánea. Para ello, recurren

a varias simplificaciones, tales como el hacer fija la cantidad de segmentos perdidos en una misma ventana de transmisión (en este caso, basado en simulaciones, lo tomaron como 2). De este análisis concluyen que existe una ventaja para conexiones TCP con un RTT pequeño sobre conexiones simultáneas TCP con un RTT mayor. También, y de forma coincidente con [26], observan que se degrada mucho el desempeño de Reno cuando hay varias pérdidas en una sola ventana de transmisión. En general, obtienen una buena aproximación de sus resultados a los de simulación.

En [31] se desarrolla un modelo markoviano simple del tiempo necesario para completar una conexión TCP de un flujo de corta duración como función del RTT, de la probabilidad de pérdida y del valor del *ssthresh* inicial.

En [17] se expone un modelo para representar el incremento de la *cwnd* durante la fase SS inicial. Los parámetros del modelo son: el tamaño de la *iwnd* y el cociente de la duración del RTT entre la duración del temporizador del ACK ( $T_{ACK}$ ). El modelo se valida mediante su comparación con los resultados de pruebas de simulación, realizadas para varios valores de la *iwnd* y para varios valores de la relación  $RTT/T_{ACK}$ . Además, se presenta su comparación contra los resultados de los modelos en [33], [43] y [44] para el rango completo de relaciones  $RTT/T_{ACK}$ . De esta comparación se concluye que todos estos modelos fallan en analizar el crecimiento de la *cwnd* y, por tanto, que su cálculo de la cantidad de segmentos enviada en el SS inicial es incorrecto. Sin embargo, se observa que el modelo en [33] es el más adecuado para los casos en que  $RTT/T_{ACK} \leq 2$  (con errores en el rango del 5 % al 33 %), mientras que el modelo en [17] es el mejor para cuando  $RTT/T_{ACK} > 2$  (con errores de entre el 5 % y el 13 %).

El trabajo en [32], conocido como modelo PFTK, es fundamental en el desarrollo de este tópico, pues marca el inicio de una línea de investigación que usa su mismo principio de análisis. Consiste en el modelo del THR promedio en estado estable como función de: la probabilidad de evento de pérdida, el RTT promedio, la RTO promedio y el tamaño de la *rwnd*. Su gran contribución se debe a que presenta de manera clara y simple la abstracción de la dinámica de TCP Reno como una concatenación de ciclos estadísticamente idénticos, de manera que el THR en estado estable de un flujo TCP de larga duración puede determinarse analizando sólo uno de esos ciclos. El análisis se expone en dos etapas. En la primera, cada ciclo consiste en las fases CA - FR, correspondientes a los mecanismos del mismo nombre. En la segunda, la secuencia de fases CA - FR - CA - FR - ... - TO constituye un ciclo. En los ciclos CA - FR, se supone que las pérdidas son detectadas solamente por TDs. En el otro tipo

de ciclos, también puede haber detección de pérdidas mediante TOs. El THR se define como el cociente del valor esperado de la cantidad de segmentos enviados (exitosamente o no) en un ciclo sobre el valor esperado de su duración. Dentro de cada ciclo, las fases se desarrollan en rondas, cada una con duración aproximada de un RTT y en la que se transmiten *cwnd* segmentos de tamaño MSS. El análisis se enfoca a flujos de larga duración y supone que la aplicación fuente siempre tiene datos para transmitir. Se suponen eventos de pérdida de tipo ráfaga, ésto es, se considera que un evento de pérdida está provocado por el desbordamiento de una cola *Drop-and-Tail* en el nodo “cuello de botella” del enlace, de manera que una vez que se pierde un segmento, todos los segmentos subsiguientes dentro de la misma ronda se consideran también perdidos. Además, supone que: el tiempo de transmisión es mucho menor que el RTT, lo que se traduce en que la cantidad de bytes de datos en transmisión (*FlightSize*) equivale al tamaño de la *cwnd* en ese momento; la probabilidad de cada evento de pérdida es independiente entre rondas y que la probabilidad de evento de pérdida así como la duración de cada ronda es independiente del tamaño de la *cwnd*. En [32] los autores comparan su modelo contra mediciones de 37 conexiones entre 18 nodos distribuidos en E.U.A y Europa. De sus mediciones observan que existe una gran cantidad de detecciones de pérdida realizadas mediante TOs (de ahí la gran importancia de incluir este mecanismo en el análisis). Obtienen una buena aproximación de su modelo a los resultados de medición en escenarios con probabilidad de pérdida baja; sin embargo, encuentran que las curvas se van alejando conforme la probabilidad de pérdida aumenta, con una sobre-estimación del THR que se hace más evidente a mayor probabilidad de pérdida. Esta tendencia de los resultados del modelo PFTK de sobrestimar el THR se ha corroborado en otros trabajos, tales como [36], [37] y [44], de donde se concluye que este error puede deberse a la exclusión en el análisis de las fases FR y TO - SS, encargadas de la recuperación de las pérdidas. La gran importancia de este trabajo se debe a que introduce una técnica de análisis que ha servido de base para una gran cantidad de trabajos posteriores, tales como [33 - 34], [36 - 37], [43 - 44], [48 - 50] y [52], aunque existen otras propuestas que usan técnicas distintas como [45], [46] y [53]. Más adelante se revisarán con más detalle algunos aspectos de este trabajo.

En [33] se toma por objetivo el de modelar el THR de un flujo TCP en función de: la tasa de eventos de pérdida, el RTT, la RTO, la *rwnd* y el tamaño del flujo. Para ello, se retoma el modelo propuesto en [32], que se limita a flujos de gran tamaño, y se completa con el análisis correspondiente para flujos de corta duración. Los autores: a) analizan la latencia en la fase de establecimiento de la conexión TCP y b) obtienen tanto la cantidad de datos enviada durante el SS inicial como su duración, suponiendo que esta fase termina con la primera pérdida detectada en la conexión, a partir del cual opera el modelo PFTK. Comparan sus re-

sultados con los de [32] para una diversidad de condiciones. En pruebas con flujos grandes, que experimentan una tasa de pérdida moderada o alta (superior a  $1e-3$ ), encuentran una gran similitud entre ellos. Sin embargo, los resultados divergen cuando se trata de flujos pequeños (hasta los 256 kB), o cuando la tasa de pérdidas es muy pequeña ( $1e-4$ ) aunque sea un flujo de gran tamaño. Por otro lado, los autores realizan la comparación de su modelo contra resultados de simulación, considerando primero un escenario sin pérdidas y luego con pérdidas. En el escenario sin pérdidas encuentran que la aproximación de sus resultados de latencia tiene un error promedio del 22 %. En el escenario con pérdidas incluyen la comparación con [32] concluyendo que ambos modelos se acercan adecuadamente a la simulación para tasas de pérdida de moderadas a altas, pero para tasas de pérdida muy pequeñas, sólo el modelo en [33] sigue de cerca a la simulación.

En [34] se cumplen dos objetivos: investigar un método no-invasivo apropiado para obtener estimaciones confiables y adecuadas de parámetros que caractericen la red, tales como el RTT y la probabilidad de pérdida; y usar estas estimaciones como parámetros de entrada de un modelo del THR en estado estable de flujos TCP de larga duración. Un método no-invasivo usa información disponible en la red, en este caso dentro de bases de información del protocolo de administración de la red. El modelo propuesto es una modificación del modelo PFTK. La alteración consiste en cambiar al modelo de ser una función de la probabilidad de evento de pérdida a serlo de la probabilidad de segmento perdido. Este cambio se refleja tanto en la probabilidad de tener un TO como en la estimación del valor de la *cwnd* al inicio de la fase CA. En sus resultados se muestra que con este método no-invasivo se logra una buena estimación de la tasa de pérdidas en el caso de tener colas tipo RED en los nodos “cuello de botella” (siendo más precisa mientras mayor es el tamaño de la cola, por contribuir éste a la aleatoriedad de las pérdidas). Por otro lado, cuando se tienen colas tipo *Drop-Tail*, la tasa de pérdidas se subestima. Además, se verifica que las curvas del modelo para el THR de TCP se apegan tanto a las de la simulación y como a las del experimento de laboratorio cuando la estimación de la tasa de pérdidas es adecuada.

En [43] se analiza el THR de flujos TCP de corta duración como función de: la tasa de pérdida de segmentos, el RTT y el tamaño del flujo. Los autores usan la expresión de la latencia de la fase de establecimiento obtenida en [33]. Desarrollan una expresión, en forma de serie de Fibonacci, para modelar el crecimiento de la *cwnd* en el SS inicial. Modelan la latencia en esa fase y la probabilidad de terminarla con un TO. A diferencia de otros trabajos en los que se usa como aproximación para esta probabilidad la de tener un TO estando en CA, en [43] se analizan las condiciones por las que estando en el SS inicial se termina con un TO y

se deriva la probabilidad de este evento. Es necesario notar que, como se esperaba, la probabilidad de tener un TO es distinta estando en el SS que en la CA, por lo que modelarla de la misma forma, como en [33], conlleva un error. Para evaluar su modelo, los autores usan una plataforma de simulación en la que incluyen un módulo de pérdidas en ráfaga. Los resultados del modelo se comparan contra los de [33] y [44], encontrando que, para varios tamaños de flujo (desde 0.5 hasta 11 KB) y con varias tasas de pérdida (desde 0 hasta 0.1), éste modelo reporta un error promedio menor a los reportados por los otros dos.

En [44] se obtiene un modelo para la latencia y el THR en estado estable de TCP Tahoe, Reno y SACK. Los autores utilizan el modelo de transmisión por rondas incorporado en [32] y toman en cuenta los mecanismos de ACK retardado y de control de flujo. Consideran las siguientes etapas en una conexión TCP: establecimiento de la conexión, SS inicial y CA. Retoman la expresión de la latencia de la fase de establecimiento desarrollada en [33]. Proponen una expresión que modela el crecimiento de la *cwnd* en el SS inicial y de ahí derivan el número de rondas (latencia) necesario para transmitir cierta cantidad de segmentos. Calculan la latencia, en rondas, de la transferencia de cierta cantidad de datos en la CA a partir de un cierto tamaño de ventana inicial. Estas expresiones las combinan, junto con la duración de una secuencia de TOs obtenida en [32], en un modelo de la latencia para las tres variantes de TCP en tres escenarios distintos: sin pérdidas, con una sola pérdida y con pérdidas múltiples. Comparan los resultados de los tres tipos de TCP en los diferentes escenarios. Demuestran que en un caso con múltiples pérdidas correlacionadas, Tahoe puede superar el desempeño de los otros dos, al evitar la generación de varios TOs. También realizan la comparación del THR para TCP Reno con el modelo en [32] y la simulación, encontrando una cercanía entre los tres conjuntos de resultados.

En [48] se desarrolla un modelo para el THR en estado estable de una conexión TCP NewReno de larga duración como función de: la probabilidad de evento de pérdida, la cantidad de segmentos perdidos en el evento de pérdida y el valor estimado del RTT. El desarrollo se hace considerando ambas versiones del NewReno: “lento pero firme” e “impaciente”. El modelo en [48] supone una serie de simplificaciones de la dinámica de TCP NewReno: considera que todas las pérdidas se detectan mediante TDs, que los segmentos transmitidos durante la FR y el SS no son sujetos a perderse, que la recuperación de los segmentos perdidos activada por un TO supone un esquema de *Go-Back-N* (en el que se retransmiten de nuevo todos los segmentos desde el que se detectó como perdido hasta el último segmento transmitido en ese momento), etc., Con ello, los autores logran abstraer la parte de la dinámica de TCP que diferencia a ambas versiones, con fines de comparación entre ellas. Con una técnica de análisis

semejante a la usada en [32], con la misma definición del THR en estado estable e incluyendo una aproximación muy simple de la cantidad de datos enviada durante la FR, los autores obtienen una expresión para el THR. Sus resultados los validan por simulación en dos escenarios: en el primero, las pérdidas se generan mediante un módulo de pérdidas tipo Bernoulli; en el segundo, existe tráfico adicional que provoca la pérdida de segmentos por la congestión del nodo “cuello de botella”. En ambos escenarios se verifica que, debido a las suposiciones en las que se basa el modelo, éste sobrestima el THR cuando la tasa de segmentos perdidos es baja y la subestima cuando es alta. Sus resultados muestran que la versión “impaciente” sobrepasa en desempeño a la versión “lento pero firme” sólo bajo condiciones de operación de la red muy extremas, en particular cuando la probabilidad de evento de pérdida es baja pero existen muchos paquetes perdidos en cada evento. En todos los demás escenarios de operación, la versión “lento pero firme” supera a la “impaciente”. Es importante notar que esta conclusión difiere de la mencionada en [14] y [49], en donde no se toma a ninguna de las dos versiones de NewReno como la mejor, sino que, dependiendo del escenario, una de las versiones puede superar a la otra. Más adelante se revisarán con más detalle algunos aspectos de este trabajo.

En [50] se desarrolla un modelo para el THR en estado estable de una conexión TCP NewReno “lento pero firme” de larga duración como función de: la probabilidad de evento de pérdida, la cantidad de segmentos perdidos en el evento de pérdida, el valor estimado del RTT, la cantidad necesaria de segmentos recibidos adecuadamente para generar un ACK y la *rwnd*. Basados en el modelo PFTK, los autores derivan una expresión para la cantidad de datos enviada durante la FR, con una aproximación mejor que la de [48]. Sin embargo, al igual que en [48], se considera: que no hay posibilidad de pérdida de los segmentos nuevos transmitidos durante la FR, que todas las pérdidas se detectan mediante TDs y que las retransmisiones no se pierden. Los autores validan sus resultados por simulación. Usan dos topologías: una con un sólo nodo “cuello de botella” y la otra con un par de nodos “cuellos de botella”. En ambos escenarios obtienen que su modelo sigue a la simulación con un error muy pequeño mientras el valor de la tasa de pérdidas es relativamente pequeño (menor que 0.01) y se separa de ella conforme la tasa de pérdidas aumenta. Ésto se debe principalmente a las simplificaciones usadas para el análisis.

En [49] se desarrolla un modelo para el THR en estado estable de una conexión TCP NewReno “impaciente” de larga duración como función de: la probabilidad de evento de pérdida, la cantidad de segmentos perdidos en el evento de pérdida, el valor estimado del RTT, la cantidad necesaria de segmentos recibidos adecuadamente para generar un ACK, la *rwnd* y el

cociente RTO/RTT. Se modifica el modelo de [50], que es para la versión “lento pero firme” de NewReno, para adaptarlo a la versión “impaciente”. Específicamente, se modela que la fase SS se activa después del TO cuando la cantidad de segmentos perdidos en un evento de pérdida es mayor que el cociente RTO/RTT. La recuperación de los segmentos perdidos activada por un TO supone un esquema de *Go-Back-N*. Los autores realizan una comparación del desempeño de ambas versiones de NewReno, usando el modelo de [50]. La conclusión en [49] se asemeja a la mencionada en [14]: en casos con eventos de pérdida que incluye pocos segmentos perdidos, el desempeño de ambas versiones es prácticamente el mismo; mientras que en casos donde la cantidad de pérdidas en un evento de pérdida es muy grande, la versión “impaciente” es más eficiente que la otra. Más adelante se revisarán con más detalle algunos aspectos de este trabajo.

En [51] se revisa cómo los procesos de re-ordenamiento y almacenamiento por lotes, realizados en redes “mix” (sistemas de comunicaciones anónimas), afectan al desempeño de TCP. El almacenamiento por lotes impone un tamaño máximo de la *cwnd* e incrementa el RTT. El re-ordenamiento provoca la emisión de más ACKs duplicados, que eventualmente pueden provocar FRs innecesarias. Los autores proponen una expresión para el cálculo del THR como: el valor esperado del cociente de la cantidad de datos enviados entre el tiempo transcurrido para ello. Esta definición del THP distingue a este trabajo de muchos basados en [32], pues en ellos el THR de un flujo TCP en estado estable se define como el cociente de los valores esperados de las variables: cantidad de datos enviados y tiempo transcurrido para ello. Para aislar el efecto de los procesos en redes “mix” suponen un escenario muy simple: sin pérdidas. Aunque el enfoque de este trabajo es distinto al del conjunto estudiado, se toma en cuenta como un antecedente en el tópico de modelado de TCP porque su propuesta para el cálculo del THR es alternativa a la usada en [32] y en otros trabajos.

En [52] se modifica el modelo PFTK para ser función del valor estimado del ancho de banda disponible en la red, en lugar de serlo, como originalmente esta propuesto, de la probabilidad de evento de pérdida y del valor estimado del RTT. Según los autores, la motivación de dicha modificación radica en la dificultad y poca precisión hallada en la estimación de estos parámetros de la red. La validación de sus resultados es a través de su comparación contra el THR medido en un experimento de laboratorio con una red LAN. Con esta propuesta se obtiene una mejor aproximación que con el modelo PFTK, y los autores consideran que esto se debe a la imprecisión de los parámetros estimados para éste.

En [53] los autores emplean una técnica de predicción del THR de TCP que emplea un

entrenamiento, el cual se basa en información histórica del THR y la asocia con algún(os) parámetro(s) de la red, tal(es) como: la tasa de pérdidas ( $L$ ), el retardo en la cola del nodo “cuello de botella” ( $Q$ ), el ancho de banda disponible y el tamaño del flujo. Los resultados muestran que los parámetros que más afectan la predicción del THR son  $L$  y  $Q$ , los cuales pueden ser medidos pasiva o activamente en la red. La predicción del THR obtenida con este método es poco precisa en los siguientes casos: para flujos de corta duración (por el carácter exponencial del crecimiento de la tasa de transmisión en la fase SS) y para escenarios con  $L$  alta.

Hasta nuestro mejor conocimiento, el modelo más reciente para TCP NewReno se presenta en [37]. Los autores desarrollan un modelo para el THR en estado estable de una conexión TCP NewReno de larga duración como función de: la probabilidad de evento de pérdida, la probabilidad de pérdida de un segmento dentro del evento de pérdida y el valor estimado del RTT. Se considera la versión “lento pero firme” de NewReno. Al igual que el modelo PFTK, el desarrollo está presentado en dos etapas: a) todas las pérdidas se detectan mediante TDs, y b) también existe la posibilidad de la detección de pérdidas a través de TOs. Suponiendo una serie de simplificaciones, los autores obtienen una expresión para el THR. Una diferencia fundamental con respecto a [32] es que en [37] el THR se refiere a la tasa efectiva de transmisión, mientras que en [32] el THR es el valor esperado de la tasa de transmisión de TCP. Ya que esta propuesta es básica para el desarrollo de este trabajo doctoral, en una sección siguiente se explicará con mayor detalle.

Actualmente, el análisis y modelado de TCP se ha diversificado y especializado en los innumerables ambientes, tipos de redes, escenarios, etc., en los que se usa este protocolo, así como en la comparación de nuevas variantes del mismo contra otras previamente existentes. En la mayoría de estos trabajos, para tener un grado de complejidad manejable, se intentan abstraer los aspectos característicos del problema de estudio y por ende normalmente se consideran sólo los mecanismos más representativos de TCP (SS, CA, TO), al igual que en [32], sin profundizar en tantos detalles. Algunos de los temas de actualidad en el área de análisis y modelado de TCP son: el impacto del tipo de cola y del esquema de administración de la cola en el desempeño de flujos TCP [57], el análisis de variantes TCP tipo pro-activas y su coexistencia con variantes reactivas [58], análisis de TCP en un tipo específico de redes, como redes mesh [59] y celulares [60], etc.

Este trabajo se concentra en el tema clásico de modelado de TCP en redes cableadas, y los modelos que se revisarán a detalle en las siguientes secciones son los que constituyen una

rama principal en el modelado de TCP NewReno.

## 2.2. El modelo PFTK.

En [32] se obtiene un modelo simple para el THR en estado estable de una conexión TCP Reno de larga duración. El desarrollo se presenta en tres etapas:

1. En la primera etapa no se considera el efecto del control de flujo, de forma que el tamaño de la *cwnd* no se ve limitado por la *rwnd* y se supone que las pérdidas son detectadas únicamente por TDs.
2. En la segunda etapa se amplía el análisis de la primera para considerar que las pérdidas también pueden ser detectadas mediante TOs.
3. En la tercera etapa se completa el análisis previo incorporando el mecanismo de control de flujo.

En esta sección se revisan con detalle las primeras dos de estas etapas. Los modelos revisados en las secciones siguientes y el desarrollo realizado en este trabajo doctoral excluye el efecto del control de flujo en el análisis de TCP NewReno, por lo que la explicación detallada de la tercera etapa del análisis en [32] no se abarca en esta tesis. La serie de suposiciones en las que se basa este modelo se mencionan a continuación.

### 2.2.1. Suposiciones generales.

1. Fuente infinita y continua de datos.
2. Se ignoran las fases de establecimiento de conexión y SS inicial.
3. El tiempo de transmisión de todos los segmentos permitidos por la *cwnd* es mucho menor que el RTT.
4. Segmentos de tamaño MSS.
5. Sólo ocurren pérdidas de segmentos en la ruta del transmisor al receptor. Es decir, los ACKs no se pierden.
6. No se modela el algoritmo Nagle ni los mecanismos para evitar el síndrome de ventana “tonta”.

7. No se consideran los retardos de procesamiento en los nodos ni el límite en la cola de salida en el nodo transmisor.
8. El número de secuencia del segmento, SN, se refiere al segmento en lugar de a su primer octeto de datos.
9. El tamaño de la *cwnd* se considera como un número entero de segmentos de tamaño MSS.
10. La dinámica de TCP se modela en rondas.
11. Cada ronda comienza con la transmisión en ráfaga (*back-to-back*) de los segmentos de la *cwnd* y termina con la recepción del primer RA que reconoce alguno de ellos.
12. Se supone que todos los ACKs de los segmentos de una misma ventana de transmisión se reciben en ráfaga al final de la ronda.
13. Cada ronda dura un RTT.
14. Como se desprecia la fase SS inicial, la primera ronda ocurre al inicio de la fase CA.
15. La dinámica de TCP se puede ver como la concatenación de ciclos estadísticamente idénticos, de manera que el THR se puede determinar a partir del análisis de uno sólo de esos ciclos. El ciclo  $i$  consiste en la evolución de la *cwnd*, desde un tamaño  $W_{i-1}/2$  hasta  $W_i$ , con una emisión de  $Y_i$  segmentos, y una duración  $A_i$ .
16. Cada ciclo está constituido por una o más fases. Una fase corresponde al mecanismo de TCP con el mismo nombre.
17. Un evento de pérdida (que se refiere a uno o más segmentos perdidos dentro de la misma ventana de transmisión) comienza con el primer segmento perdido en un ciclo.
18. La probabilidad de ocurrencia de un evento de pérdida,  $p$ , sigue una distribución de Bernoulli.
19. La probabilidad de evento de pérdida es independiente entre rondas.
20. La duración de cada ronda así como la probabilidad de un evento de pérdida son independientes del tamaño de la *cwnd*.
21. El tamaño de la *iwnd* es 1.

22. Sea  $N_t$  la cantidad de segmentos transmitidos en el intervalo  $[0, t]$ , se define al THR en estado estable del flujo TCP Reno de larga duración como el límite cuando  $t \rightarrow \infty$  del cociente  $N_t/t$ . Considerando que  $\{W_i\}_i$  es un proceso regenerativo con ganancias  $\{Y_i\}_i$ , se puede determinar el THR como el cociente del valor esperado de la cantidad de datos enviada en un ciclo,  $E[Y]$ , sobre el valor esperado de la duración del mismo,  $E[A]$ , es decir [54]:

$$\text{THR} = \lim_{t \rightarrow \infty} \frac{N_t}{t} = \frac{E[Y]}{E[A]} \quad (2.2)$$

Esta serie de suposiciones se retoman en varios trabajos dedicados a analizar este mismo problema, particularmente en aquellos revisados en esta tesis.

### 2.2.2. Suposiciones particulares del modelo.

Las suposiciones que no necesariamente aparecen en los siguientes trabajos revisados en esta tesis se consideran como particulares de este modelo.

1. No se consideran la fase FR de Reno.
2. Cuando se presenta un TO, no se considera la fase SS para la recuperación de las pérdidas, de manera que se modela como si del TO siguiera la fase CA.
3. Los eventos de pérdida son del tipo ráfaga, es decir, de los segmentos transmitidos en una ronda, a partir del primer segmento perdido, todos los demás también se consideran perdidos.
4. La detección de segmentos perdidos es a través de TDs o de TOs.
5. El tamaño de la  $cwnd$  puede verse limitado por la  $rwnd$  como efecto del control de flujo de TCP.
6. El receptor TCP puede generar ACKs retardados, donde la variable  $b$  representa la cantidad necesaria de segmentos que el receptor debe recibir adecuadamente antes de generar un ACK.
7. Los ciclos en la primera parte del modelo, llamados “periodos triple-duplicado”, (TDP: *Triple-Duplicate Period*), están separados por un TD y consisten únicamente en la fase CA. En la segunda parte, los ciclos consisten de una concatenación de tamaño arbitrario de ciclos TDP seguida de un ciclo TO (que consiste en una sucesión de TOs sucesivos).

### 2.2.3. Desarrollo del modelo: primera etapa.

Las variables usadas en el modelo son (ver figura 2.1):

- $Y_i$  es la cantidad de segmentos enviados en el ciclo  $i$ .
- $A_i$  es la duración del ciclo  $i$ .
- $\alpha_i$  es el primer segmento perdido en el ciclo  $i$ .
- $\beta_i$  es la cantidad de DAs recibidos al entrar a la FR del ciclo  $i$ .
- $\delta_i$  es el tamaño de la ráfaga de pérdidas en el ciclo  $i$ .
- $X_i$  es la duración, en rondas, de la fase CA en el ciclo  $i$ .
- $W_i$  es el tamaño de la  $cwnd$  al entrar a la FR en el ciclo  $i$ .
- $b$  es la cantidad necesaria de segmentos que el receptor espera para enviar un ACK.
- $p$  es la probabilidad de evento de pérdida.

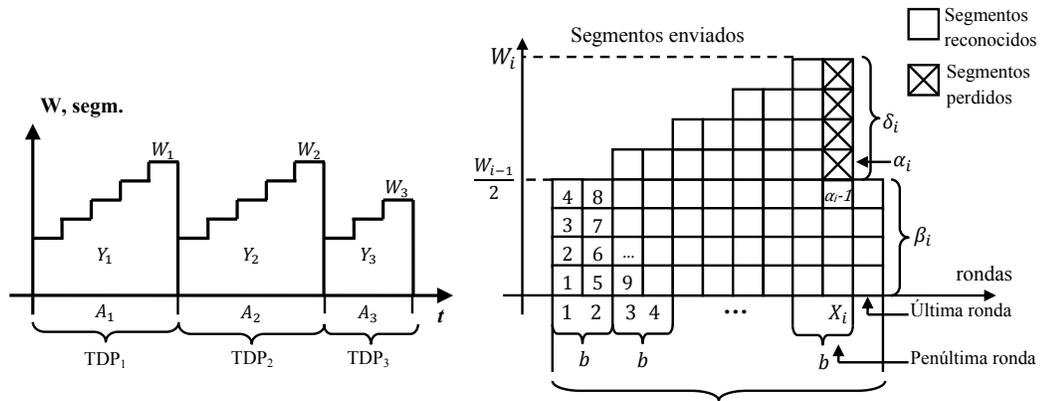


Figura 2.1: Dinámica de la  $cwnd$  en TCP Reno (despreciando la fase FR) [32].

Durante la CA, el tamaño de  $cwnd$  se incrementa en una fracción  $1/cwnd$  con cada RA recibido. Por ello, considerando que  $b = 2$ , la tasa de crecimiento del tamaño de la  $cwnd$  en la CA es de un segmento cada dos rondas. Por ejemplo, si el tamaño de la  $cwnd$  al inicio del primer ciclo es  $W_1 = 4$ , al inicio de la ronda 2 se reciben 2 RAs correspondientes a la transmisión exitosa de los segmentos  $\{1, 2, 3, 4\}$  y la ventana se actualiza a  $W_2 = (4 + 2/4)$  que corresponde a  $W_2 = 4$  (porque el tamaño de la ventana se considera entero). Al finalizar la ronda 2, se reciben otros 2 RAs correspondientes a la transmisión exitosa de los segmentos  $\{5, 6, 7, 8\}$

y la ventana se actualiza a  $W_3 = 5$ . Este proceso continúa en la CA hasta la detección de la primera pérdida  $\alpha_i$ , en la ronda  $X_i$ , en donde el tamaño de la ventana es  $W_i$ , como se muestra en la figura 2.2.

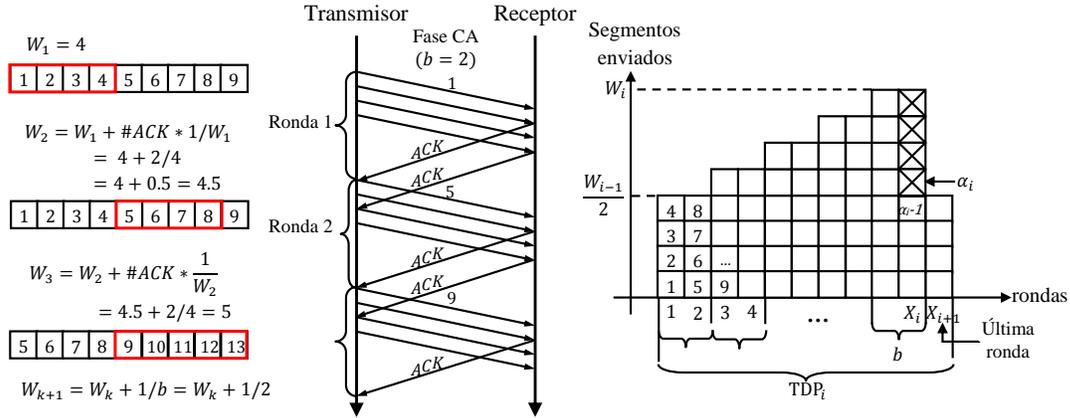


Figura 2.2: Crecimiento de la *cwnd* en la CA.

Debido a la llegada de los RAs correspondientes a los segmentos enviados exitosamente en la ronda  $X_i$ ,  $\beta_i$  segmentos adicionales se transmiten después del evento de pérdida, en la última ronda de la CA. Por ejemplo, en la figura 2.3, un evento de pérdida de tamaño  $\delta_i = 4$  se presenta en la ronda  $X_i$ . El primer segmento perdido es  $\alpha_i = 5$  de la ventana de tamaño  $W_i = 8$ . El receptor genera dos RAs correspondientes a los segmentos  $\{53, 54, 55, 56\}$ . El tamaño de la ventana del transmisor no cambia con estos RAs. En la ronda  $X_{i+1}$ , la ventana del transmisor se recorre para abarcar los  $\beta_i = 4$  segmentos  $\{61, 62, 63$  y  $64\}$ . El receptor responde a estos segmentos fuera de orden con  $\beta_i = 4$  DAs.

Así, el valor esperado de la cantidad de segmentos enviada durante la CA, y por lo tanto, en un ciclo TDP es:

$$E[Y] = E[\alpha] + E[W] - 1 = 1/p + E[W] - 1 \quad (2.3)$$

Y el valor esperado de la duración, en rondas, del ciclo TDP está dado por:

$$E[A] = (E[X] + 1)E[RTT] \quad (2.4)$$

Revisando la dinámica de crecimiento de la *cwnd* dentro de la CA se tiene que:

$$W_i = \frac{W_{i-1}}{2} + \frac{X_i}{b}; \quad i = 1, 2, \dots \quad (2.5)$$

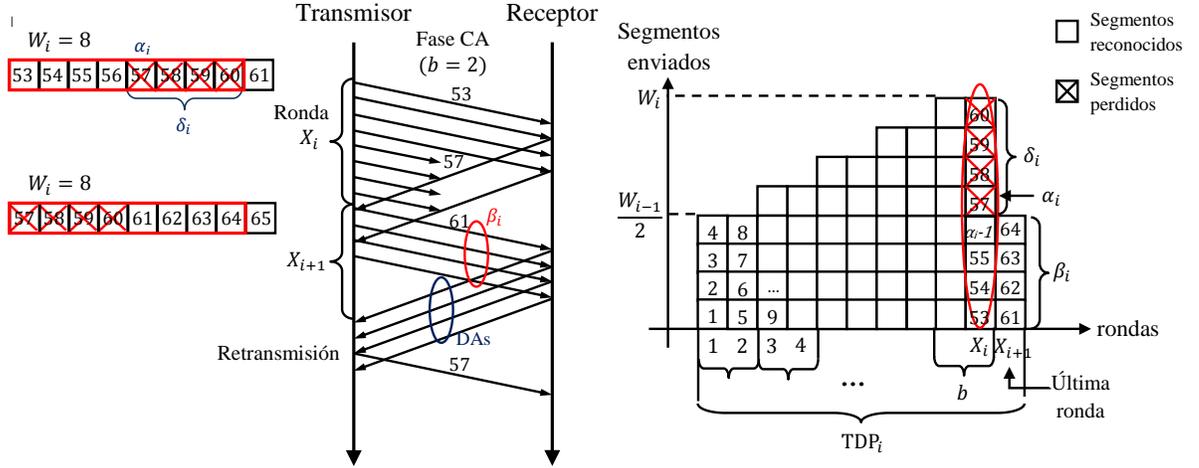


Figura 2.3: Crecimiento de la  $cwnd$  en la CA.

Y por lo tanto, una expresión equivalente para el valor esperado de la cantidad de segmentos enviada durante la CA es:

$$E[Y] = E\left[\sum_{k=0}^{X_i/b-1} \left(\frac{W_{i-1}}{2} + k\right)b + \beta_i\right] = \frac{E[X]}{2} \left(\frac{E[W]}{2} + E[W] - 1\right) + E[\beta] \quad (2.6)$$

Suponiendo que  $\{W_i\}$  y  $\{X_i\}$  son dos secuencias mutuamente independientes de variables aleatorias idénticamente distribuidas (VAs iid), se tiene que:

$$E[W] = \frac{2}{b} E[X] \quad (2.7)$$

Además, suponiendo que  $\beta_i$  está distribuido uniformemente entre 1 y  $W_i$ , se tiene que:

$$E[\beta] = \frac{E[W]}{2} \quad (2.8)$$

Igualando (2.3) con (2.6) se obtiene que:

$$E[W] = \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2} \quad (2.9)$$

Para valores pequeños de  $p$  y con  $b = 1$ , la simplificación de (2.9) equivale a (2.1).

Usando (2.4), (2.7) y (2.9) se obtiene que:

$$\text{THR} = \frac{\frac{1-p}{p} + E[W]}{E[A]} = \frac{\frac{1-p}{p} + \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2}}{E[\text{RTT}]\left(\frac{2+b}{6} + 1 + \sqrt{\frac{2b(1-p)}{3p} + \left(\frac{2+b}{6}\right)^2}\right)} \quad (2.10)$$

#### 2.2.4. Desarrollo del modelo: segunda etapa.

Se considera como ciclo una sucesión de tamaño indeterminado de ciclos TDP seguida de un ciclo TO. El ciclo TO consta de una serie de TOs sucesivos.

Las variables adicionales usadas en esta etapa son (ver figura 2.4):

- $n_i$  es la cantidad de ciclos TDP en el ciclo  $i$ .
- $Z_i^{\text{TD}}$  es la duración de la sucesión de ciclos TDP, de tamaño  $n_i$ , en el ciclo  $i$ .
- $Z_i^{\text{TO}}$  es la duración del ciclo TO en el ciclo  $i$ .
- $T_o$  es el valor esperado de la RTO en el momento en que comienza un ciclo TO.
- $S_i$  es la duración del ciclo  $i$ .
- $R_i$  es la cantidad de segmentos enviada en el ciclo TO dentro del ciclo  $i$ .
- $M_i$  es la cantidad de segmentos enviada en el ciclo  $i$ .
- $Y_{ij}$  es la cantidad de segmentos enviados en el ciclo TDP  $j$  del ciclo  $i$ .
- $A_{ij}$  es la duración del ciclo TDP  $j$  del ciclo  $i$ .
- $Q$  es la probabilidad de que una pérdida se detecte por un TO.

Se define el THR como:

$$\text{THR} = \frac{E[M]}{E[S]} \quad (2.11)$$

Donde:

$$E[S] = E[Z^{\text{TD}}] + E[Z^{\text{TO}}] = E\left[\sum_{j=1}^{n_i} A_{ij}\right] + E[Z^{\text{TO}}] \quad (2.12)$$

$$E[M] = E\left[\sum_{j=1}^{n_i} Y_{ij}\right] + E[R] \quad (2.13)$$

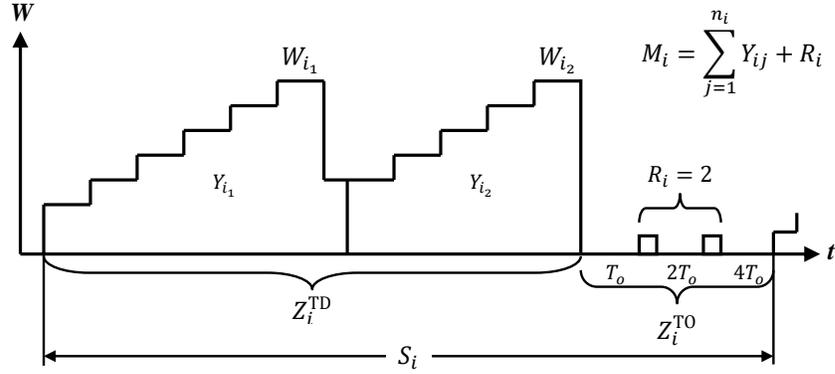


Figura 2.4: Dinámica de la *cwnd* en TCP Reno (despreciando la fase FR) y considerando que las pérdidas se detectan tanto por TDs como por TOs.

Después de una pérdida detectada por un TO, si la retransmisión se pierde entonces se produce un TO sucesivo. Por lo tanto, una secuencia de  $k$  TOs consecutivos se produce si existen  $k - 1$  retransmisiones fallidas seguidas de una retransmisión exitosa. En consecuencia, la cantidad de segmentos enviados en el ciclo TO,  $R$ , correspondientes a puras retransmisiones, tiene una distribución geométrica con probabilidad de éxito  $p$ , es decir:

$$P[R = k] = p^{k-1}(1 - p) \quad (2.14)$$

y su valor esperado está dado por:

$$E[R] = \frac{1}{1 - p} \quad (2.15)$$

Suponiendo que  $\{n_i\}_i$  es una secuencia de VAs iid, independiente de  $\{Y_{ij}\}$  y  $\{A_{ij}\}$ , se tiene que (2.12) y (2.13) se convierten en:

$$E[S] = E[n]E[A] + E[Z^{TO}] \quad (2.16)$$

$$E[M] = E[n]E[Y] + E[R] \quad (2.17)$$

Recordando que un ciclo consiste en  $n_i$  TDPs y un TO, la probabilidad de que exista una detección de pérdida por un TO en un ciclo,  $Q$ , es:

$$Q = \frac{1}{E[n]} \quad (2.18)$$

Con lo que, a partir de (2.11), (2.16), (2.17) y (2.18) se obtiene:

$$\text{THR} = \frac{E[Y] + QE[R]}{E[A] + QE[Z^{T_0}]} \quad (2.19)$$

Donde  $E[Y]$ ,  $E[A]$  y  $E[R]$  están definidas en (2.3), (2.4) y (2.15), respectivamente.

Considerando el mecanismo de *back-off*, con un límite de crecimiento de la RTO de hasta 64 veces la RTO inicial,  $T_0$ , se tiene que:

$$\begin{aligned} E[Z^{T_0}] &= \sum_{k=1}^6 (2^k - 1)T_0 P[R = k] + \sum_{k=7}^{\infty} (63 + 64(k - 6))T_0 P[R = k] \\ &= T_0 \frac{1+p+2p^2+4p^3+8p^4+16p^5+32p^6}{1-p} \end{aligned} \quad (2.20)$$

Por último, para obtener la probabilidad  $Q$  se considera lo siguiente:

$Q$  es la probabilidad de que se presente un TO estando en la CA. Suponiendo que se presenta un evento de pérdida en la ronda  $i$ , las tres únicas condiciones para que éste se detecte mediante un TO son:

- Que el tamaño  $w$  de la *cwnd* en la ronda  $i$  sea menor o igual que tres,  $w \leq 3$ , ya que entonces no se podrían generar los tres DAs que constituyan el TD, puesto que por lo menos un segmento se perdió.
- Por otro lado, si  $w > 3$ , se requiere que:
  - El tamaño de la ráfaga de pérdidas sea  $\delta_i > w - 3$ , de manera que la cantidad de segmentos transmitidos en la ronda  $i + 1$  sea  $\beta_i \leq 2$  y, por lo tanto, no haya DAs suficientes para generar un TD.
  - Si  $\delta_i \leq w - 3$ , una nueva ráfaga de pérdidas de tamaño mayor o igual que  $\beta_i - 2$  se presente en la ronda  $i + 1$ .

Como la probabilidad de que, en la ronda donde se presenta el evento de pérdidas, los primeros  $k$  segmentos de la ventana de tamaño  $w$  sean exitosos, se obtiene como:

$$A(w, k) = \frac{(1-p)^k p}{1 - (1-p)^w} \quad (2.21)$$

Y la probabilidad de que, de una secuencia de  $n$  segmentos transmitidos se pierdan todos los segmentos a partir del  $m + 1$  es:

$$C(n, m) = \begin{cases} (1-p)^m p & ; m \leq n-1 \\ (1-p)^m & ; m = n \end{cases} \quad (2.22)$$

Usando (2.21) y (2.22) se tiene que la probabilidad de tener un TO en una ronda con ventana de tamaño  $w$  es:

$$\hat{Q}(w) = \begin{cases} 1 & ; w \leq 3 \\ \sum_{k=0}^2 A(w, k) + \sum_{k=3}^w A(w, k) \sum_{m=0}^2 C(k, m) & ; \text{otro caso} \end{cases} \quad (2.23)$$

Haciendo la siguiente aproximación:

$$Q \approx \hat{Q}(E[W]) \quad (2.24)$$

se tienen ya todas las expresiones necesarias para obtener el THR mediante (2.19):

$$\text{THR} = \frac{\frac{1-p}{p} + E[W] + Q\left(\frac{1}{1-p}\right)}{E[\text{RTT}](E[X] + 1) + QT_o\left(\frac{1+p+2p^2+4p^3+8p^4+16p^5+32p^6}{1-p}\right)} \quad (2.25)$$

donde  $E[W]$  está definida en (2.9) y  $E[X]$  se obtiene con (2.7) y (2.9). De esta manera, el modelo para el THR en estado estable de TCP Reno se obtiene en [32].

El modelo PFTK se ha convertido en uno de los trabajos fundamentales en esta área. Una gran cantidad de modelos para TCP en sus distintas variantes han sido desarrollados tomándolo como base [33 - 34] [36 - 37] [43 - 44] [48 - 50] [52]. Esto pese a que el modelo PFTK recurre a una serie de simplificaciones importantes, principalmente aquellas que excluyen del modelo los algoritmos de las fases SS y FR, lo que resulta, como se ha demostrado en otros trabajos, en una sobre-estimación del THR [36] [37] [44].

### **2.3. Modelos simplificados para TCP NewReno “lento pero firme” e “impaciente” desarrollados en [48], [49] y [50].**

Estas propuestas toman como base al modelo PFTK e incluyen en el análisis el mecanismo FR de NewReno en sus dos variantes: “lento pero firme” e “impaciente”. Todas las supo-

siciones generales bajo las que el modelo PFTK se desarrolla y que están mencionadas en la sección 2.2.1 de esta tesis, se conservan en estos trabajos. Adicionalmente, estos modelos consideran otro conjunto de suposiciones particulares a cada uno de ellos. Ya que son estudios enfocados a un mismo problema, es interesante la revisión de las suposiciones particulares que presentan en común y de aquellas en las que difieren.

### **2.3.1. Suposiciones particulares comunes en los modelos.**

De entre las suposiciones bajo las cuales se desarrollan los modelos en [48], [49] y [50], aquellas comunes en todos ellos son:

- Los eventos de pérdida son de tipo ráfaga, es decir, de los segmentos transmitidos en una ventana de tamaño  $W$ , a partir del primer segmento perdido, todos los demás se considera que también se pierden.
- La variable  $m$  en [48] ( $\delta$  en [49] y [50]) representa el valor esperado de la cantidad de segmentos perdidos en un evento de pérdida (es decir, del tamaño de la ráfaga de pérdidas).
- Todos los eventos de pérdida se detectan mediante un TD (es decir,  $m < W - 2$ , donde  $W$  es el valor esperado del tamaño de la *cwnd* al inicio de la FR).
- Las retransmisiones siempre son exitosas.
- Los nuevos segmentos transmitidos dentro de la FR no se pierden.
- La recuperación de los segmentos perdidos, activada a partir de un TO, supone un esquema de *Go-Back-N*.

### **2.3.2. Suposiciones particulares del modelo presentado en [48].**

- No toma en cuenta el efecto del control de flujo.
- El receptor TCP genera un ACK por cada segmento recibido.
- Cada ciclo consiste en las fases CA - FR, para “lento pero firme” y en CA - FR - TO - SS, para “impaciente”.

### 2.3.3. Suposiciones particulares de los modelos desarrollados en [49] y [50].

- El tamaño de la *rwnd* puede ser una limitante para el tamaño de la *cwnd*, por efecto del control de flujo.
- El receptor TCP puede generar ACKs retardados, donde la variable *b* representa la cantidad necesaria de segmentos que el receptor debe recibir antes de generar un ACK.
- En [49] cada ciclo consiste en las fases FR - CA.
- En [50], cada ciclo consiste en las fases FR - CA cuando el tamaño de la ráfaga de segmentos perdidos es menor o igual que la relación  $RTO/RTT$ , y en FR - SS - CA en caso contrario.
- En el caso de una FR que no termina con la recuperación de las pérdidas, es decir, cuando la FR concluye con un TO, se considera que los datos transmitidos durante la FR se retransmiten en la fase SS posterior al TO. Por ello, en este caso, la fase FR queda excluida del modelo.

### 2.3.4. Desarrollo del modelo reportado en [48].

En [48] se analizan las versiones de TCP NewReno “lento pero firme” e “impaciente”. La versión de TCP “lento pero firme” supone el restablecimiento del temporizador con cada PA que se recibe dentro de la FR. Por su lado, TCP “impaciente” sólo lo restablece con la llegada del primer PA.

La diferencia entre ellas se hace significativa en escenarios con una gran cantidad de segmentos perdidos dentro de la misma ventana de transmisión, puesto que, mientras que con TCP “impaciente” seguramente se terminará por activar el TO para recuperar el total de las pérdidas, con el TCP “lento pero firme” la FR durará tantos RTTs como sea el número de segmentos perdidos (figura 2.5). Según sea la relación del tamaño de la ráfaga de errores respecto del cociente  $RTO/RTT$ , en algunas ocasiones el desempeño de una de estas versiones será mejor que el de la otra.

Las variables utilizadas en este modelo son:

- *W* es el valor esperado del tamaño de la *cwnd* al entrar a la FR en cada ciclo.
- *m* es el valor esperado de la cantidad de segmentos perdidos en un evento de pérdida.

- $\alpha$  es el primer segmento perdido en el ciclo.
- $S$  es el valor esperado de la cantidad de segmentos enviados en cada ciclo.
- $D$  es el valor esperado de la duración de cada ciclo.
- $R$  es el valor esperado del RTT.
- $p$  es la probabilidad de evento de pérdida.

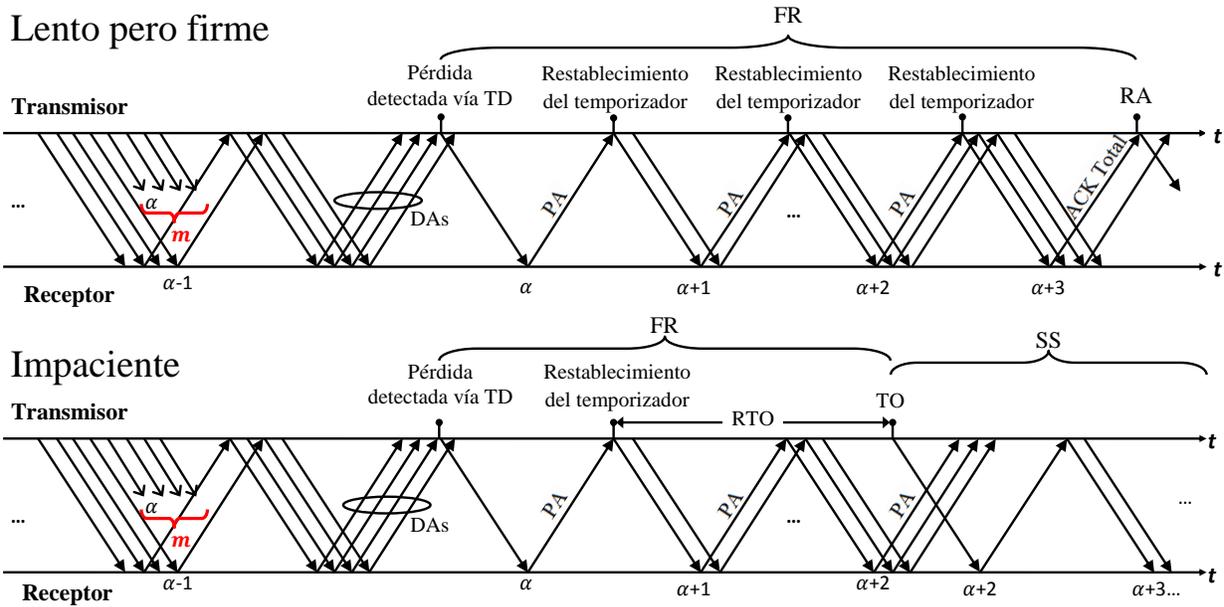


Figura 2.5: TCP NewReno “lento pero firme” vs. TCP NewReno “impaciente”.

En TCP NewReno “lento pero firme” la FR dura  $m$  RTTs y se sale de la FR hacia la CA con  $cwnd = W/2$ . La dinámica de TCP NewReno “lento pero firme”, bajo las suposiciones expuestas, se traduce en ciclos CA - FR, como se muestra en la figura 2.6. Durante la CA, la  $cwnd$  crece desde  $W/2$  hasta  $W$ , con tasa de un segmento por RTT. Al presentarse un TD, la fase CA termina y la FR comienza. Durante la FR, se aproxima la cantidad de nuevos segmentos enviada en cada ronda como  $W/4$ <sup>1</sup>. De lo anterior, el valor esperado de la cantidad de segmentos enviados en cada ciclo,  $S$ , queda expresado como:

$$S = \sum_{i=\frac{W}{2}}^W i + m \frac{W}{4} = \frac{W}{4} \left( 3 \frac{W}{2} + m + 3 \right) \quad (2.26)$$

<sup>1</sup>En [49] se mejora esta aproximación. Sin embargo, el análisis en [49] sigue sin considerar la posibilidad de que haya pérdida de segmentos nuevos transmitidos durante la FR. En el análisis desarrollado en esta tesis se obtiene una expresión que corrige ambas aproximaciones.

Considerando, además, que el valor esperado de la cantidad de segmentos enviada en un ciclo CA - FR equivale a  $1/p$ , e igualándola con (2.26), se obtiene la siguiente expresión del valor esperado de  $W$  como función de  $p$ :

$$W = \frac{-3p - mp + \sqrt{9p^2 + 6mp^2 + m^2p^2 + 24p}}{3p} \quad (2.27)$$

Por otro lado, los autores derivan la expresión para el valor esperado de la latencia del ciclo como función de  $W$  como:

$$D = \left(\frac{W}{2} + 1\right)R + mR \quad (2.28)$$

De forma que el THR en estado estable de un flujo TCP “lento pero firme” de larga duración se puede calcular como:

$$\text{THR} = \frac{S}{D} = \frac{\frac{1}{p}}{R\left(\frac{W}{2} + m + 1\right)} \quad (2.29)$$

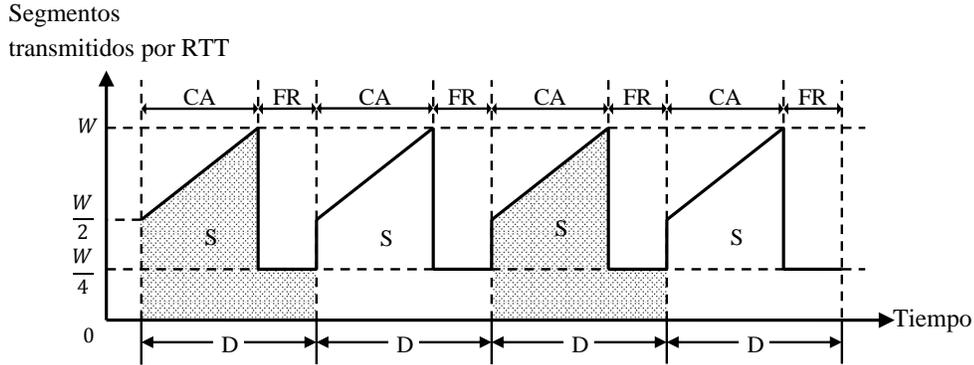


Figura 2.6: Dinámica de la  $cwnd$  en TCP NewReno “lento pero firme” [48].

En TCP NewReno “impaciente”, si  $m < (RTO/RTT+1)$  entonces la FR dura  $m$  RTTs, y si  $m \geq (RTO/RTT+1)$  entonces tiene una duración de  $RTT+RTO$ . En este último caso, la FR de NewReno “impaciente” termina con un TO. Bajo las suposiciones expuestas, la dinámica de TCP NewReno “impaciente” se traduce en ciclos CA - FR - TO - SS, en los que se puede despreñar la fase FR por su corta duración (figura 2.7).

Al entrar a la FR, el  $ssthresh$  se actualiza a  $W/2$ . Cuando el TO ocurre, el  $ssthresh$  se recalcula quedando como  $W/4$ . Así, durante el SS, la  $cwnd$  crece desde  $iwnd$  hasta  $W/4$ , duplicando su tamaño en cada ronda. Ya en la CA, la  $cwnd$  crece desde  $W/4$  hasta  $W$ , con una tasa de un

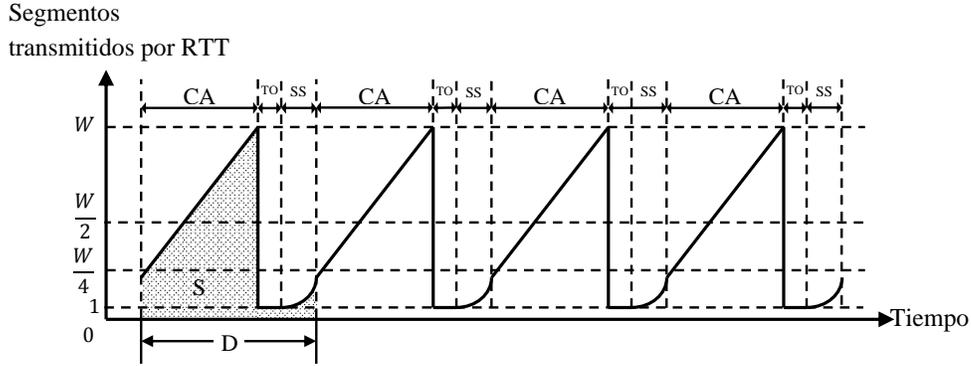


Figura 2.7: Dinámica de la *cwnd* en TCP NewReno “impaciente” [48].

segmento por RTT.

La fase FR se excluye del modelo debido a dos razones: su corta duración y la suposición de que los nuevos datos transmitidos durante la FR de cualquier manera se terminarán retransmitiendo en la fase SS posterior al TO.

Siguiendo el mismo procedimiento que para el modelo de TCP NewReno “lento pero firme”, los autores obtienen la expresión para el THR de la siguiente forma: primero, la expresión del valor esperado de la cantidad de segmentos enviados en el ciclo queda como <sup>2</sup>:

$$S = \left(1 + 2 + \dots + \frac{W}{8}\right) + \sum_{i=\frac{W}{4}}^W i \quad (2.30)$$

Recordando que el valor esperado de la cantidad de segmentos enviados en el ciclo equivale a  $1/p$ , e igualando esta expresión con (2.30), se obtiene el valor esperado de  $W$  como función de  $p$ :

$$W = \frac{-14p + 2\sqrt{169p^2 + 120p}}{15p} \quad (2.31)$$

Por otro lado, la expresión para el valor esperado de la latencia del ciclo se deriva como función de  $W$ :

$$D = \left(\log \frac{W}{8} + 1\right)R + \left(\frac{3W}{4} + 1\right)R + (R + \text{RTO}) \quad (2.32)$$

De forma que, usando la definición de  $W$  en (2.31), el THR queda expresado como:

<sup>2</sup>Por sencillez, la última ronda del SS se considera como parte de la CA siguiente.

$$\text{THR} = \frac{S}{D} = \frac{\frac{1}{p}}{\left(\log \frac{W}{8} + 1\right)R + \left(\frac{3W}{4} + 1\right)R + (R + RTO)} \quad (2.33)$$

En [48] concluyen que la variante TCP NewReno “lento pero firme” supera en desempeño a NewReno “impaciente” en la mayoría de los escenarios.

### 2.3.5. Desarrollo de los modelos presentados en [49] y [50].

En [49] y [50] se obtienen los modelos analíticos para el THR en estado estable de TCP NewReno en sus versiones “impaciente” y “lento pero firme”, respectivamente. La principal diferencia con el análisis en [48] es que se introduce un modelado con una mejor aproximación de la cantidad de segmentos enviada dentro de la FR. Adicionalmente, se incluyen otros parámetros que hacen el modelo más general: el factor  $b$ , que representa el número de segmentos que el receptor espera para enviar un ACK ( $b = 1$  si no hay ACK retardado y  $b = 2$  si lo hay) y el tamaño máximo de la ventana  $W_{\max}$ , que está determinado por la  $rwnd$  como consecuencia del mecanismo de control de flujo.

En [49] y [50] se usa la misma nomenclatura que en [32], a excepción de la variable  $\tau$ , definida como:

$$\tau = \frac{E[RTO]}{E[RTT]} \quad (2.34)$$

Se establece que los ciclos se separan por un TD. En TCP “lento pero firme”, cada ciclo consiste en las fases FR - CA, como se muestra en la figura 2.8. En TCP “impaciente”, si  $E[\delta] < \tau + 1$ , entonces no hay diferencia respecto de TCP “lento pero firme”, pero si  $E[\delta] \geq \tau + 1$ , cada ciclo consta de las fases FR - SS - CA (figura 2.9).

Suponiendo que los nuevos segmentos transmitidos dentro de la FR no son sujetos de perderse, se modela la cantidad de segmentos transmitidos durante cada ronda de la FR,  $k = 1, \dots, k_{\max}$ , en el ciclo  $i$  como:

$$\text{máx} \left( 0, \frac{W_{i-1}}{2} - \delta_{i-1} + k - 1 \right) \quad ; 1 \leq k \leq k_{\max} \quad (2.35)$$

donde  $k_{\max} = \tau + 1$  para “impaciente” y  $k_{\max} = \delta_{i-1}$  para “lento pero firme”. Descartando el efecto del control de flujo, el desarrollo planteado en [49] y [50] es el siguiente:

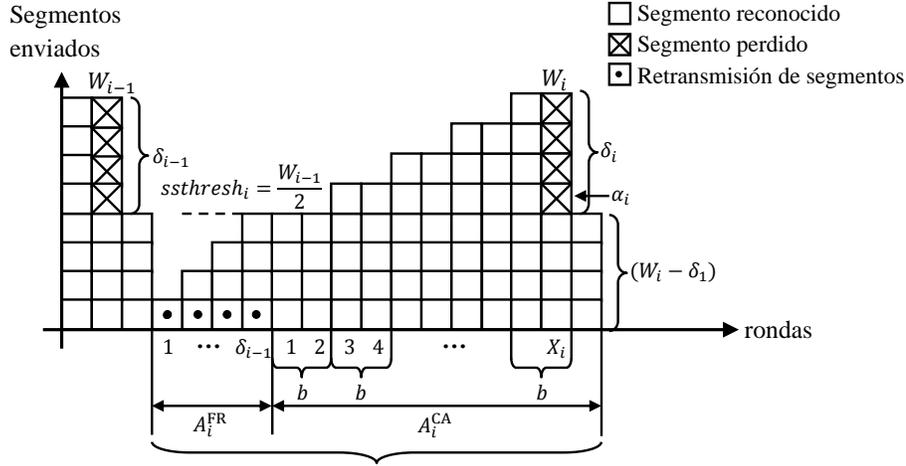


Figura 2.8: Dinámica de la  $cwnd$  en TCP NewReno “lento pero firme” [49].

El valor esperado de la duración de la FR en cada ciclo es:

$$E[A^{FR}] = E[k_{\max}]E[RTT] \quad (2.36)$$

Mientras que el valor esperado de la cantidad de segmentos enviados durante la FR en cada ciclo es:

$$E[Y^{FR}] = E\left[k_{\max} + \sum_{k=1}^{k_{\max}} \max\left(0, \frac{W_{i-1}}{2} - \delta_{i-1} + k - 1\right)\right] \quad (2.37)$$

Para el caso de TCP “impaciente”, si  $E[\delta] \geq \tau + 1$  entonces se considera que el  $ssthresh$  al salir de la FR es casi el mismo que al entrar a esta fase, es decir, se supone  $ssthresh = W/2$ . La FR termina con un TO, después del cual comienza la fase SS. Dentro del SS, la cantidad de segmentos enviada en  $N$  rondas es:

$$Y^{SS} = 1 + 2 + \dots + 2^{N-1} = 2^N - 1 \quad (2.38)$$

La fase SS termina en la ronda en la que el tamaño de la  $cwnd$  es  $W/2$ . Considerando la última ronda del SS como ya dentro de la CA, entonces, el valor esperado de la duración del SS, medida en RTTs, está dada por:

$$E[A^{SS}] = \log_2(E[W]) - 1 \quad (2.39)$$

Mientras que el valor esperado de la cantidad de segmentos enviados en el SS es:

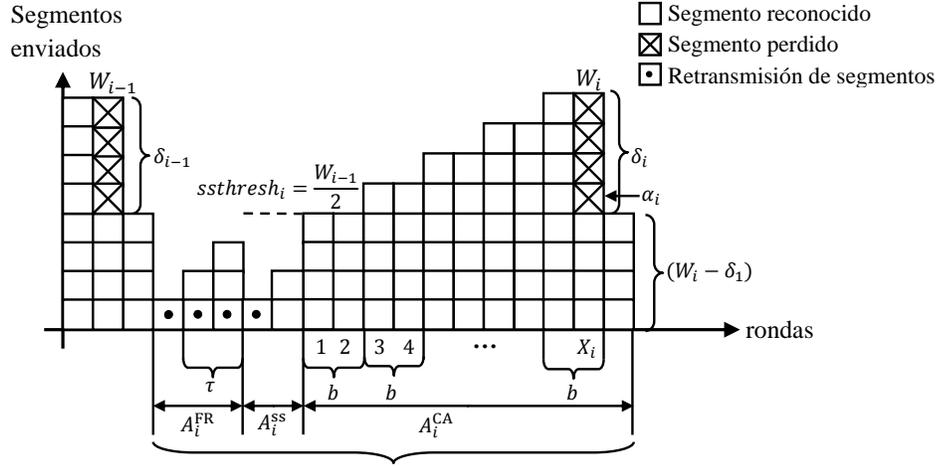


Figura 2.9: Dinámica de la  $cwnd$  en TCP NewReno “impaciente” [49].

$$E[Y^{SS}] = 2^{E[A^{SS}]} - 1 = \frac{E[W]}{2} - 1 \quad (2.40)$$

Las expresiones para la fase CA son iguales para ambas versiones. El valor esperado de su duración (medido en RTTs) es:

$$E[A^{CA}] = E[X + 1] = b \left( \frac{E[W]}{2} + 1 \right) + 1 \quad (2.41)$$

Mientras que el valor esperado de la cantidad de segmentos enviados en esa fase es:

$$E[Y^{CA}] = 3b \frac{E[W]}{4} \left( \frac{E[W]}{2} + 1 \right) + E[W] - E[\delta] \quad (2.42)$$

Finalmente, para ambas versiones, la expresión del valor esperado de la cantidad de segmentos enviados en cada ciclo se puede obtener como:

$$E[Y] = E[Y^{FR}] + E[Y^{SS}] + E[Y^{CA}] \quad (2.43)$$

donde  $E[Y^{SS}] = 0$  para NewReno “lento pero firme”. Alternativamente, también para ambas versiones se tiene la siguiente expresión:

$$E[Y] = E[\alpha] + E[W] - 1 \quad (2.44)$$

con  $E[\alpha] = 1/p$ , como se explica en [32]. Igualando (2.43) con (2.44), se puede despejar  $E[W]$  y usarla en la definición del THR, la cual, a partir de (2.36), (2.39) y (2.41) queda como:

$$\text{THR} = \frac{1/p + E[W] - 1}{E[\text{RTT}] \left( \tau + 1 + \log_2(E[W]) + b \left( \frac{E[W]}{2} + 1 \right) \right)} \quad ; \text{ "impaciente" } \quad (2.45)$$

$$\text{THR} = \frac{\frac{1-p}{p} + E[W]}{E[\text{RTT}] \left( E[\delta] + b \left( \frac{E[W]}{2} + 1 \right) + 1 \right)} \quad ; \text{ "lento pero firme" } \quad (2.46)$$

A diferencia de [48], en donde los autores concluyen que la variante TCP NewReno “lento pero firme” supera en desempeño a NewReno “impaciente” en la mayoría de los escenarios, en [49] se concluye que ninguna de las soluciones es óptima, operando más eficientemente una en algunos casos y la segunda en otros, lo que coincide con la conclusión de [14].

### 2.3.6. Conclusión de los trabajos reportados en [48], [49] y [50].

La principal contribución de estos trabajos es que son los primeros que modelan explícitamente el mecanismo FR de NewReno. Sin embargo, todos ellos recurren a varias simplificaciones, de entre las cuales principalmente se tiene que:

- Se enfocan a flujos de larga duración, excluyendo del modelo a la gran cantidad de flujos de corta duración que existen en Internet.
- Se considera que no hay pérdidas durante las fases SS, lo cual no tiene una justificación válida bajo la suposición de que los eventos de pérdida son iid entre rondas.
- No se toma en cuenta la detección de pérdidas mediante TOs, lo cual, como se puntualiza en [32], no refleja la realidad.
- Se considera que no hay pérdidas durante la FR, lo que al igual que el razonamiento hecho para la fase SS, no coincide con la definición de la probabilidad de los eventos de pérdida.
- Todas las retransmisiones se suponen exitosas, cuya validez se cuestiona igual que en el punto anterior, pues toda retransmisión está separada de la transmisión original por lo menos por un RTT.
- En el caso de una FR que no concluye con la recuperación de las pérdidas y termina con un TO, se considera que todos los segmentos a partir del segmento asociado con el

TO se retransmiten. Aunque, en principio, así funciona el mecanismo de recuperación de pérdidas detectadas mediante un TO, se puede verificar por simulación que, debido a la capacidad del receptor de almacenar segmentos fuera de orden y al mecanismo de ACK acumulado, comúnmente no se terminan retransmitiendo todos estos segmentos.

## **2.4. Modelo para TCP NewReno propuesto en [37].**

En [37] se desarrolla un modelo del THR en estado estable de un flujo TCP NewReno de larga duración basado en [32]. Su principal aportación es que modela el algoritmo FR de NewReno (versión “lento pero firme”) de una forma más precisa que los modelos en [48], [49] y [50].

La secuencia del desarrollo realizado en [37] está en dos etapas:

1. El análisis del THR cuando las pérdidas son detectadas únicamente por TDs.
2. El análisis del THR cuando las pérdidas son detectadas tanto por TDs como por TOs.

### **2.4.1. Suposiciones generales del modelo presentado en [37].**

Usa las suposiciones generales del modelo PFTK, listadas en la sección 2.2.1, a excepción de la 22, la cual se modifica como sigue: en [32] se considera al THR como la tasa de transmisión de TCP (definida como: el cociente del valor esperado de la cantidad de segmentos enviados sobre el valor esperado del tiempo transcurrido para ello), mientras que en [37] el THR se refiere a la tasa efectiva de transmisión (que se define como: el cociente del valor esperado de la cantidad de segmentos enviados exitosamente, sobre el valor esperado del tiempo transcurrido para ello).

### **2.4.2. Suposiciones particulares del modelo reportado en [37].**

1. En cada ciclo, el evento de pérdida (LE: *Loss Event*) comienza a partir del primer segmento perdido y dura un RTT.
2. Se define como ventana de pérdida (LW: *Loss Window*) al conjunto de segmentos enviados dentro del LE, es decir, abarca *cwnd* segmentos a partir del primer segmento perdido –inclusive.

3. Así, un LE involucra a todos los demás segmentos perdidos en la LW.
4. Si existe un LE en alguna ronda de la FR, ésta involucra forzosamente la pérdida de la retransmisión. Es decir, cualquier pérdida dentro de la FR se traduce en un TO. En este caso, las transmisiones hechas dentro de la FR no se consideran en el cálculo de los segmentos del flujo TCP.
5. El mecanismo de FR opera en su variante "lento pero firme".
6. El receptor TCP genera un ACK por cada segmento recibido, es decir, no se considera el mecanismo del ACK retardado.
7. La detección de segmentos perdidos es a través de TDs o de TOs.
8. El tamaño de la *cwnd* no se ve limitado por la *rwnd*, es decir, no se considera el efecto del mecanismo de control de flujo de TCP.
9. La recuperación de los segmentos perdidos activada partir de un TO supone un esquema de *Go-Back-N*, en el cual se retransmiten todos los segmentos a partir del segmento asociado con el TO y hasta el último segmento que se había transmitido hasta el momento del TO.
10. Se consideran ciclos CA - FR en la primera parte del modelo. En la segunda parte, los ciclos consisten de una concatenación de tamaño arbitrario de ciclos CA - FR seguidos de un ciclo CA - TO - SS, el cual se inicia con la fase CA, seguida de un ciclo TO (sucesión de TOs consecutivos) y concluye con la fase SS.
11. La duración de las rondas es de un RTT, excepto por la ronda en donde se genera el LE, cuya duración puede ser menor.
12. Mientras que los LEs suceden de manera independiente entre ellos con una probabilidad  $p$  (con distribución Bernoulli), los segmentos perdidos dentro de la misma LW suceden de forma independientemente con probabilidad  $q$ , que es una VA con distribución Bernoulli.
13. En general  $p \neq q$ .

### **2.4.3. Desarrollo de la primera etapa del modelo presentado en [37].**

Las variables usadas en este modelo son:

- $S_{CAFR}$  es el valor esperado de la cantidad de segmentos enviados exitosamente en el ciclo CA - FR (en la figura 2.10, desde la ronda uno hasta la siete –se incluyen también los segmentos perdidos porque finalmente se van a recuperar en la FR).
- $S_{CA}$  es el valor esperado de la cantidad de segmentos enviados exitosamente durante la fase CA.
- $S_{\beta}$  es el valor esperado de la cantidad de segmentos enviados en la ronda posterior al LE (en la figura 2.10, la ronda ocho), como respuesta a los segmentos exitosos de la ronda donde sucede el LE.
- $S_{FR}$  es el valor esperado de la cantidad de segmentos enviados exitosamente durante la fase FR, sin incluir las retransmisiones.
- $D_{CAFR}$  es el valor esperado de la duración del ciclo CA - FR.
- $D_{CA}$  es el valor esperado de la duración de la fase CA.
- $D_{\beta}$  es el valor esperado de la duración de la ronda posterior al LE, que termina con la indicación de la pérdida TD y puede ser menor que RTT.
- $D_{FR}$  es el valor esperado de la duración de la fase FR.
- $R$  es el valor esperado del RTT.
- $W$  es el valor esperado del tamaño de la *cwnd* al entrar a la FR en el ciclo CA - FR.
- $\alpha$  es el valor esperado de la cantidad de segmentos enviados exitosamente entre la última pérdida de un LE y el comienzo del LE siguiente (entre D y J, sin incluirlos, en el caso de la figura 2.10). La definición de esta variable es distinta de la encontrada en [32], [48], [49] y [50].
- $\delta$  es el valor esperado de la separación, medida en segmentos, entre la primera y la última pérdida en un mismo LE (que en la figura 2.10 está entre J y L). La definición de esta variable es distinta de la encontrada en [32], [48], [49] y [50].
- $m$  es la cantidad de segmentos perdidos uniformemente en la LW.

La relación entre el LE y la LW se muestra en la figura (2.11). Un LE comienza con el primer segmento perdido en un ciclo. Cuando hay un segmento perdido, el límite izquierdo de la *cwnd* se recorre hasta éste, formando la LW. Si este segmento perdido no es el primer segmento de una ronda, la transmisión de los últimos segmentos de la LW será después de recibir

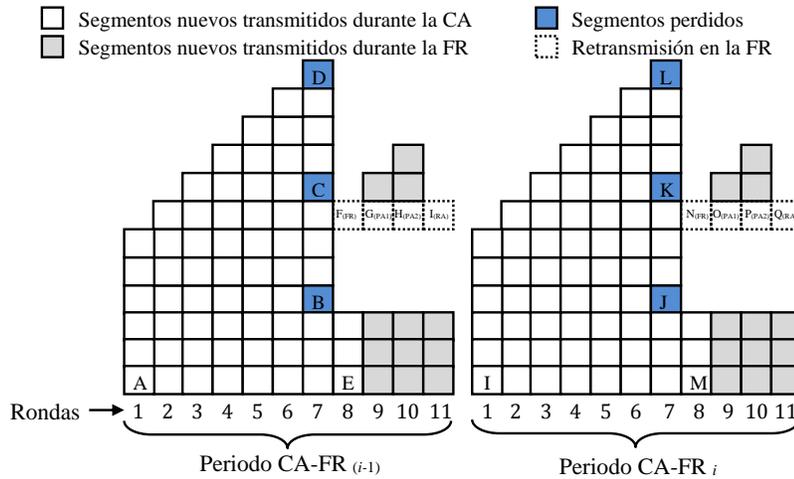


Figura 2.10: Dinámica de transmisión en NewReno [37].

los RAs de los segmentos previos a la primera pérdida, los cuales tardan en llegar un RTT, por lo que se considera que la duración del LE es de un RTT. Siendo esta la definición de un LE, entonces se puede concluir que toda pérdida detectada por un TD o por un TO indica el inicio de un LE.

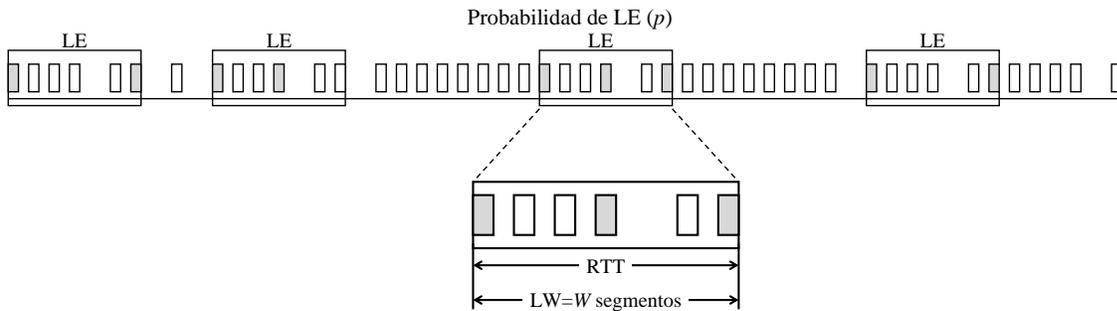


Figura 2.11: Modelo de pérdidas en [37].

En la primera parte se desarrolla el modelo sin considerar los TOs:

Se define el THR como:

$$\text{THR} = \frac{S_{\text{CAFR}}}{D_{\text{CAFR}}} \tag{2.47}$$

De [32] se tiene que:

$$E[\alpha] = \frac{1}{p} \quad (2.48)$$

Por lo tanto,  $S_{\text{CAFR}}$  se expresa como:

$$S_{\text{CAFR}} = \frac{1}{p} + \delta \quad (2.49)$$

Aunque en principio (2.48) es diferente de (2.3), en realidad son equivalentes. Esto se debe a que la definición de  $\alpha$  cambia entre el modelo [37] y el de [32]. Como el tamaño de la LW es  $W$ , el valor esperado de la separación entre la primera y la última pérdida en un mismo LE se aproxima como:

$$\delta \approx W - \frac{W}{E[m]} \quad (2.50)$$

Recordando que una LW comienza con un segmento perdido, se define  $A(W, m)$  como la probabilidad de que haya  $m$  segmentos perdidos en la LW (de tamaño  $W$ ):

$$A(W, m) = \binom{W-1}{m-1} (1-q)^{W-m} q^{m-1} \quad (2.51)$$

Ya que los LEs son detectados por TDs,  $m \leq W-3$ , por lo que:

$$E[m] = \sum_{m=1}^{W-3} mA(W, m) \approx 1 + (W-1)q \approx 1 + Wq \quad (2.52)$$

Sustituyendo (2.50) y (2.52) en (2.49) se obtiene una expresión para  $S_{\text{CAFR}}$ . Al igual que en [32], se deriva una expresión alternativa para  $S_{\text{CAFR}}$  con la finalidad de poder despejar de su igualación a  $W$ :

$$S_{\text{CAFR}} = S_{\text{CA}} + S_{\beta} + S_{\text{FR}} \quad (2.53)$$

Los autores usan (2.35) para la cantidad de segmentos enviados en cada ronda dentro de la FR, pero con una ligera modificación: traducen la función  $\text{máx}(\cdot)$  en los intervalos de  $m$  para los cuales se cumple que se transmiten  $W/2 - m + k - 1$  nuevos segmentos en la ronda  $k$  de la FR, con lo que se tiene que:

$$S_{\text{FR}} = \sum_{m=1}^{\frac{W}{2}} A(W, m) \sum_{k=1}^m \left( \frac{W}{2} - m + k - 1 \right) + \sum_{m=\frac{W}{2}+1}^{W-3} A(W, m) \sum_{k=m-\frac{W}{2}}^m \left( \frac{W}{2} - m + k - 1 \right) \quad (2.54)$$

Por otro lado, en la CA la  $cwnd$  crece desde  $W/2$  hasta  $W$ , de manera que:

$$S_{CA} = \sum_{i=\frac{W}{2}}^W i = \frac{3}{8}W^2 + \frac{3}{4}W \quad (2.55)$$

Ya que el primer segmento perdido puede ser cualesquiera de una ronda, y como dentro de la LW los segmentos se pierden independientemente y de manera uniforme,

$$S_{\beta} = \frac{W}{2} \quad (2.56)$$

Sustituyendo (2.54), (2.55) y (2.56) en (2.53) e igualándola con (2.49) se puede despejar  $W$ , cuya expresión simplificada queda como:

$$W \approx \frac{10pq - 5p + \sqrt{p(24 + 32q + 49p)}}{p(3 + 4q)} \quad (2.57)$$

Por otro lado, la duración del ciclo CA - FR es:

$$D_{CAFR} = D_{CA} + D_{\beta} + D_{FR} \quad (2.58)$$

Haciendo los cambios de variable pertinentes, según la definición de las variables en cada modelo,  $D_{CA} + D_{FR}$  se obtiene como en (2.28):

$$D_{CA} + D_{FR} = \left(\frac{W}{2} + 1\right)R + E[m]R = \left(\frac{W}{2} + Wq + 2\right)R \quad (2.59)$$

Y  $D_{\beta} = \frac{R}{2}$ . De esta manera se obtiene que:

$$THR = \frac{\frac{1}{p} + \frac{W^2q}{1+Wq}}{\left(\frac{W}{2} + Wq + \frac{5}{2}\right)R} \quad (2.60)$$

con  $W$  definida en (2.57).

#### 2.4.4. Desarrollo de la segunda etapa del modelo desarrollado en [37].

En la segunda parte se desarrolla el modelo que toma en cuenta los TOs, llamado “modelo completo” (ver figura 2.12):

De acuerdo a las simplificaciones realizadas en este modelo, un TO se puede producir sólo en dos circunstancias: estando en la fase CA o estando en la FR. Estando en la CA, un TO se produce bajo las mismas condiciones mencionadas en [32] y reflejadas en  $Q$  en (2.24) y

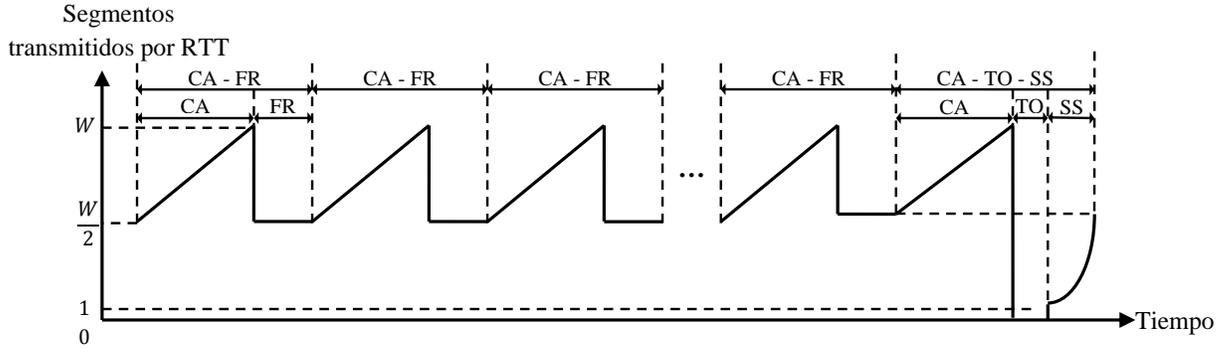


Figura 2.12: Ciclos para el modelo completo [37].

(2.23). Estando en la FR, un TO se produce, según las suposiciones particulares de este modelo, ante la presencia de cualquier pérdida.

Las variables adicionales usadas para el modelo completo son:

- $p_{TO}$  es la probabilidad de que suceda un TO (estando en la CA o en la FR).
- $p_{DTO}$  es la probabilidad de que, estando en la CA, suceda un TO.
- $p_{IFR}$  es la probabilidad de que, estando en la FR, suceda un TO (lo que se denomina como FR incompleta: IFR – *Incomplete FR* –).
- $S_{SS}$  es el valor esperado de la cantidad de segmentos enviados exitosamente en la fase SS después del ciclo TO.
- $S_{TO}$  es el valor esperado de la cantidad de segmentos nuevos enviados exitosamente durante el ciclo TO. En realidad  $S_{TO} = 0$ , pues, por definición, los segmentos enviados en la sucesión de TOs consecutivos son las retransmisiones del primer segmento perdido del LE.
- $D_{SS}$  es el valor esperado de la duración de la fase SS posterior al ciclo TO.
- $D_{TO}$  es el valor esperado de la duración del ciclo TO.
- $T_0$  es el valor esperado de la RTO.

La expresión para el THR es:

$$\text{THR} = \frac{(1 - p_{TO})S_{CAFR} + p_{TO}(S_{CA} + S_{TO} + S_{SS})}{(1 - p_{TO})D_{CAFR} + p_{TO}(D_{CA} + D_{\beta} + D_{TO} + D_{SS})} \quad (2.61)$$

No se consideran para el cálculo del THR los segmentos  $S_\beta$ . Esto porque se supone como *Go-Back-N* al mecanismo de recuperación de las pérdidas detectadas por un TO, por lo que, a partir del segmento detectado como perdido, se retransmiten todos los segmentos que se habían enviado hasta el momento de la detección de la pérdida.

La probabilidad  $p_{TO}$  se define como:

$$p_{TO} = p_{DTO} + p_{IFR} \quad (2.62)$$

Donde,  $p_{DTO}$ , que es la probabilidad de que suceda un TO estando en la CA, se obtiene con:

$$p_{DTO} = \sum_{m=W-2}^W A(m, W) \quad (2.63)$$

Las condiciones detrás de la expresión anterior son las mismas que las que se consideraron en [32] para obtener  $Q$ , sin embargo, (2.63) es distinta a (2.23). Esto se debe a que la definición de  $A(w, k)$  en [32] difiere de la de  $A(W, m)$  en [37].

La única posibilidad de que la fase FR de NewReno “lento pero firme” termine en un TO es que se pierda una retransmisión (o su ACK, pero en este caso se tiene la consideración general que ningún ACK se pierde). Sin embargo, para obtener la  $p_{IFR}$  se realiza la siguiente simplificación al mecanismo de FR de NewReno “lento pero firme”:

- En lugar de modelar explícitamente la pérdida de una retransmisión en la FR, se supone que cualquier LE dentro de la FR involucra la pérdida de una retransmisión y por lo tanto fuerza un TO.
- Se considera que en cada una de las  $m$  rondas dentro de la FR se transmiten (incluyendo la retransmisión)  $W/2$  segmentos.

Bajo estas condiciones, la  $p_{IFR}$  se calcula como la probabilidad de que cualquiera de los  $W/2$  segmentos transmitidos en cada ronda se pierda y por lo tanto genere un LE siguiente:

$$p_{IFR} = \sum_{m=1}^{W-3} A(W, m) \left[ p + (1-p)p + \dots + (1-p)^{\frac{mW}{2}-1} p \right] \quad (2.64)$$

$$p_{IFR} = \sum_{m=1}^{W-3} A(W, m) \left[ 1 - (1-p)^{\frac{mW}{2}} \right]$$

Por otro lado,  $D_{TO}$  está dado como en [32] por (2.20) y las expresiones para  $S_{SS}$  y  $D_{SS}$  (medida en RTTs) equivalen a (2.40) y (2.39), respectivamente, sustituyendo  $E[W]$  por  $W$ .

A partir de este desarrollo, el THR queda definido como:

$$\text{THR} = \frac{\frac{1}{p} + \frac{W^2 q}{1+Wq}}{\left(\frac{W}{2} + \frac{3}{2} + (1 - p_{TO})(1 + Wq)\right)R + p_{TO}\left((1 + 2p + 4p^2)T_0 + \left(1 + \log \frac{W}{4}\right)R\right)} \quad (2.65)$$

donde  $W$  está definido en (2.57).

En [37] se hace una recomendación para obtener el valor de  $q$  a partir de la medición de  $p$ , como la tasa promedio de LEs, y de la tasa promedio de errores,  $q'$ . Primero:

$$E[m] = \frac{q'}{p} \quad (2.66)$$

Usando (2.52) se obtiene que:

$$q \approx \frac{E[m] - 1}{W - 1} \quad (2.67)$$

donde  $W$  se calcula con (2.57) usando la aproximación  $q = q'$ .

#### 2.4.5. Validación del modelo reportado en [37].

Se presenta la validación del modelo de [37] a través de la simulación de dos escenarios en ns-2 [55]:

- CASO 1: Sin tráfico de fondo
- CASO 2: Con tráfico de fondo

CASO 1.

La topología empleada para este caso se muestra en la figura 2.13. Corresponde, con cambios mínimos, al caso IV-C de [37] y se realizó tomando como base el código publicado en [56], incluyendo algunas modificaciones y correcciones al mismo. Se evalúa el THR de un flujo TCP NewReno de larga duración (1000 s) que atraviesa un enlace “cuello de botella” configurado como sigue:

- Velocidad del enlace de 15 Mbps.
- Cola tipo *Drop-Tail*, con capacidad suficientemente grande como para no provocar pérdidas por congestión.
- Las pérdidas de los paquetes se fuerzan mediante un módulo de pérdidas, que opera con una probabilidad de pérdida de segmento con distribución de Bernoulli.
- Retardo fijo.

El retardo en los enlaces de acceso de la fuente y el destino TCP se calcula para que, junto con el retardo en el enlace “cuello de botella”, resulte en un RTT de 75 ms.

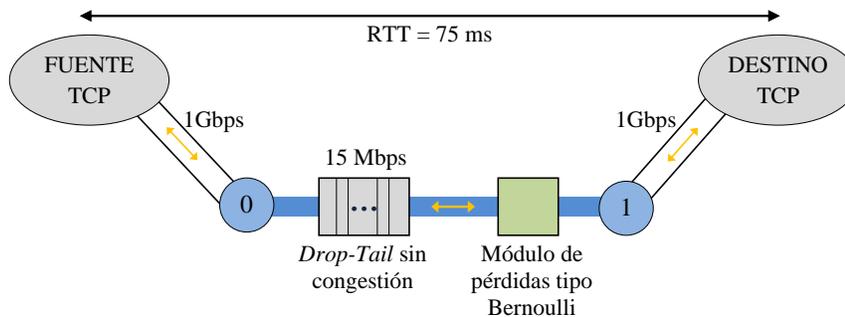


Figura 2.13: Topología usada en el CASO 1 de la validación del modelo en [37].

El modelo es función de los siguientes parámetros:

- La tasa promedio de LEs ( $p$ ), que se obtiene directamente de la simulación y según [56].
- La tasa promedio de segmentos perdidos dentro de cada LE ( $q$ ), que se obtiene con (2.67), (2.66) y (2.57). Para ello, se obtiene la tasa promedio de errores,  $q'$ , directamente de la simulación.
- El  $E[\text{RTT}]$  se iguala al RTT promedio obtenido de la simulación.
- El  $E[\text{RTO}]$  se aproxima como  $3E[\text{RTT}]$ .

A continuación se presenta una discusión sobre la medición de estos parámetros:

La tasa promedio de errores,  $q'$ , puede obtenerse desde el punto de vista de la red (en particular, vista desde el enlace “cuello de botella”),  $q_{\text{red}}$ ; desde el punto de vista del flujo TCP

mediante el monitor de flujos en ns-2,  $q_{\text{flujo}}$ ; o puede ser calculada dentro del transmisor TCP NewReno en ns-2 (como en [56]), haciendo un cálculo de la cantidad de segmentos perdidos,  $q_{\text{cal}}$ . En un caso como este, en donde no hay flujos adicionales al flujo TCP NewReno de estudio que atraviesen el “cuello de botella”, la  $q_{\text{red}}$  equivale a la  $q_{\text{flujo}}$  y representan exactamente la tasa de segmentos perdidos, pues son el cociente de la cantidad de segmentos perdidos sobre la cantidad de segmentos enviados en total, medidas durante un intervalo de observación. Por otro lado, el criterio según [56] para calcular la  $q_{\text{cal}}$  está restringido por la información disponible en el transmisor TCP.  $q_{\text{cal}}$  se calcula como el cociente del total de segmentos perdidos (TL: *Total Loss*) sobre el total de segmentos enviados (TS: *Total Send*), en el intervalo de observación. Sin embargo, en el transmisor TCP no hay información suficiente como para determinar de manera exacta ambas cantidades dentro de un intervalo arbitrario de observación, como se analiza a continuación.

Respecto al cálculo del TL, el transmisor TCP NewReno no tiene manera para saber exactamente cuáles y cuántos son los segmentos que se han perdido, sobre todo si el mecanismo de recuperación de pérdidas que se activa es el asociado con un TO. Consecuentemente, tampoco puede inferir la cantidad de segmentos enviados exitosamente (TA: *Total Acknowledged*). El TL se actualiza dentro de la FR o con el TO de la manera siguiente:

- Durante la FR, el TL se incrementa en uno al entrar a la FR y con cada PA. Esto es adecuado porque tanto la activación de la FR como cada PA indica la pérdida de un segmento.
- Cuando se presenta por primera vez el TO asociado con un segmento, se considera que todos los segmentos a partir de él se han perdidos (y por tanto, se retransmiten), lo que resulta en un incremento en el TL igual a esta cantidad de segmentos. Esto, es una sobre-estimación de la cantidad de segmentos perdidos, pues cuando ocurre el TO, indicando la pérdida de un segmento, no necesariamente todos los segmentos siguientes se han perdido también.<sup>3</sup>

El TS se calcula como la suma TL + TA, aproximando el TA como la cantidad de segmentos de datos distintos que constituye el flujo TCP en el intervalo de observación. Es decir, el TA es ligeramente subestimado, pues no cuenta como transmisión exitosa a las retransmisiones innecesarias que no se pierden.

---

<sup>3</sup>Esto se cumple sobre todo cuando se tiene una condición en que el RA más actual es menor que el *recover* (como por ejemplo, cuando una FR termina en un TO), y la recuperación de pérdidas no puede darse más que a través de un TO, aunque se reciban más de tres DAs. Así, incluso la pérdida de un solo segmento en una ventana de tamaño mayor a tres, deberá recuperarse mediante un TO

La diferencia entre la cantidad exacta de segmentos perdidos y su cálculo aproximado como el TL, así como la de los segmentos exitosamente transmitidos y el TA se muestra en la figura 2.14. Como se muestra en la figura, conforme la probabilidad de pérdida de segmentos aumenta, la sobre-estimación del TL es mayor e inclusive se vuelve de casi el doble para una tasa Bernoulli de pérdidas del 10 %. Por otro lado, la sub-estimación del TA es ligera, lo que corresponde a una cantidad de retransmisiones innecesarias correspondientemente pequeña.

Ya que la cantidad de segmentos perdidos es significativamente menor que la cantidad de segmentos exitosamente transmitidos, y por tanto, que la cantidad de segmentos enviados en total, la aproximación del TS no resulta tan errónea, como se muestra en la figura 2.15, teniendo un error de casi 7 % para una tasa Bernoulli de pérdidas del 10 %.

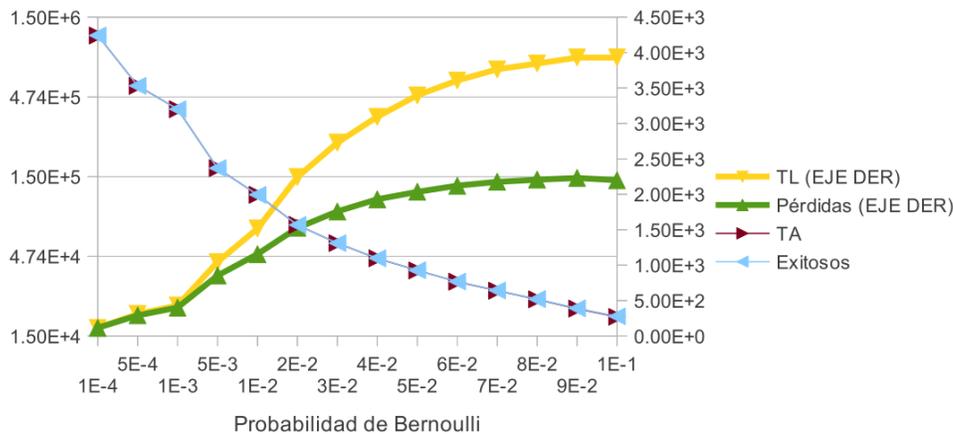


Figura 2.14: Cantidad de segmentos perdidos y exitosos.

Así, bajo la definición  $q_{cal} = TL/TS$ , en la figura 2.16 se observa cómo éste valor es una sobre-estimación de la tasa de segmentos perdidos real representada, en este caso, tanto como por la  $q_{red}$  como por la  $q_{flujo}$ . El error producido por esta aproximación es en promedio del 28 % y llega a ser de hasta el 39 % para una tasa Bernoulli de pérdidas del 10 %. En la figura 2.16 también se muestra el valor del parámetro  $q$  requerido por el modelo y calculado según (2.67), (2.66) y (2.57) y usando  $q' = q_{cal}$ , como en [56].

Para el parámetro  $p$  del modelo, se usa un conteo de LEs en el transmisor TCP. Según [56],  $p = E/TS$ , donde  $E$  es el contador de LEs en el flujo TCP NewReno de estudio durante el intervalo de observación. De acuerdo con la definición de que un LE dura un RTT y se inicia con el primer segmento perdido en una ronda, dado que no hubo segmentos perdidos en los

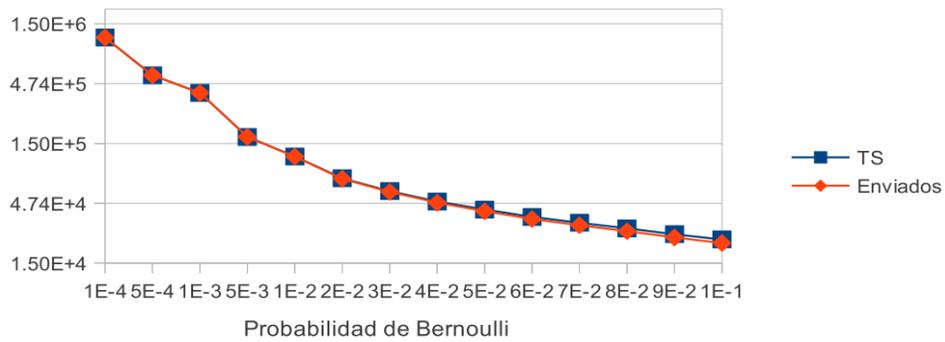


Figura 2.15: Cálculo de la cantidad de segmentos enviados vs. cantidad real de segmentos enviados.

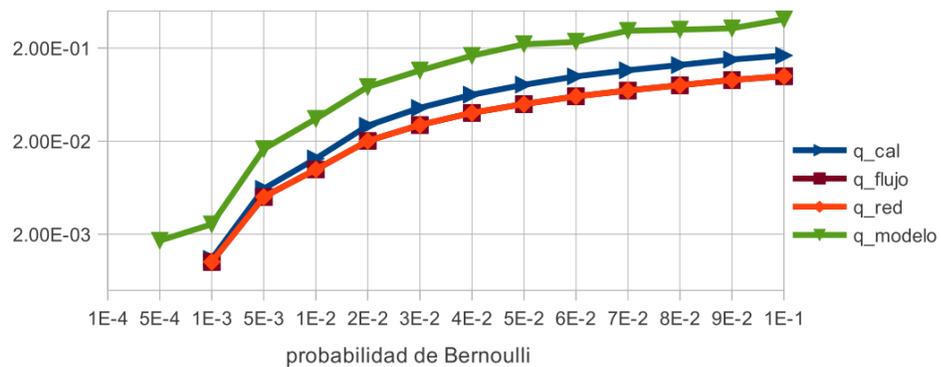


Figura 2.16: Varias aproximaciones para  $q'$ . Además,  $q$  vs  $q'$ .

$cwnd$  segmentos anteriores, entonces se puede concluir que toda pérdida detectada por un TD o un TO indica el inicio de un LE. Así, durante el intervalo de observación, la variable  $E$  se actualiza en cada inicio de una FR o en cada TO, incrementándose en uno. En la gráfica de la figura 2.17 se muestra la curva para este valor de  $p$  (como  $p_{cal}$ ), en comparación con el cálculo de  $p$  como el cociente de  $E$  sobre la cantidad total de segmentos enviados en ese intervalo, vista desde el monitor del flujo TCP. Como se mencionó arriba, TS sobrestima la cantidad real de segmentos enviados en el intervalo de observación, lo que se traduce en una sub-estimación creciente de  $p$  a través de  $p_{cal}$ , conforme se incrementa la tasa de pérdidas.

En la figura 2.18 se muestran dos aproximaciones del THR obtenido a partir de la simulación: la primera es el THR según [56], es decir  $THR = TA/(\text{intervalo de observación})$ , mientras que la segunda es el THR real, visto desde el monitor del flujo como el cociente de la cantidad de segmentos exitosos sobre la cantidad de segmentos enviados en total, durante el intervalo de

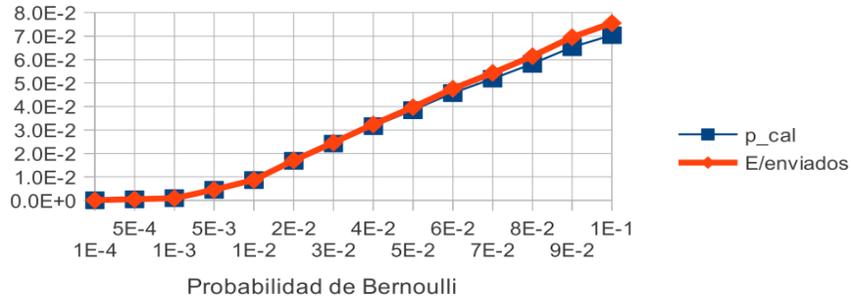


Figura 2.17: Cálculo de  $p$ .

observación. Como ya se había mencionado, TA es una sub-estimación ligera de la cantidad de segmentos exitosos real, que no considera las retransmisiones exitosas innecesarias. El error en esta aproximación es muy poco significativo, resultando en un error promedio entre estas dos curvas del 0,16 %.

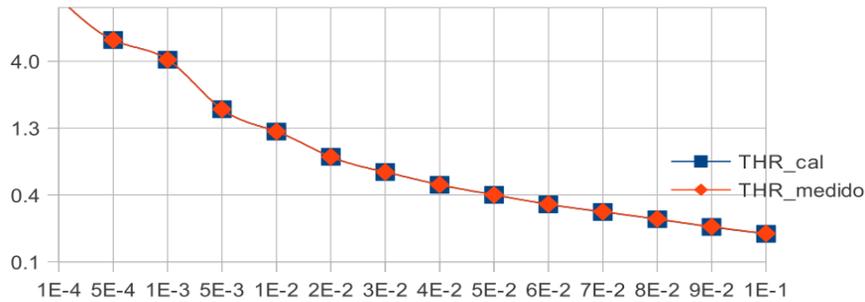


Figura 2.18: THR (en Mbps) obtenido de la simulación en ns-2.

Por otro lado, las distintas aproximaciones para obtener los parámetros  $p$  y  $q'$  resultan en valores diferentes para el THR calculado según [37]. Por ejemplo, si consideramos  $p = E/ enviados$  y  $q' = q_{red}$ , es decir, si tomamos los parámetros  $p$  y  $q'$  de los módulos de medición de ns-2 (excepto para el conteo de  $E$ ), se obtiene una curva para el THR (THR valores medidos) distinta a la que surge si  $p = p_{cal}$  y  $q' = q_{cal}$  (THR valores calculados). Como se muestra en la figura 2.19, la diferencia entre ambas curvas en es promedio de casi el 5 % y como máximo de aproximadamente el 12 %.

Finalmente, la comparación entre el THR dado por [37] y el THR obtenido de la simulación se puede verificar en la figura 2.20, en donde aparece la curva del THR según [37] con valores

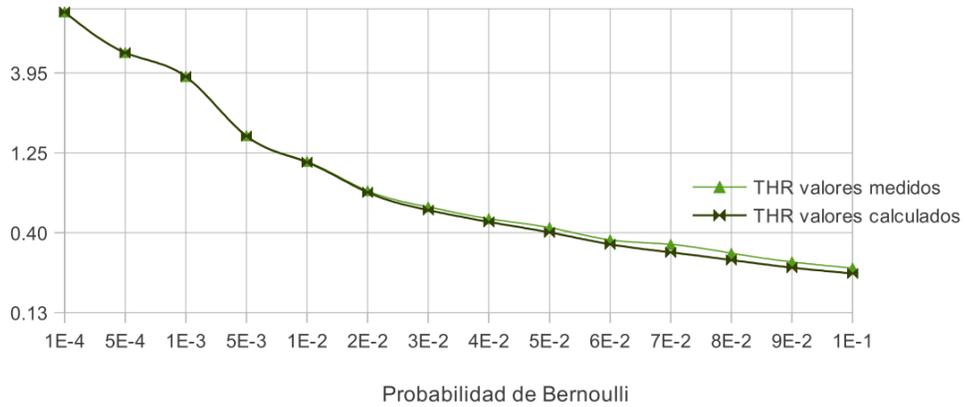


Figura 2.19: THR (en Mbps) obtenido con [37].

de los parámetros medidos, la curva del THR según [37] con los valores de los parámetros calculados y la curva del THR calculado directamente de la simulación. En esta gráfica se observa que, el THR modelado en [37] se acerca más al THR real usando los valores calculados según [56] de  $p$  y  $q'$ , aunque éstos sean aproximaciones de los valores reales de  $p$  y  $q'$ , respectivamente.

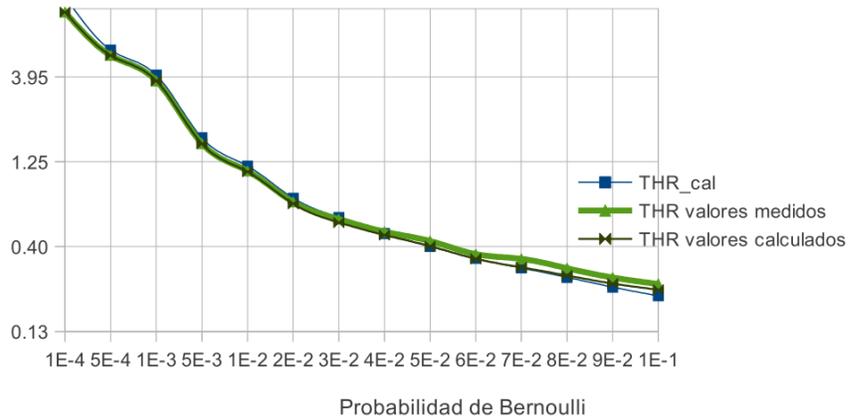


Figura 2.20: THR (en Mbps) obtenido con [37] vs THR simulado.

## CASO 2.

La topología empleada para este caso se muestra en la figura 2.21. Corresponde, con cambios mínimos, al caso D de [37] y se realizó con el código publicado en [56], incluyendo algunas

modificaciones y correcciones al mismo. Se evalúa el THR de un flujo TCP NewReno de larga duración (1000 s) que atraviesa un enlace “cuello de botella” configurado como sigue:

- Velocidad del enlace de 15 Mbps.
- Cola con capacidad de 150 paquetes. En el CASO 2a, la cola es de tipo *Drop-Tail*. En el CASO 2b, la cola es RED con límites mínimo y máximo del 1/3 y 2/3 de la capacidad máxima, respectivamente.
- Las pérdidas de los paquetes se deben a la convivencia del flujo de estudio con una cantidad variable de flujos de tráfico de fondo. En el CASO 2a, el número de flujos de fondo va desde 50 hasta 500. Para el CASO 2b, desde 80 a 220. El tráfico de fondo es una mezcla del 75 % de tráfico tipo HTTP con un 25 % de flujos FTP de larga duración (semejante a la duración del flujo de estudio).
- Retardo fijo.

El retardo en los enlaces de acceso de la fuente y el destino TCP se calcula para que, junto con el retardo en el enlace “cuello de botella”, resulte en un RTT de 75 ms.

Cada sesión HTTP consiste de un par cliente/servidor único. El servidor responde, inmediatamente después de recibir una solicitud del cliente para que lo atienda, con la emisión de un archivo de un tamaño con distribución de Pareto (con media de 48 kB y factor de forma de 1.2). El cliente espera un tiempo exponencial con media de 500 ms entre sesiones.

Todos los flujos de fondo usan TCP NewReno con una *rwnd* máxima de 64 kB y con un tamaño fijo de paquete de 1 kB. El RTT para los flujos de fondo se distribuye uniformemente entre 20 y 460 ms.

Los parámetros para el modelo se obtienen igual que en el CASO 1. En promedio, la diferencia entre  $q_{cal}$  y  $q_{red}$  es del 18,2 % en el CASO 2a, y del 37 % en el CASO 2b. La diferencia entre  $q_{cal}$  y  $q_{flujo}$  es, en promedio, del 8,4 %, en el CASO 2a, y del 38 % en el CASO 2b.

Igualmente, entre  $p_{cal}$  y  $p = E/\text{enviados}$  existe una ligera diferencia de aproximadamente el 4 % en promedio.

La diferencia entre la cantidad exacta de segmentos perdidos y su cálculo aproximado como TL se vuelve de casi el 30 % a partir de 200 flujos de tráfico de fondo para el CASO 2a y del 40 % en todo el CASO 2b (figura 2.22). Por otro lado, la sub-estimación del TA es en

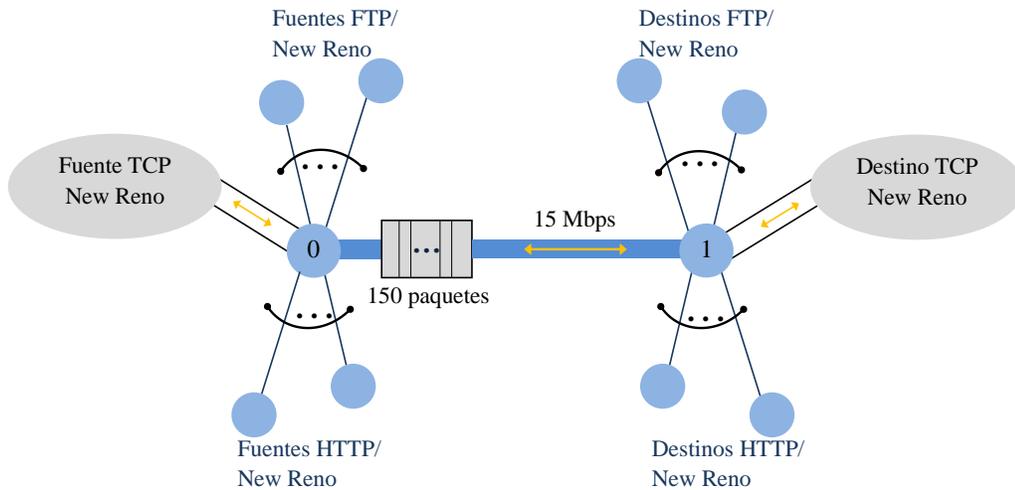


Figura 2.21: Topología usada en el CASO 2 de la validación del modelo en [37].

promedio del 0,66 % en el CASO 2a (y del 0,15 % en el CASO 2b) y la sobre-estimación del TS sobrepasa el 6 % a partir de los 300 flujos de tráfico de fondo en el CASO 2a (y a partir de los 200 para el CASO 2b).

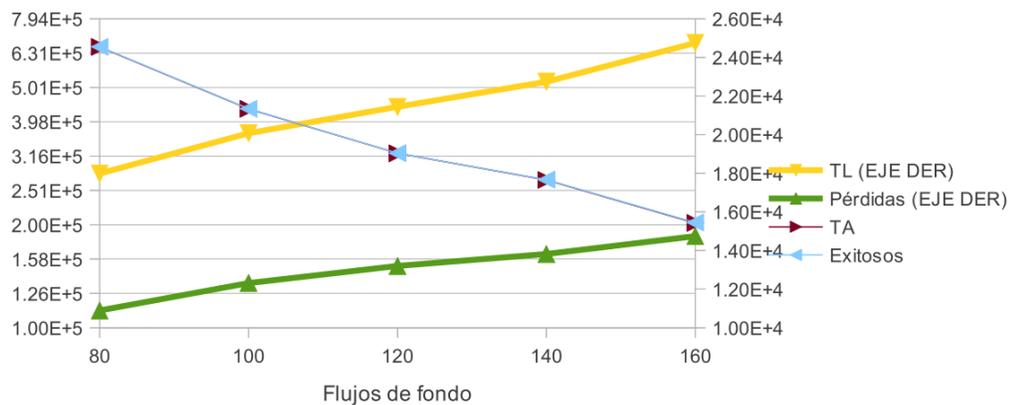


Figura 2.22: Cantidad de segmentos perdidos y exitosos.

Finalmente, la comparación entre el THR dado por [37] y el THR obtenido de la simulación para el CASO 2a se puede verificar en la figura 2.23, en donde aparece la curva del THR según [37] con valores de los parámetros medidos, la curva del THR según [37] con los valores de los parámetros calculados y la curva del THR calculado directamente de la simulación. El error promedio entre el THR calculado de la simulación y el THR según [37] es de aproxi-

madamente el 19 %, con los valores de los parámetros calculados, y casi del 16 % con valores de los parámetros medidos. Por lo que se puede inferir que, ante un escenario más cercano a la realidad, el modelo responde mejor usando los valores de los parámetros más precisos posibles.

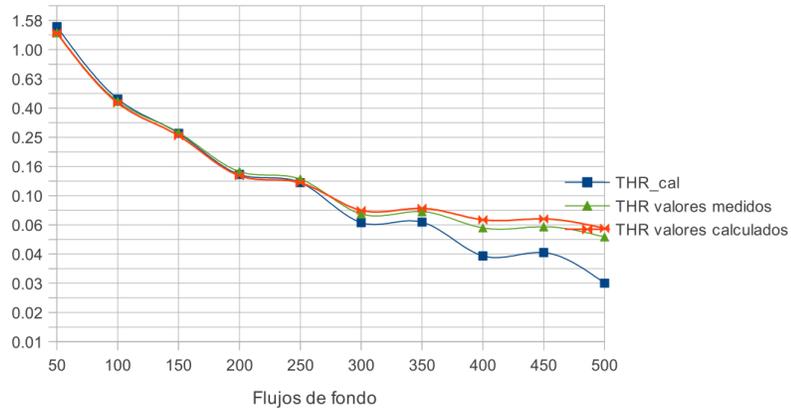


Figura 2.23: THR (en Mbps) obtenido con [37] vs THR simulado para el CASO 2a.

Para el CASO 2b, la comparación entre las curvas del THR según [37] con valores de los parámetros calculados y la curva del THR calculado directamente de la simulación se presenta en la gráfica de la figura 2.24. El error promedio entre estas dos curvas, para una variación de flujos de fondo de desde 80 a 220, es de aproximadamente el 8 %.

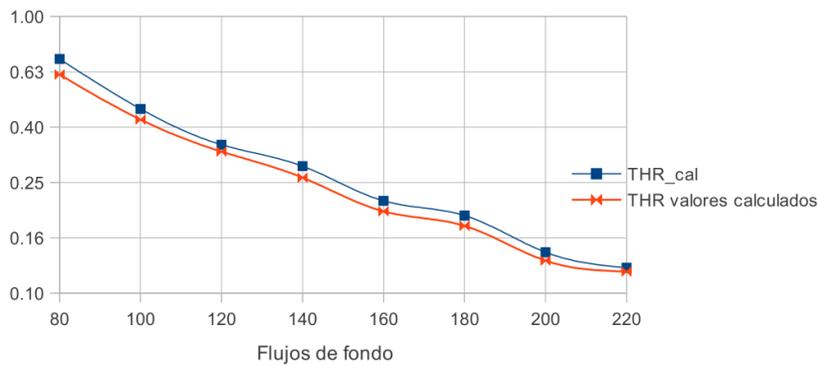


Figura 2.24: THR (en Mbps) obtenido con [37] vs THR simulado para el CASO 2b.

#### 2.4.6. Conclusiones del modelo reportado en [37].

La principal contribución de este trabajo es que es el primero que modela explícitamente el mecanismo de FR de NewReno incluyendo la posibilidad de detección de pérdidas mediante un TO. Esto permite una aproximación significativa del THR modelado con el THR real, según se verificó en los casos anteriores. No obstante, el error de la aproximación está sujeto a factores múltiples, entre ellos el escenario de estudio. En [37] se puede revisar que la aproximación resulta en distintos errores según sea el ambiente de operación, es decir, según sea la dinámica de pérdida de los segmentos en la red, la cual es la directriz del comportamiento de TCP.

Por otro lado, para el desarrollo de este modelo se recurre a varias simplificaciones, de entre las cuales principalmente se tiene que:

- No es válido para flujos de corta duración (los cuales representan el mayor porcentaje de la cantidad de flujos transitando en Internet [38] [39] [41] [42] [47]).
- No hay pérdidas durante las fases SS, lo que puede resultar en una sobre-estimación del THR o simplemente en una mala aproximación.
- En lugar de modelar explícitamente la pérdida de una retransmisión en la FR, se supone que cualquier LE dentro de la FR involucra la pérdida de una retransmisión y, por lo tanto, fuerza un TO.
- Se considera que en cada ronda dentro de la FR se transmiten (incluyendo la retransmisión)  $W/2$  segmentos.
- Al ocurrir un TO, se considera que son retransmitidos todos los segmentos a partir del asociado con el TO y hasta el último segmento transmitido hasta el momento en que se presenta el TO, lo cual no necesariamente se cumple.

En el modelo desarrollado dentro de este trabajo doctoral se elimina la mayoría de estas simplificaciones, dejando sólo las suposiciones y características generales que se comparten en este conjunto de trabajos.



## Capítulo 3

# Algoritmo para la evaluación del desempeño en estado estable de una conexión TCP NewReno de larga duración.

En este capítulo se desarrolla una herramienta de evaluación que mejora sustancialmente la evaluación del desempeño en estado estable de una conexión TCP NewReno de larga duración, en comparación con la obtenida con [37], que es el trabajo más reciente y completo en este problema. Usando como base la línea de trabajo constituida por [32], [48], [49], [50] y [37], se busca lograr dicho objetivo a través de la incorporación en el análisis de una representación más cercana de los mecanismos del protocolo, eliminando algunas de las simplificaciones a las que se han recurrido en los modelos existentes.

Basados en [32], los modelos en [48], [49], [50] y [37] fueron propuestos como una concatenación de ciclos estadísticamente idénticos. Esta forma de análisis tiene una enorme ventaja: la posibilidad de analizar toda la conexión a través del análisis de sólo uno de estos ciclos, y por ende, la factibilidad de contar con una expresión analítica cerrada para representar al THR. Un ciclo puede dividirse en sub-ciclos. En general, cada ciclo o sub-ciclo consiste de una secuencia de mecanismos o fases del protocolo. Por ejemplo, en el modelo más reciente para TCP NewReno [37], que también es el más completo, cada ciclo está constituido por sub-ciclos CA - FR estadísticamente idénticos seguidos de un sub-ciclo CA - TO - SS. Un sub-ciclo se caracteriza por considerar un valor típico del tamaño de la *cwnd* al finalizar (o al comenzar) el mismo, sin importar cuáles mecanismos lo constituyen; así, cada sub-ciclo comienza suponiendo que el sub-ciclo anterior termina con dicho valor de la *cwnd*. Esta característica es la que permite que se puedan concatenar distintos sub-ciclos y que el análisis se

pueda reducir para considerar sólo un ciclo que contenga las combinaciones posibles de los sub-ciclos. Sin embargo, a su vez, esta característica también tiene implícita una restricción: que los mecanismos involucrados en los ciclos operan únicamente bajo el conjunto de condiciones que permiten que los sub-ciclos terminen con ese valor de la *cwnd*. En este capítulo se desarrolla un análisis que elimina parcialmente esta simplificación al incorporar todos los casos faltantes que pueden ocurrir en los mecanismos SS y CA de NewReno; sin embargo, se mantienen la mayoría de las suposiciones que se usan para los mecanismos de recuperación de pérdidas con el objetivo de reducir la complejidad del estudio.

En la figura 3.1 están representadas todas las transiciones entre las fases del protocolo consideradas en este trabajo. Así mismo, se muestran en la figura las condiciones asociadas a cada transición. Como contraste con los modelos actuales, se resaltan aquellas que son nuevas. Revisando con detalle un conjunto de experimentos por simulación que registra una muestra de las posibilidades en la dinámica de TCP NewReno fue posible deducir que la secuencia de fases que sigue una conexión particular está supeditada a la ubicación de las pérdidas en su flujo de segmentos. Asimismo, la duración y la cantidad de segmentos enviados en cada fase dependen de la trayectoria de fases que ha seguido la conexión. Por ello, para esta propuesta se descarta la idea de continuar el modelado con su representación mediante ciclos estadísticamente idénticos, pues aunque podría plantearse que las transiciones y sus causas faltantes en [37] den origen a otros sub-ciclos, por ejemplo CA - TO - SS - TO - SS ó CA - TO - SS - FR, no podría quedar establecido un único orden de concatenación entre ellos. Además, la introducción de estas condiciones de operación faltantes en los modelos actuales rompe con la suposición fundamental del análisis cíclico de [37] y sus predecesores, pues no existen ciclos que incorporen estos casos y que, aun así, cumplan con la característica de terminar con el valor de la *cwnd* típico. Es por esta razón que se abandona el método de análisis usado en dichos trabajos y se opta por desarrollar un algoritmo como método de evaluación del THR en estado estable de una conexión TCP NewReno de larga duración. La ventaja que presenta este algoritmo respecto de los modelos analíticos existentes es que, gracias a la representación más fidedigna de los mecanismos del protocolo, se obtienen resultados significativamente más cercanos a los obtenidos por simulación, evitando a su vez la complejidad y el consumo de tiempo y recursos asociados con ésta.

Particularmente, este algoritmo realiza un seguimiento de las fases por las que atraviesa una conexión TCP NewReno de larga duración. El algoritmo utiliza un modelo para caracterizar el comportamiento de pérdidas de la red. El orden en que se siguen las fases en una conexión se determina por la secuencia de segmentos erróneos a lo largo de ella, la cual se genera por

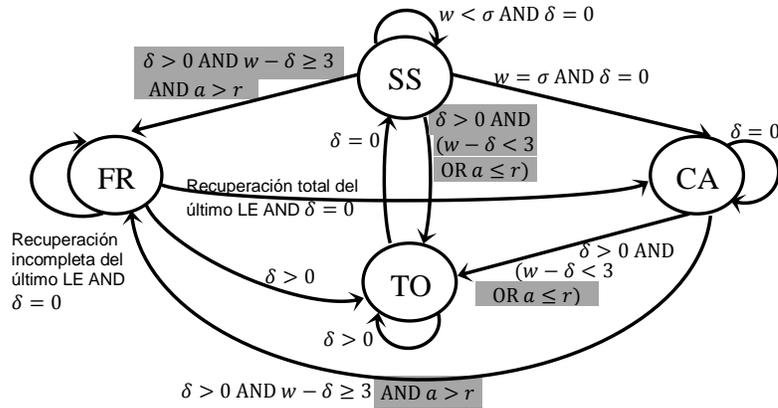


Figura 3.1: Fases del protocolo TCP NewReno y las transiciones entre ellas.

el modelo de pérdidas.

El algoritmo se valida a través de su comparación con los resultados de un conjunto de pruebas de simulación en ns-2 y se contrasta con los resultados del modelo presentado en [37], que es el modelo analítico más actual y completo de TCP NewReno. En una primera parte el comportamiento de pérdidas de la red se modela a través del proceso de Bernoulli, obteniendo un error promedio del 1.4 % en contraste con el 7 % correspondiente al modelo en [37]. Posteriormente se emplea el modelo de Gilbert para representar el comportamiento de pérdidas de la red. En este caso, el error promedio del algoritmo crece hasta un 13.1 % mientras que el error del modelo reportado en [37] asciende hasta un 1213 %.

### 3.1. Suposiciones sobre las que se desarrolla el algoritmo.

#### 3.1.1. Suposiciones generales.

Se conservan las suposiciones generales 1 - 10, 12 - 14 y 20 - 21 de la sección 2.2.1, correspondientes al modelo en [32] y retomadas en [48], [49], [50] y [37].

#### 3.1.2. Suposiciones que se derivan del modelo presentado en [37].

Se retoman las suposiciones particulares 2 - 9 de la sección 2.4.2.

### 3.1.3. Suposiciones particulares usadas para este algoritmo.

1. La probabilidad de pérdida de un segmento obedece a un modelo de pérdidas de la red, que puede cambiar según el ambiente que se quiera caracterizar mediante él.
2. El lugar de asociar a la dinámica de TCP con una concatenación de ciclos estadísticamente idénticos, se considera que una conexión TCP transita entre distintas fases comenzando con la fase CA.
3. Las posibles transiciones entre las fases del protocolo y las condiciones que las activan se muestran en la figura 3.1
4. La secuencia de fases es determinada a su vez por la secuencia de segmentos perdidos generados por el modelo de pérdidas.
5. El inicio de cada fase coincide con el inicio de una ronda.
6. El tránsito a través de cada fase dura un número entero de rondas, excepto cuando la siguiente fase es un TO, en cuyo caso la RTO (ponderada según el mecanismo de *back-off*) se adiciona a su duración.
7. Las rondas en cada fase se enumeran a partir de cero hasta  $i$ .
8. La suposición 11 de la sección 2.2.1 se modifica ligeramente como: una ronda comienza con la transmisión en ráfaga (*back-to-back*) de todos los segmentos con permiso de ser transmitidos, incluyendo las retransmisiones y los segmentos de la *cwnd*, y termina con la recepción de los RAs que reconozcan algunos o todos ellos. De forma que su duración es un RTT promedio.
9. La definición de LE, mencionada en la suposición 1 de la sección 2.4.2 se modifica como sigue: Un evento de pérdida, LE, comienza con el primer segmento perdido en una fase, tiene longitud  $w$  y una duración de un RTT.

## 3.2. Descripción del algoritmo.

Este algoritmo tiene como objetivo la evaluación del THR en estado estable de una conexión TCP NewReno de larga duración. Sus parámetros de entrada son el RTT promedio,  $R$ ; la duración promedio, en rondas, del temporizador de pérdidas cuando el factor de *back-off* es unitario, RTO; la duración máxima de la RTO,  $RTO_{\max}$ ; el tamaño máximo de segmento en

bits, SMSS; el valor inicial del  $ssthresh$ ,  $\sigma_{ini}$ ; la duración típica de una conexión TCP de larga duración,  $T$ ; y los parámetros del modelo de pérdidas empleado para caracterizar el comportamiento de pérdidas de la red.

Tabla 3.1 Variables globales.

Variable	Descripción
$a$	SN recibido exitosamente máximo.
$\Delta$	Primer segmento perdido en la fase.
$\delta$	Cantidad de segmentos perdidos en la fase.
$n_{TO}$	Es el exponente de la base 2 con que se pondera a la RTO.
$r$	Variable <i>recover</i> de TCP NewReno.
$m$	SN transmitido máximo.
$\sigma$	$ssthresh$ .
$w$	Tamaño de la <i>cwnd</i> .

Si  $X$  es la cantidad de segmentos, todos de tamaño SMSS, enviados exitosamente durante un intervalo de  $A$  rondas, y  $Y$  es el número de segmentos, todos de tamaño SMSS, enviados en total en ese mismo intervalo, entonces la tasa efectiva de transmisión en estado estable, en bps, se calcula como:

$$THR_E = \lim_{A \rightarrow \infty} \frac{X(SMSS)}{AR} \quad (3.1)$$

Mientras que la tasa de transmisión en estado estable, en bps, está dada por:

$$THR = \lim_{A \rightarrow \infty} \frac{Y(SMSS)}{AR} \quad (3.2)$$

Cada fase del protocolo tiene asociada una parte del algoritmo, de forma que éste se mueve a través de las fases según lo determine el modelo que representa el comportamiento de pérdidas de la red. Cada fase está representada por un conjunto de variables globales, listadas en la tabla 3.1

Los valores de inicialización de las variables globales son:

$$a = m = r = X = Y = 0.$$

$$A = 0.$$

El algoritmo comienza con la fase CA. El modelo de pérdidas se ejecuta hasta que ocurre la primera pérdida,  $\Delta$ .  $w$  toma el valor inicial según:

$$w = \min(\max(2, \frac{\Delta}{2}), \sigma_{ini}) \quad (3.3)$$

Donde  $\sigma_{ini}$  toma un valor arbitrariamente alto [15]. La idea detrás de (3.3) es la siguiente: En el análisis en estado estable de una conexión TCP de larga duración se desprecia su paso por las fases iniciales y se comienza por una fase CA cualquiera. Sin embargo, para el análisis de esta fase, según el diseño el algoritmo, se requiere del valor de  $w$  a su entrada. Una manera de aproximar este valor es mediante (3.3). Se supone que la CA a partir del cual se analiza el flujo TCP ocurre después de un SS que termina sin pérdidas. Si  $\frac{\Delta}{2} > \sigma_{ini}$ , entonces se considera que la CA procede al SS inicial y, por lo tanto, que la CA comenzó con  $w = \sigma_{ini}$ . De otra forma, se supone que la pérdida ocurre en la primera ventana de la CA, de forma que  $w = \frac{\Delta}{2}$  al inicio de la CA. Debido a que se trata de una conexión TCP de largo término, el valor inicial de  $w$  tiene un poco de peso en el análisis completo.

Según el algoritmo de Karn [11], que dice que un ACK de un segmento retransmitido no se debe ocupar para el cálculo del RTT ni para la actualización de la RTO, el mecanismo de *back-off* se restablece cada vez que se reconoce un segmento con SN mayor que  $r$ , es decir, si  $a > r$ , entonces  $n_{TO} = 0$ .<sup>1</sup>

Las variables  $a$ ,  $m$ ,  $X$ ,  $Y$  y  $A$  son acumulativas. El algoritmo se ejecuta hasta que  $A$  excede un valor que represente una conexión TCP de larga duración (por ejemplo, y según [39], en las pruebas se usa un valor de  $AR_ > 1e3$  s, normalmente  $AR_ > 1e5$  s).

Usando las figuras 3.2 y 3.3 como un ejemplo de un par de fragmentos de una conexión TCP de larga duración, el análisis en cada fase del protocolo es como sigue:

### 3.2.1. Fase CA.

De acuerdo con [15], a partir del valor  $w = \sigma$ ,  $w$  crece en un SMSS cada RTT durante la CA debido a la recepción de RAs. Esto sucede a partir de la ronda cero y hasta la ronda  $i$ , en la que ocurre la primera pérdida en esta fase, el  $\Delta$ -ésimo segmento. De la figura 3.2, se puede verificar que durante la CA, la cantidad de segmentos enviados hasta la ronda  $i$ , inclusive, está dada por  $\sum_{f=0}^i (w + f)$  donde  $w$  es el tamaño de la *cwnd* al inicio de la CA. Por lo tanto:

---

<sup>1</sup>Un RA con  $a = r$  forzosamente corresponde con el RA que termina la recuperación de una ventana de pérdida y se genera por una retransmisión exitosa.

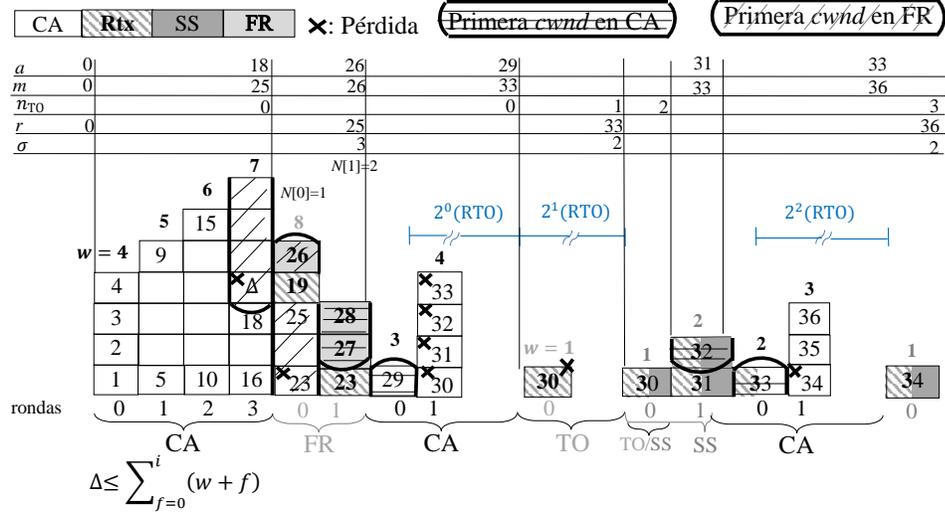


Figura 3.2: Fragmento de una conexión TCP NewReno: Caso 1.

$$\Delta \leq \sum_{f=0}^i (w + f) = w(i + 1) + \frac{1}{2}(i + 1)i \quad (3.4)$$

A partir de (3.4) se puede despejar la ronda  $i$  en la cual ocurre el LE (ver paso 4 del Algoritmo CA). De la figura 3.2, es claro que las variables  $a$  y  $w$  se actualizan al terminar la CA como  $w = w + i$  y  $a = a + \Delta - 1$ . En la primera CA de la figura 3.2,  $i = 3$ ,  $\Delta = 19$  y las variables  $a$  y  $w$  quedan como  $a = 18$  y  $w = 7$ . La variable  $\delta$  almacena la cantidad de pérdidas en la LW, por lo que  $\delta = 2$  en este caso. Las variables  $r$  y  $\sigma$  no se modifican al pasar por la fase CA.

La variable  $m$  corresponde al SN transmitido máximo y se actualiza como  $m = a + w$ , con los valores de  $a$  y  $w$  ya actualizados, siempre y cuando este valor exceda al valor actual de la variable. Si no lo excede, entonces el valor de  $m$  se conserva. Este caso significa que la fase termina sin intentar la recuperación completa de un LE anterior, es decir, que aún existen retransmisiones pendientes.

La cantidad de segmentos enviados durante la CA es la suma de los  $\Delta - 1$  segmentos previos a la primera pérdida más los segmentos de la LW, si se trata del total de segmentos enviados en la fase, o más la cantidad de segmentos no perdidos dentro de la LW (i.e.  $w - \delta$ , con  $w$  ya actualizada), si se consideran los segmentos enviados exitosamente. De forma que las variables  $X$  y  $Y$  se actualizan como  $X = X + \Delta - 1 + w - \delta$  y  $Y = Y + \Delta - 1 + w$ .

### *Transición CA – FR.*

Según [14], una pérdida se detecta en TCP NewReno por cualquiera de dos eventos: TD o TO. El evento TD equivale a que se cumpla la condición  $w - \delta \geq 3$ , con el valor de  $w$  ya actualizado. Si ocurre un TD, la recuperación de las pérdidas del LE será efectuada a través de una FR sólo si todos los LEs previos han sido recuperados completamente, es decir si se cumple con la condición  $a > r$ , con el valor de  $a$  ya actualizado.<sup>2</sup> Esta última condición permite además que se restablezca  $n_{TO} = 0$  y permite la actualización de la variable  $m$ .

Siguiendo con el ejemplo de la figura 3.2, la primera CA transita hacia la fase FR debido a que ambas condiciones anteriores son satisfechas. En este caso, la fase se extiende hasta la ronda  $i + 1$ , con la transmisión del segmento 25, por lo que  $m = a + w = 25$ . En esta misma ronda comienza la FR con la retransmisión del segmento 19. Como una aproximación, en el algoritmo se supone que las fases duran un número entero de rondas (excepto cuando la siguiente fase es un TO) y que las fases no se traslapan. Por ello, en este caso, se considera que la CA cubre hasta la ronda  $i$  y que la fase FR comienza en la siguiente ronda. De este caso se deriva que la duración de la CA, cuando transita a la FR, se calcula como  $i + 1$ . Es necesario notar que si el  $\Delta$ -ésimo segmento estuviera ubicado al final de la *cwnd* en la ronda  $i$ , el TD no sucedería sino hasta con el DA de un segmento enviado en la ronda  $i + 1$  y la FR comenzaría en la ronda  $i + 2$ , por lo que la duración de la CA sería  $i + 2$ . Sin embargo, para el algoritmo, se toma como caso general el primero (con la duración de la CA como  $i + 1$ ), debido a la poca probabilidad de ocurrencia del segundo caso (i.e. de que la primera pérdida ocurra en los últimos tres segmentos de una ventana), sobre todo para valores grandes de  $w$ .

### *Transición CA – TO.*

Si no se cumplen ambas condiciones que activan la FR, es decir, si sucede que  $w - \delta < 3$  ó  $a \leq r$ , entonces la CA transita a la fase TO para comenzar con la recuperación del LE.

Por ejemplo, en la segunda CA de la figura 3.2, el LE se presenta en la ronda  $i = 1$ , con

---

<sup>2</sup>En realidad, si  $a = r$ , ya se recuperaron todos los errores de las LWs anteriores (variante *Less careful*) [25], pero en la variante *Careful* del estándar RCF 2582 [25], incorporada como parte del nuevo estándar [61], se indica que la recuperación total de los LEs previos ocurre cuando se reconoce al menos un byte más que *recover*. Esta es la condición que se usa en el algoritmo. Adicionalmente, en el algoritmo se considera que la variable  $a$  indica el máximo SN recibido exitosamente por el receptor TCP, por lo que esta condición se traduce en  $a > r$ ; pero en el estándar [61]  $a$  indica al siguiente SN esperado, por lo que si se usara así la definición de  $a$ , la condición sería  $a - 1 > r$ .

$\Delta = 4$  and  $\delta = 4$ . Los valores de las variables con que se inicia esta CA son  $w = 3$ ,  $a = 26$  y  $r = 25$  (ver explicación de la fase FR). Para salir de la CA, éstas se actualizan como  $w = w + i = 3 + 1 = 4$  y  $a = a + \Delta - 1 = 27$ . Como  $a > r$ , se actualizan las variables  $n_{TO} = 0$  y  $m = 33$ . Por otro lado,  $w - \delta = 4 - 4 < 3$ . Consecuentemente, un TO ocurre una RTO después del último RA (correspondiente al segmento 29) recibido al final de la ronda  $i$ . En este caso, la cantidad de segmentos enviados exitosamente se refiere únicamente a los tres segmentos enviados antes del LE, puesto que el LE es de tipo ráfaga, es decir, que abarca toda la LW.

En el caso de la tercera CA de la figura 3.2, ésta se inicia con los valores  $w = 2$ ,  $a = 31$  (ver la explicación de la fase SS) y  $r = 33$  (ver la explicación de la fase TO). El LE se presenta en la ronda  $i = 1$ , con  $\Delta = 2$  and  $\delta = 1$ . Las variables se actualizan como  $w = w + i = 3$  y  $a = a + \Delta - 1 = 33$ . Puesto que ninguna de las dos condiciones necesarias para activar una FR se cumple, es decir, como  $w - \delta = 3 - 1 < 3$  y  $33 \leq 33$ , esta fase también termina en un TO. La cantidad de segmentos transmitidos exitosamente equivale a los  $\Delta - 1 + w - \delta$  segmentos transmitidos antes del LE y no perdidos dentro de la LW. La condición  $a \leq r$  significa que aún no se han recuperado por completo los LEs previos, sin embargo en esta fase se intentó concluir con las retransmisiones pendientes (hasta la del segmento 33), por lo que se actualiza  $m = 36$ . Adicionalmente, como consecuencia de la condición  $a \leq r$ , el mecanismo de *back-off* permanece activo (i.e. el factor  $n_{TO}$  no se modifica). Así, el TO sucede en un tiempo  $2^{n_{TO}}$  RTO, con el valor actual de  $n_{TO}$ , después del último RA, el cual se recibe al final de la ronda  $i$ . Generalizando los casos de la segunda y tercera CAs en la figura 3.2, la duración de la fase CA cuando transita a un TO se calcula como  $i$  rondas más un tiempo  $2^{n_{TO}}$  RTO (mientras este valor no exceda la RTO máxima,  $RTO_{max}$ )<sup>3</sup>. Otra posible expresión para la duración podría ser  $i + 1 + 2^{n_{TO}}$  RTO, la cual abarcaría los casos en los que el último RA que se reciba ocurra al final de la misma ronda  $i$ . Por ejemplo, en la tercera CA de la figura 3.2, esto ocurriría si  $\Delta = 5$ . En el algoritmo se usa el primer caso como general.

### 3.2.2. Fase TO.

El temporizador de retransmisión se actualiza con cada RA recibido [11], incluyendo los PAs (variante “lento pero firme” de TCP NewReno [14]). En el emisor TCP, un evento de expiración del temporizador, TO, se interpreta como la indicación de que el primer segmento aún no reconocido se ha perdido, por lo que se inicia con la recuperación del LE a través del siguiente algoritmo:

<sup>3</sup>Según el RFC 2988 [11], el temporizador está sujeto a un máximo de al menos 60 s.

- TO.1. Se actualizan las variables  $r = m$ , la  $cwnd$  se reduce a la  $lwnd$ , que equivale a 1 SMSS [15], es decir,  $w = 1$ .
- TO.2. Se retransmite el segmento perdido.
- TO.3. Se incrementa el factor de *back-off*,  $n_{TO} = n_{TO} + 1$ , de forma que un siguiente TO asociado con una retransmisión pendiente o incluso con el segmento inmediato posterior a la LW (ver el caso la tercera CA en la figura 3.2), tardará el doble en generarse. Se reinicia el temporizador [11].
- T.O.4. Si es el primer TO para un cierto SN, entonces  $\sigma = \max(2, \lfloor \frac{w}{2} \rfloor)$ , considerando el valor de  $w$  a la entrada de la fase [15].
- T.O.5. Si la retransmisión es exitosa, se restablece la variable  $\delta = 0$ <sup>4</sup> y se continúa la transmisión a partir de ella usando un SS siguiente. Esta retransmisión exitosa es, en realidad, el primer segmento transmitido en el SS, es decir, la transmisión en su primera ronda, cuando  $w = 1$ . Por lo tanto, esta retransmisión no se refleja en las variables globales sino hasta salir del SS.
- T.O.6. Si la retransmisión falla, un nuevo TO ocurre (TOs sucesivos) después de un tiempo igual a la RTO ponderada por el mecanismo de *back-off*, y se recomienza con el paso TO.2. En este caso, sólo un segmento es enviado en la fase (la retransmisión fallida), de forma que  $Y = Y + 1$ , y la duración de la fase es de  $2^{n_{TO}}$  RTO rondas, siempre y cuando este valor no exceda la  $RTO_{max}$  [11]. Las variables  $a$  y  $m$  no se alteran. La actualización de la variable  $\delta = 1$  es opcional, pues este valor no se usan más adelante.

La variable  $\Delta$  no tiene sentido en esta fase, pues si ocurre un nuevo LE, éste se debe siempre a que la retransmisión falla.

En el caso de ocurrir un TO sucesivo, es decir, un TO subsecuente para el mismo SN, la actualización de las variables  $r$  y  $w$  en el paso TO.1 no altera los valores que éstas adoptaron en el primer TO para este SN. Por lo tanto, el algoritmo anterior se refina incluyendo el paso TO.1 dentro de la condición del paso TO.4, después de la actualización de la variable  $\sigma$ .

Como ejemplo de lo anterior considérese el caso de la figura 3.2. La segunda CA concluye con un LE, cuya LW abarca desde el segmento 30 al 33, todos ellos perdidos. Consecuentemente, ningún DA se genera y termina por producirse un TO. Al finalizar esta CA,  $m = 33$  y

---

<sup>4</sup>En el algoritmo,  $\delta$  no se actualiza al finalizar el TO, porque este valor no se usa más adelante.

$w = 4$  (ver la explicación de la fase CA del algoritmo). Se retransmite el segmento 30 (paso TO.2) y se activa el mecanismo de *back-off* con  $n_{TO} = n_{TO} + 1$  (paso TO.3). Como es el primer TO asociado con el SN = 30, se actualizan las variables  $\sigma = \max(2, \lfloor \frac{w}{2} \rfloor) = \lfloor \frac{4}{2} \rfloor = 2$ ,  $r = m = 33$  y  $w = 1$  (paso TO.4, que incluye al paso TO.1). Como la retransmisión falla (paso TO.6), la fase termina después de  $2(RTO)$  rondas con un TO consecutivo. Se ejecutan de nuevo los pasos TO.2 y TO.3. La retransmisión es exitosa, por lo que el TO transita al SS. La retransmisión exitosa se considera como la primera transmisión del SS siguiente, así que la única variable que se ve alterada en el TO consecutivo es  $n_{TO} = 2$ .

En el caso de la tercera CA de la figura 3.2, la LW abarca el segmento 34 al 36. Del LE anterior,  $r = 33$ . La condición impuesta en [11] establece que el mecanismo de *back-off* se restablece sólo si se reconoce un SN mayor que  $r$ , es decir, cuando  $a > r$ . Por ello, aunque estrictamente el nuevo LE no se traslapa con el LE anterior, el mecanismo de *back-off* sigue activo, pues con  $a = r = 33$ , esta condición no se cumple en este caso. De forma que el TO siguiente se produce en un tiempo  $2^2(RTO)$ .  $w = 3$  y  $n = 36$  al finalizar esta CA (ver la explicación de la fase CA del algoritmo). En la fase TO, se retransmite el segmento 34 (paso TO.2) y se actualiza  $n_{TO} = 3$  (paso TO.3). Como es el primer TO asociado con el segmento 34, se actualizan las variables  $\sigma = \max(2, \lfloor \frac{w}{2} \rfloor) = \max(2, \lfloor \frac{3}{2} \rfloor) = 2$ ,  $r = m = 36$  y  $w = 1$  (paso TO.4, que incluye al paso TO.1). La retransmisión es exitosa, por lo que el TO transita al SS.

### 3.2.3. Fase SS.

Durante el SS,  $w$  se incrementa en un SMSS por cada RA <sup>5</sup> a partir de la *iwnd* y mientras que  $w < \sigma$ . Se considera que el tamaño de la *iwnd* es de un SMSS <sup>6</sup>. Como se observa en la figura 3.3, con la recepción del RA del  $n$ -ésimo segmento enviado en esta fase, el valor de  $w$  se actualiza a  $n + 1$ . De forma que, si la transmisión de todos los segmentos dentro de la *cwnd* es exitosa, entonces  $w$  se duplica al inicio de la siguiente ronda (i.e.  $w$  crece exponencialmente, con base 2, en cada ronda). Por lo tanto, la ronda en la que se transmite el  $n$ -ésimo segmento se puede calcular como  $\lceil \log_2 n \rceil$ .

Cada ocasión que un TO ocurre, se comienza con la retransmisión del segmento que provocó el TO. Si la retransmisión es exitosa, entonces ésta se considera como la primera transmisión

<sup>5</sup>En el RFC 5681 [15] se menciona que este incremento es el máximo que se puede tener. Para efectos del algoritmo, sin embargo, no existen las condiciones que pudieran generar incrementos menores.

<sup>6</sup>En el estándar más reciente para TCP Reno, RFC 5681 [15], se establece que el tamaño de la *iwnd* debe ser 2, 3 ó 4 SMSS según el tamaño del SMSS. En el algoritmo no se incluye esta modificación del estándar. Sin embargo, ajustarlo no supone gran complicación.

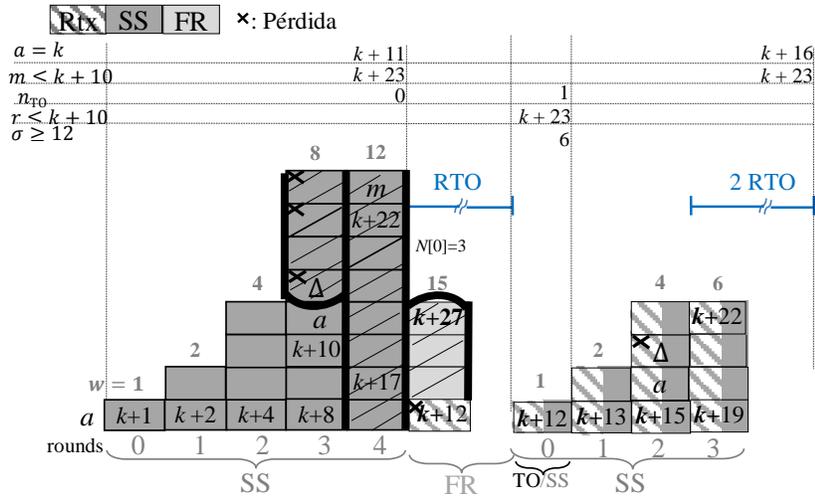


Figura 3.3: Fragmento de una conexión TCP NewReno: Caso 2.

dentro del SS que prosigue al TO, el cual comienza con un tamaño de ventana igual al valor de la  $lwnd$ , cuyo valor es de 1 SMSS [15].

Al igual que sucede en la CA, las variables  $r$  y  $\sigma$  no se modifican al pasar por la fase SS y el  $n_{TO} = 0$  siempre y cuando las retransmisiones faltantes sean exitosamente enviadas durante el SS (i.e.  $a > r$  al salir del SS), de otra forma el mecanismo de *back-off* sigue activo con el mismo valor de  $n_{TO}$ .

Si no existe ninguna pérdida en el SS, éste termina con una transición SS - CA. Por el contrario, y en contraste con los modelos actuales para el THR de TCP NewReno, la ejecución del modelo de pérdidas estando en SS puede dar como resultado la pérdida del  $\Delta$ -ésimo segmento, con  $1 < \Delta < \sigma$ .<sup>7</sup> En este caso, el SS terminará con una transición SS - FR o SS - TO, como se explica más adelante.

#### Transición SS – CA.

El estándar marca que el SS se termina cuando se cumple que  $w \geq \sigma$ . Específicamente, en el algoritmo se considera que esta transición ocurre al recibir al primer RA siendo  $w = \sigma$ . Como se observa en la figuras 3.2 y 3.3, el valor de  $w$  en cada instante coincide con el número

<sup>7</sup>Para el algoritmo, el SS sigue a una fase TO cuya retransmisión es exitosa. Esta retransmisión se toma como el primer segmento enviado en el SS. De forma que la primera pérdida en el SS será cualquier segmento posterior a dicha retransmisión no fallida ( $\Delta > 1$ ) y menor que la máxima cantidad de segmentos que pueden enviarse sin pérdida en un SS antes de transitar a la CA ( $\Delta < \sigma$ ).

del segmento al inicio de la *cwnd*. Por lo tanto, la transición SS - CA ocurre al recibir el RA del  $\sigma$ -ésimo segmento<sup>8</sup>, hacia el final de la ronda  $i = \lfloor \log_2 \sigma \rfloor$  (ver el SS de la figura 3.2). Consecuentemente, un SS sin pérdidas dura  $i + 1$  rondas y concluye con  $w = \sigma$  y  $a = a + \sigma$ . Por ejemplo, el SS de la figura 3.2 comienza después de una serie de TOs sucesivos, con los valores  $a = 29$ ,  $r = m = 33$  y  $\sigma = 2$  (ver la explicación de la fase TO). Por lo tanto, el SS termina en la ronda  $i = 1$ , con los valores de  $w = 2$ ,  $a = 31$  y su duración es de  $i + 1$  rondas.

La variable  $m$  almacena el SN transmitido máximo. En el ejemplo de la figura 3.2,  $m = 33$  al inicio del SS, es decir, el  $m$  apunta al último segmento de la LW anterior, de tamaño 4. Después del TO, la recuperación de la LW se comienza con el SS. Si  $\sigma$  es menor que el tamaño de la LW, entonces el SS transitará hacia la CA antes de que se haya recuperado por completo los LEs previos. En este ejemplo, el SS transita a la CA al recibir el RA del segmento 31, es decir, antes de realizar todas las retransmisiones faltantes (hasta la del segmento  $m = 33$ ), por lo que la variable  $m$  se queda igual al salir del SS. Generalizando, si al concluir una fase no se han completado todas las retransmisiones faltantes (i.e. el SN transmitido máximo en una fase es menor que el valor actual de la variable  $m$ ), entonces  $m$  no se modifica al pasar por esta fase.

Todos los segmentos transmitidos en la ronda  $i$ , posteriores al  $\sigma$ -ésimo segmento, en realidad se enviaron durante el SS. Sin embargo, como se mencionó en la sección 3.2.1, en el algoritmo se considera que la CA comienza con la transmisión de  $w = \sigma$  segmentos. Por esta razón, todos los segmentos posteriores al  $\sigma$ -ésimo se toman como parte de la primera *cwnd* transmitida en la CA siguiente. En el caso de la figura 3.2, el segmento 32, que en realidad fue enviado en la segunda ronda del SS, se supone como el primer segmento enviado en la ronda cero de la CA siguiente. Por lo tanto, al final de un SS sin pérdidas, se considera que el total de segmentos enviados, todos exitosamente, es igual a  $\sigma$  (i.e.  $m = \max(a, m)$ ,  $X = X + \sigma$  y  $Y = Y + \sigma$ ).

#### *Transiciones SS - FR y SS - TO.*

Según [61], al igual que en la CA, en caso de existir un LE en el SS ( $\delta > 0$ ), las condiciones para poder recuperarlo a través de una FR son:  $w - \delta \geq 3$  y  $a > r$  (condición que asegura que se han recuperado por completo los LEs anteriores).<sup>9</sup> Si alguna de estas condiciones no se cumple, la pérdida se recuperará a través de un TO.

<sup>8</sup>El RA del  $\sigma$ -ésimo segmento ya no produce un incremento en  $w$ , porque ocurre cuando  $w = \sigma$ .

<sup>9</sup>Ver nota al pie 2.

En cualquiera de ambos casos, el SS termina con la transmisión de la LW, la cual comienza con el  $\Delta$ -ésimo segmento y es de tamaño  $\Delta$ . La variable  $\delta$  almacena la cantidad de pérdidas en la LW. La variable  $a$  termina apuntando al segmento anterior a la primera pérdida. De forma que las variables  $a$  y  $w$  se actualizan como  $w = \Delta$  y  $a = a + \Delta - 1$ .

La variable  $X$  se actualiza con la cantidad de segmentos enviados exitosamente, que es la suma de los  $\Delta - 1$  segmentos antes del LE mas los  $\Delta - \delta$  segmentos no perdidos de la LW. Para la variable  $Y$  se toman en cuenta además las pérdidas en la LW, de forma que  $Y = Y + 2\Delta - 1$ .

La variable  $m$  equivale a  $a + w$ , con los valores de  $a$  y  $w$  ya actualizados, siempre y cuando este valor exceda el valor actual de la variable. Si no lo excede, entonces el valor de  $m$  se conserva. Este caso significa que la fase termina sin recuperar por completo un LE anterior, es decir, que aún existen retransmisiones pendientes.

De nuevo, debido al crecimiento exponencial (con base 2) de  $w$  en cada ronda, la ronda en la que ocurre el LE puede calcularse como  $i = \lfloor \log_2 \Delta \rfloor$ .

Lo anterior queda claro si se observa la figura 3.3 en la cual el primer SS comienza con los valores  $a = k$ ,  $r = m < k + 10$ ,  $\sigma \geq 12$  and  $w = 1$ , donde  $k$  es un número cualquiera dentro de la secuencia de segmentos del flujo TCP. El LE se presenta con la pérdida del  $\Delta$ -ésimo segmento ( $\Delta = 12$ ), con  $\delta = 3$ . Las variables se actualizan como  $w = \Delta = 12$ ,  $a = a + \Delta - 1 = k + 11$  y  $X = X + 2\Delta - 1 - \delta$ . Como  $k + 23$  es mayor que el valor actual de la variable  $m$ , ésta se actualiza con este valor.

Si se cumple con las condiciones  $w - \delta \geq 3$  y  $a > r$ , con los valores de  $w$  y  $a$  ya actualizados, entonces la pérdida es factible de recuperarse mediante una FR. Como  $a > r$ , esto permite que se restablezca  $n_{TO} = 0$ . Tal es el caso del primer SS en la figura 3.3. En éste, el TD se alcanza con el DA provocado por el segmento  $K + 17$ . Esto ocurre en la ronda  $i + 1$ . De forma que la duración del SS es de  $i + 2$  rondas. Si las condiciones del LE hubieran sido de tal forma que el TD ocurriera hacia el final de la ronda  $i$ , es decir, por un segmento enviado en la misma ronda que la pérdida, entonces la duración del SS sería  $i + 1$ . Para el algoritmo, el primer caso es considerado como el caso general, lo que corresponde con valores relativamente pequeños de  $\sigma$ , que implican valores de  $w$  relativamente pequeños dentro del SS.

Por otro lado, si ocurre un LE tal que el SS termine con  $w - \delta < 3$  ó  $a \leq r$ , entonces éste se

comenzará a recuperar con un siguiente SS después de uno o más TOs sucesivos. La duración del SS es la suma de la RTO, ponderada según el mecanismo de *back-off*, más el tiempo en el que se recibe el último RA. De forma similar a la que se examinó en la CA, existen dos casos de análisis para la duración de un SS que termina con una transición SS – TO. Si el LE coincide con el inicio de la ronda  $i$ , entonces el último RA se recibe al final de la ronda  $i - 1$  y la duración del SS es  $i + 2^{n_{\text{TO}}}$  RTO. En cualquier otro caso, el último RA corresponde a un segmento enviado en la ronda  $i$  y la duración es  $i + 1 + 2^{n_{\text{TO}}}$  RTO. Para el algoritmo, se toma el segundo como caso general. Es decir, para el algoritmo el primer caso se toma en cuenta en la CA y el segundo en el SS, con la intención de compensar el error que ambas aproximaciones introducen al cálculo del THR.

Como ejemplo de lo anterior considérese el segundo SS en la figura 3.3, el cual comienza con los valores  $a = k + 11$ ,  $r = m = k + 23$ ,  $\sigma = 6$  y  $n_{\text{TO}} = 1$  (ver la explicación de la fase FR). El LE ocurre con el segmento  $\Delta = 6$ , en la ronda  $i = 2$ , con  $\delta = 1$ . Consecuentemente, el SS finaliza con  $w = 6$  y  $a = k + 16$ . Aunque existe un TD (es decir, aunque  $w - \delta = 6 - 1 \geq 3$ ), la condición  $a \leq r$  evita que una nueva FR se active, por lo que el SS transita a un TO. Como  $a \leq r$ ,  $n_{\text{TO}}$  permanece sin cambio. Esta fase concluye sin enviar las retransmisiones faltantes (en este caso hasta el segmento  $k + 23$ ), por lo que  $m$  no se actualiza. El último RA, recibido al final de la ronda  $i$ , corresponde al segmento  $k + 16$ . Así que la duración de la fase es de  $i + 1 + 2^{n_{\text{TO}}}$  RTO =  $3 + 2(\text{RTO})$  rondas.

### 3.2.4. Fase FR.

En [15] se especifica que cuando un receptor TCP recibe un segmento fuera de orden debe emitir inmediatamente un DA, y cuando recibe un segmento que completa parcial o totalmente un espacio en la secuencia de segmentos recibidos debe responder con un RA. Si un transmisor TCP recibe un TD,<sup>10</sup> lo interpreta como una indicación de un LE, el cual comienza con la pérdida del segmento indicado por los DAs. Si se cumple que el máximo SN reconocido abarca más que el *recover* [61], es decir, si  $a > r$ ,<sup>11</sup> entonces las  $\delta$  pérdidas se intentarán recuperar con la FR de NewReno.

Así, el algoritmo transita a la fase FR desde una fase SS o CA que termina con un LE, es

<sup>10</sup>En [15] existe la sugerencia de enviar un segmento al recibir el primer y el segundo DA. Éstos segmentos pueden servir para completar los tres DAs, si es que no se tenían. Por el momento, esta sugerencia no se incluye en el análisis.

<sup>11</sup>Ver nota al pie 2.

decir, que concluye con la emisión de una LW, con  $\delta$  pérdidas y  $w - \delta$  segmentos no perdidos (cada uno de los cuales provocará la emisión de un DA en el receptor). Por ejemplo, en la figura 3.2, la FR se presenta después de una CA que concluye con un LE. La LW abarca desde el segmento 19 hasta el 25 y contiene  $\delta = 2$  pérdidas: los segmentos 19 y 23. Como se muestra en la figura, la FR comienza con los valores  $a = 18$ ,  $m = 25$ ,  $w = 7$  y  $n_{TO} = 0$ .<sup>12</sup>

Para el análisis del mecanismo de FR de NewReno se retoman las suposiciones 4 y 5 de la sección 2.4.2, derivadas del modelo en [37], que consisten en lo siguiente:

**Suposición 2.4.2-4:** Cualquier LE dentro de la FR involucra la pérdida de una retransmisión, por lo que fuerza a la terminación temprana de la FR a través de un TO.

**Suposición 2.4.2-5:** El algoritmo se desarrolla para el mecanismo de FR de NewReno en su versión “lento pero firme” [14], en la que el temporizador se restablece con cada PA, de forma que si todas las retransmisiones son exitosas entonces la FR durará tantos RTTs como sea el número de pérdidas en el LE,  $\delta$ .

De estas suposiciones se desprende que:

- a) Si no hay ningún LE en la FR, ésta dura un total de  $\delta$  rondas,  $f = 0, \dots, \delta-1$ , al cabo de las cuales terminará por completo la recuperación de las  $\delta$  pérdidas del último LE. Por ejemplo, en la figura 3.2, la FR dura  $\delta = 2$  rondas,  $f = 0, 1$ . La primera ronda comienza con la retransmisión del segmento 19 y la segunda con la del segmento 23.
- b) El algoritmo permanece en la FR mientras no haya ninguna nueva pérdida, de forma que en cada ronda  $f = 1, \dots, \delta-1$ , la cantidad de DAs recibida corresponde a los  $N[f - 1]$  segmentos transmitidos en la ronda anterior, mientras que en la primer ronda,  $f = 0$ , se reciben  $w - \delta$  DAs provocados por los segmentos no perdidos en la LW. En la figura 3.2, se comienza la FR con la llegada de  $w - \delta = 7 - 2$  DAs.

El mecanismo FR representado en el algoritmo consta de los siguientes pasos:

FR.1. Según [15], se retransmite el segmento supuesto como perdido y se actualizan las variables  $r = m$ ,  $\sigma = \max(2, \lfloor \frac{w}{2} \rfloor)$  y  $w = \sigma$ .<sup>13</sup> En el caso de la figura 3.2,  $r = 25$  y  $\sigma = \lfloor \frac{7}{2} \rfloor = 3$ .

<sup>12</sup>Como un requisito para activar la fase FR es que  $a > r$ , siempre al inicio de una FR se tendrá que  $n_{TO} = 0$ .

<sup>13</sup>En [15] se especifica que la actualización de  $\sigma$  debe ser como  $\sigma = \max(\text{FlightSize}/2, 2 * S MSS)$ . Sin embargo, en el modelo se usa el valor de  $w$  en lugar del *FlightSize*, por la dificultad de saber este valor. Bajo las suposiciones en las que se basa el modelo, particularmente: transmisión continua, transmisión en ráfaga en cada ronda y  $rwnd \gg cwnd$ , se considera que esta aproximación es aceptable.

FR.2.  $w$  se aumenta temporalmente en un SMSS por cada DA recibido (incluyendo el TD que activó la FR). Al entrar a la FR y como resultado de los pasos FR.1 y FR.2,  $w$  se actualiza como  $w = \text{máx}(2, \lfloor \frac{w}{2} \rfloor) + w - \delta$ . Por ejemplo, en la FR de la figura 3.2,  $w$  se actualiza a  $w = \lfloor \frac{7}{2} \rfloor + 7 - 2 = 8$ . Si hay más datos por transmitir (que es el caso en el algoritmo, por la suposición de transmisión continua) y si la *cwnd* lo permite, se transmiten [15]. Así, en la primer ronda dentro de la FR,  $f = 0$ , la cantidad de segmentos transmitidos,  $N[f = 0]$ , equivale al nuevo tamaño de la *cwnd* menos los segmentos de la LW (que se transmitieron al final de la fase previa), es decir,  $N[f = 0] = \text{máx}(2, \lfloor \frac{w}{2} \rfloor) + w - \delta - w = \text{máx}(2, \lfloor \frac{w}{2} \rfloor) - \delta$ . Si este resultado es negativo, significa que hay más transmisiones en curso de las que la *cwnd* actualizada lo permite. En la figura 3.2, se remarca la ubicación de la *cwnd* al inicio de la ronda  $f = 0$  de la FR. Como se puede observar, la *cwnd* comienza con el segmento perdido que dio origen al LE en la fase anterior, el segmento 19, y termina con el segmento 26. Es decir, después de la retransmisión del segmento 19, en esta ronda se permite transmitir  $N[f = 0] = \lfloor \frac{7}{2} \rfloor - 2 = 1$  segmentos adicionales, el segmento 26.

FR.3. En cada ronda subsecuente dentro de la FR,  $f = 1, \dots, \delta - 1$ , si se recibe un PA: 1) se retransmite el primer segmento no reconocido aún, 2) la *cwnd* se disminuye por la cantidad de datos nuevos reconocidos, se incrementa en un SMSS (lo que refleja la retransmisión exitosa) y se desplaza hacia la derecha para comenzar con el siguiente segmento sin reconocer [61]. Como resultado de los pasos FR.2 y FR.3, y retomando las conclusiones a) y b) mencionadas arriba, la cantidad de segmentos a transmitirse en la ronda  $f = 0, \dots, \delta - 1$ , puede calcularse como  $N[f] = \text{máx}(0, \sigma - \delta + f)$ . Es decir, al inicio de la ronda  $f = 1$ ,  $w$  se incrementa en un MSS por el PA y por cada uno de los  $N[0] = \sigma - \delta$  DAs recibidos al final de la ronda cero. En la ronda  $f = 2$ , los  $N[1] = \sigma - \delta + 1$  DAs correspondientes a los segmentos transmitidos en la ronda anterior más un PA provocan que  $N[2] = \sigma - \delta + 2$  segmentos nuevos tengan permiso de ser transmitidos, y así consecutivamente hasta la ronda  $f = \delta - 1$ . Como se mencionó en el paso FR.2, si resulta un valor negativo, esto implica que en realidad hay más segmentos en tránsito que los que permitiría la *cwnd*, de forma que en realidad no se transmite ningún segmento nuevo. Es por ello que se añade la función  $\text{máx}(0, )$  en el cálculo de  $N[f]$ .

Por ejemplo en la FR de la figura 3.2, la primer ronda termina con la recepción del PA (generado por la retransmisión exitosa del segmento 19) y del DA provocado por el segmento 26 (i.e.  $N[0] = 1$ ). Por ello, al inicio de la ronda siguiente, la *cwnd* se

disminuye en 4 (por el reconocimiento de los segmentos 19 al 22), se incrementa en 2 (por los  $N[0]$  DAs y el PA) y se desplaza para comenzar con el siguiente segmento sin reconocer, el segmento 23, cubriendo entonces hasta el segmento 28. Es decir, en esta ronda se permite la transmisión de  $N[1] = \text{máx}(0, \sigma - \delta + f) = 3 - 2 + 1 = 2$  segmentos nuevos, los segmentos 27 y 28.

FR.4. La llegada de un RA total (i.e. con  $a \geq r$  indica que la retransmisión de las  $\delta$  pérdidas del último LE se ha concluido exitosamente. En este caso,  $w$  se actualiza como  $w = \sigma$  y la FR transita hacia la fase CA [61].<sup>14</sup> El RA total se recibe al finalizar la ronda  $f = \delta - 1$ , en la que se transmitieron  $N[\delta - 1] = \text{máx}(0, \sigma - \delta + \delta - 1) = \sigma - 1$  segmentos. El último segmento en la *cwnd* actual ( $w = \sigma$ ), se transmite en la ronda siguiente, que es la primera de la CA. De forma semejante a como se hizo en el análisis del SS, con el fin de cumplir con la consideración de que una fase CA comienza con la transmisión de  $w = \sigma$  segmentos, los  $N[\delta - 1]$  se toman como si fueran transmitidos en la ronda cero de la siguiente CA. Por lo tanto, la cantidad total de segmentos enviados durante la FR equivale a  $\sum_{f=0}^{\delta-2} N[f]$ . De acuerdo con la suposición 4 de la sección 2.4.2, ninguno de estos segmentos se perdió, por lo que, al finalizar la FR,  $X$  y  $Y$  se actualizan como  $X = X + \sum_{f=0}^{\delta-2} N[f]$  y  $Y = Y + \sum_{f=0}^{\delta-2} N[f]$ , se restablece  $\delta = 0$ <sup>15</sup> y las variables  $a$  y  $m$  son equivalentes y se calculan como  $m = a = m + \sum_{f=0}^{\delta-2} N[f]$ . Por ejemplo en la FR de la figura 3.2, se actualizan las variables  $m = a = m + N[0] = 25 + 1 = 26$ .

FR.5. Según las suposiciones 4 y 5 de la sección 2.4.2, si se pierde algún segmento transmitido en la FR, sea retransmisión o no, la FR termina con un TO. Específicamente, en el algoritmo esta situación se modela como si el LE sucediera con la primer retransmisión. Es decir, los segmentos transmitidos dentro de la FR se desprecian y la duración de la FR se aproxima como una RTO.<sup>16</sup> Por ejemplo, en la figura 3.3, el primer SS transita a una FR para la recuperación de la LW de tamaño  $w = 12$ , que abarca del segmento  $k + 12$  al  $k + 23$  y tiene  $\delta = 3$  pérdidas. La FR comienza con la retransmisión el segmento  $k + 23$ , seguida de  $N[0] = \lfloor \frac{12}{2} \rfloor - 3 = 3$  segmentos (del  $k + 24$  al  $k + 27$ ). La retransmisión falla. De acuerdo a la simplificación propuesta para el algoritmo, al presentarse un LE en la fase FR, ésta transita a un TO, siendo  $A = A + RTO$  la única variable actualizada al salir de la FR.

<sup>14</sup>En [61] se definen dos opciones para la actualización de  $w$  al salir del FR. En el algoritmo se usa la segunda de ellas.

<sup>15</sup>En el algoritmo,  $\delta$  no se actualiza al finalizar la FR, porque este valor no se usa más adelante.

<sup>16</sup>No sobra recordar que: si  $a > r$  el mecanismo el *back-off* se restablece; por lo que  $n_{TO} = 0$  siempre al comenzar un FR.

### 3.3. Modelo de pérdidas de la red.

El estado de pérdida de los segmentos del flujo TCP está determinado por el comportamiento de pérdidas de la red, el cual se representa mediante un modelo.

En redes cableadas, los errores en el medio de transmisión pueden ser despreciables, de forma que la principal causa de pérdidas de segmentos es la congestión en los nodos de la red [63]. Existe una gran cantidad de trabajos en el área del modelado de los procesos de pérdida de paquetes de datos en distintos tipos de redes, por ejemplo los modelos en [63] – [74]. Los modelos de Bernoulli y de Gilbert Simplificado, que son de los modelos más comúnmente usados para este propósito, son los que se usan como generadores de la secuencia de segmentos perdidos para evaluar el algoritmo propuesto.

#### 3.3.1. Modelo de Bernoulli.

El estado de pérdida de un paquete se supone independiente de los demás. El modelo tiene un solo parámetro: la tasa promedio de paquetes perdidos,  $p$ . Por su sencillez, este modelo ha sido ampliamente usado en los trabajos de análisis del desempeño de TCP como modelo de pérdidas de los segmentos en una red, tal como en [32], [37], [48], [49] y [50].

Aunque el modelo de Bernoulli es sumamente socorrido cuando se modela el comportamiento de pérdidas de una red, presenta una gran restricción: sólo puede representar pérdidas independientes, lo que resulta en un alejamiento importante de las características de los escenarios reales, pues en una red las pérdidas están asociadas principalmente al desbordamiento de las colas en los “nodos cuellos de botella”. Estos eventos de congestión implican una correlación entre las pérdidas, de forma que una vez que se presenta una pérdida (estado incipiente de congestión) es más probable que se presenten pérdidas adicionales. Por ello, en trabajos como [32], la distribución de Bernoulli se asocia no con las pérdidas de segmentos individuales sino con eventos de pérdidas tipo ráfaga. Aunque con esta suposición se intenta compensar en parte la limitante de este modelo, se recurre a una nueva restricción, la de considerar eventos de pérdida de ráfagas completas, lo cual no necesariamente es cercano a la realidad debido la condición dinámica y con múltiples elementos mezclados en un evento de congestión. En [37] se presenta otra propuesta para contrarrestar esta limitación del modelo de Bernoulli: se presenta un modelo propio para caracterizar eventos de pérdida con características de ráfaga. Este modelo usa una distribución de Bernoulli para la probabilidad de evento de pérdida y emplea una segunda VA tipo Bernoulli para la probabilidad de las pérdidas dentro de un

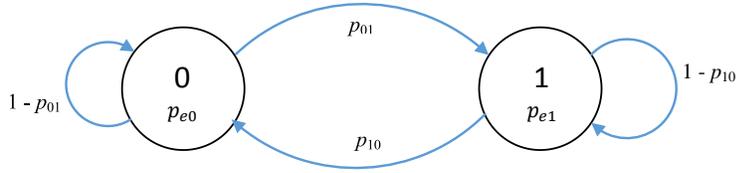


Figura 3.4: Modelo Simplificado de Gilbert para el proceso de pérdida de paquete.

LE. Con esta propuesta se intenta caracterizar de forma distinta las pérdidas dentro de un LE que entre LEs; sin embargo, las pérdidas dentro del LE siguen siendo, por definición de la distribución de Bernoulli, independientes. En el área de modelado de los procesos de pérdida de segmentos en una red de conmutación de paquetes existe una gran cantidad de modelos desarrollados para la caracterización de las pérdidas no independientes en una red, muchos de los cuales se basan en el modelo de Gilbert explicado a continuación.

### 3.3.2. Modelo de Gilbert simplificado.

En la figura 3.4 se muestra la representación del modelo de pérdidas de datos introducido por Gilbert [64] y Elliot [65] correspondiente a una cadena de Markov de primer orden, en la cual el estado de pérdida de un paquete depende del estado de pérdida del paquete anterior. Tiene dos estados: “Recepción” y “Pérdida”, y cuatro parámetros independientes:  $p_{01}$ ,  $p_{10}$ ,  $p_{e0}$  y  $p_{e1}$ . Aunque este modelo fue originalmente propuesto para representar la naturaleza de ráfaga de los errores de bit producidos en los canales ruidosos, en algunos trabajos como [66] se propone como modelo para capturar la dependencia temporal del proceso de pérdida de paquetes en redes de comunicaciones. A la fecha, su uso en el modelado del proceso de pérdida de los paquetes en redes de conmutación de paquetes se ha extendido, como se menciona en [63], [70] y las referencias citadas en ellas.

Es posible generar paquetes perdidos como eventos independientes en cada estado, con una tasa de pérdidas dependiente del estado:  $p_{e0}$  en el estado de “Recepción” y  $p_{e1}$  en el estado de “Pérdida” [70] [71]. El modelo de Gilbert simplificado se reduce al modelo de Bernoulli cuando  $p_{01} + p_{10} = 1$  [63]. Las probabilidades en estado estable de permanecer en los estados de “Recepción”,  $\pi_0$ , y de “Pérdida”,  $\pi_1$ , están dadas por [75].

$$\pi_0 = \frac{p_{10}}{p_{10} + p_{01}}; \quad \pi_1 = \frac{p_{01}}{p_{10} + p_{01}} \quad (3.5)$$

Mientras que la tasa de paquetes perdidos en estado estable,  $p$ , está dada por:

$$p = p_{e0}\pi_0 + p_{e1}\pi_1 \quad (3.6)$$

Gilbert consideró el caso especial de que el estado “Recepción” fuera libre de pérdidas (i.e.  $p_{e0} = 0$ ) e incluso propuso un modelo aún más simplificado obtenido con  $p_{e1} = \{1, 0, 5\}$ , como se describe en la tabla 3.2 [71]. La longitud promedio de una ráfaga de paquetes perdidos,  $B_L$ , se calcula como:

$$B_L = \frac{1}{p_{10}} \quad (3.7)$$

En el algoritmo, se usa el modelo simplificado de Gilbert para generar la secuencia de segmentos erróneos de una conexión TCP usando valores de los parámetros dentro del rango de aquellos calculados de mediciones en [74] y [66], y que han sido usados en trabajos como [67] y [68].

Tabla 3.2 Comparación de los modelos simplificados de tipo cadena de Markov de dos estados.

Modelo	Parámetros	Simplificación	Tasa de paquetes perdidos
Gilbert simplificado	$p_{01}, p_{10}$	$p_{e0} = 0, p_{e1} = \{1, 0, 5\}$	$p = \{\pi_1, \frac{1}{2}\pi_1\}$
Gilbert	$p_{01}, p_{10}, p_{e1}$	$p_{e0} = 0$	$p = p_{e1}\pi_1$
Elliot	$p_{01}, p_{10}, p_{e1}, p_{e0}$		$p = p_{e0}\pi_0 + p_{e1}\pi_1$

### 3.4. Algoritmo.

<p><b>PARÁMETROS DE ENTRADA:</b></p>	<p>RTT promedio, <math>R_</math>.</p> <p>RTO promedio, <math>RTO</math>, en rondas.</p> <p>RTO inicial, en rondas, <math>RTO_{ini}</math>.</p> <p>SMSS</p> <p><math>ssthresh</math> inicial, <math>\sigma_{ini}</math>.</p> <p>Duración típica de una conexión TCP NewReno de larga duración, <math>T</math>.</p> <p>Parámetros de entrada del módulo de pérdidas: tasa promedio de segmentos perdidos, <math>p</math>, si es un módulo Bernoulli o <math>p_{01}</math>, <math>p_{10}</math>, <math>p_{e0}</math> y <math>p_{e1}</math> si es un módulo Gilbert.</p>
<p><b>VARIABLES GLOBALES:</b></p>	<p>Duración de la conexión medida en rondas, <math>A</math>.</p> <p>SN recibido exitosamente máximo, <math>a</math>.</p> <p>Primer segmento perdido en la fase, <math>\Delta</math>.</p> <p>Cantidad de segmentos perdidos en la fase, <math>\delta</math>.</p> <p>Fase siguiente, FASE.</p> <p>SN transmitido máximo, <math>m</math>.</p> <p>Exponente de la base 2 con que se pondera a la RTO, <math>n_{TO}</math>.</p> <p>Variable <i>recover</i> de TCP NewReno, <math>r</math>.</p> <p><math>ssthresh</math>, <math>\sigma</math></p> <p>Tamaño de la <i>cwnd</i>, <math>w</math>.</p> <p>Cantidad de segmentos enviados exitosamente en el intervalo <math>A</math>, <math>X</math>.</p> <p>Cantidad de segmentos enviados en total en el intervalo <math>A</math>, <math>Y</math>.</p>
<p><b>INICIALIZACIONES:</b></p>	<p><math>a = m = r = X = Y = 0</math></p> <p><math>A = 0</math></p> <p>FASE = CA</p> <p><math>w = \text{mín}(\text{máx}(2, \frac{\Delta}{2}), \sigma_{ini})</math>.</p>
<p><b>PARÁMETROS DE SALIDA:</b></p>	<p>Tasa de transmisión efectiva en estado estable de la conexión TCP NewReno de larga duración, <math>THR_E</math>, en Mbps.</p> <p>Tasa de transmisión en estado estable de la conexión TCP NewReno de larga duración, <math>THR</math>, en Mbps.</p>
<p><b>CONDICIÓN DE PARO:</b></p>	<p><math>AR_ &gt; T</math></p> <p>.</p>

---

**ALGORITMO PARA LA EVALUACIÓN DEL THR EN ESTADO ESTABLE DE UNA CONEXIÓN TCP NEWRENO**

---

1. **while**  $AR_< \leq T$  **do**
  2.     **switch** FASE
  3.         **case** CA: **execute** ALGORITMO CA
  4.         **case** SS: **execute** ALGORITMO SS
  5.         **case** FR: **execute** ALGORITMO FR
  6.         **case** TO: **execute** ALGORITMO TO
  7.     **end switch**
  8. **end while**
  9. **return**  $THR_E \leftarrow \frac{X(SMSS)}{AR_<(1E6)}, THR \leftarrow \frac{Y(SMSS)}{AR_<(1E6)}$
- 

---

**ALGORITMO DE PÉRDIDAS**

---

<b>PARÁMETROS DE ENTRADA:</b>	Segmento inicial, $S_i$ . Segmento final, $S_f$ . Umbral.
<b>PARÁMETROS DE SALIDA:</b>	Número de segmento perdido, $\Delta$ .

1. **if** Umbral > 0 **then**
  2.     **for**  $s = S_i$  **to**  $S_f$  **do**
  3.         **if** se pierde el segmento  $s$  **then return**  $\Delta \leftarrow s$  **end if**
  4.     **end for**
  5.     **return**  $\Delta \leftarrow 0$ .
  - else**
  6.      $s = S_i$
  7.     **while** siempre **do**
  8.         **if** se pierde el segmento  $s$  **then return**  $\Delta \leftarrow s$ ; **else**  $s \leftarrow s + 1$  **end if**
  9.     **end while**
  10. **end if**
- 

---

**ALGORITMO LW**

---

<b>PARÁMETROS DE SALIDA:</b>	Número de pérdidas en la LW, $\delta$
------------------------------	---------------------------------------

1. **for**  $s = 2, \delta = 1$  **to**  $w$  **do**
  2.     **if** se pierde el segmento  $s$  **then**  $\delta \leftarrow \delta + 1$  **end if**
  3. **end for**
  4. **return**  $\delta$
-

---

### ALGORITMO CA

---

1.  $\Delta \leftarrow$  ALGORITMO DE PÉRDIDAS {desde el primer segmento enviado en CA y sin Umbral}.
  2. **if** es la primera vez que se ejecuta esta fase **then**  $w \leftarrow \min\left(\max\left(2, \frac{\Delta}{2}\right), \sigma_{ini}\right)$  **end if**
  3.  $\delta \leftarrow$  ALGORITMO LW
  4.  $i \leftarrow \left\lceil -W - \frac{1}{2} + \sqrt{\left(W - \frac{1}{2}\right)^2 + 2\Delta} \right\rceil$ .
  5.  $w \leftarrow w + i, a \leftarrow a + \Delta - 1, m \leftarrow \max(m, a + w), X \leftarrow X + \Delta - 1 + w - \delta, Y \leftarrow Y + \Delta - 1 + w$ .
  6. **if**  $a > r$  **then**  $n_{TO} = 0$  **end if**
  7. **if**  $w - \delta \geq 3$  AND  $a > r$  **then**  $A \leftarrow A + i + 1, \text{FASE} \leftarrow \text{FR}$
  8. **else**  $A \leftarrow A + i + \min(\text{RTO}_{\max}, 2^{n_{TO}} \text{RTO}), \text{FASE} \leftarrow \text{TO}$  **end if**
  9. **return**
- 

---

### ALGORITMO TO

---

1.  $n_{TO} \leftarrow n_{TO} + 1$
  2. **if** es el primer TO para el mismo SN **then**  $r \leftarrow m, \sigma \leftarrow \max\left(2, \lfloor \frac{w}{2} \rfloor\right), w \leftarrow 1$  **end if**
  3.  $\Delta \leftarrow$  ALGORITMO DE PÉRDIDAS {para la retransmisión}.
  4. **if**  $\Delta > 0$  **then**  $A = A + \min(\text{RTO}_{\max}, 2^{n_{TO}} \text{RTO}), Y \leftarrow Y + 1, \text{FASE} \leftarrow \text{TO}$
  5. **else**  $\text{FASE} \leftarrow \text{SS}$  **end if**
  9. **return**
- 

---

### ALGORITMO FR

---

1. **for**  $f = 0$  to  $\delta - 2$  **do**
  2.  $N[f] \leftarrow \max\left(0, \max\left(2, \lfloor \frac{w}{2} \rfloor\right) - \delta + f\right)$
  3.  $\Delta \leftarrow$  ALGORITMO DE PÉRDIDAS {desde la retransmisión y para todos los  $N[f]$  segmentos}.
  4. **if**  $\Delta > 0$  **then**  $A = A + \text{RTO}, \text{FASE} \leftarrow \text{TO},$  **return end if**
  5. **end for**
  6.  $\Delta \leftarrow$  ALGORITMO DE PÉRDIDAS {para la última retransmisión}.
  7. **if**  $\Delta > 0$  **then**  $A \leftarrow A + \text{RTO}, \text{FASE} \leftarrow \text{TO},$  **return end if**
  8.  $r \leftarrow m, w \leftarrow \sigma \leftarrow \max\left(2, \lfloor \frac{w}{2} \rfloor\right), m \leftarrow a = m + \sum_{f=0}^{\delta-2} N[f], X \leftarrow X + \sum_{f=0}^{\delta-2} N[f],$   
 $Y \leftarrow Y + \sum_{f=0}^{\delta-2} N[f], A \leftarrow A + \delta, \text{FASE} \leftarrow \text{CA}.$
  9. **return**
-

---

**ALGORITMO SS**

---

1.  $\Delta \leftarrow$  ALGORITMO DE PÉRDIDAS {desde el segundo segmento enviado en SS y hasta  $\sigma$ }.
  2. **if**  $\Delta > 0$  **then**
  3.      $w \leftarrow \Delta, a \leftarrow a + \Delta - 1, m \leftarrow \text{máx}(m, a + w), X = X + 2\Delta - 1 - \delta, Y = Y + 2\Delta - 1, i = \lfloor \log_2 \Delta \rfloor$ .
  4.      $\delta \leftarrow$  ALGORITMO LW
  5.     **if**  $a > r$  **then**  $n_{\text{TO}} = 0$  **end if**
  6.     **if**  $w - \delta \geq 3$  **AND**  $a > r$  **then**  $A \leftarrow A + i + 2, \text{FASE} \leftarrow \text{FR}$
  7.     **else**  $A \leftarrow A + i + 1 + \text{mín}(\text{RTO}_{\text{max}}, 2^{n_{\text{TO}}}\text{RTO}), \text{FASE} \leftarrow \text{TO}$  **end if**
  8.     **else**  $i = \lfloor \log_2 \sigma \rfloor, w = \sigma, a = a + \sigma, m = \text{máx}(m, a), X = X + \sigma, Y = Y + \sigma,$   
       $A = A + i + 1, \text{FASE} \leftarrow \text{CA}$
  9. **return**
- 

### 3.5. Validación del algoritmo.

La validación del algoritmo se realiza a través de su comparación contra un conjunto de pruebas de simulación en ns-2, de forma semejante a como se llevó a cabo en la sección IV - C en [37] y se reproduce en el CASO 1 de la sección 2.4.5 de esta tesis. En la figura 3.5 se presenta un esquema de la topología empleada para la simulación, que consiste en una red simple, con sólo un enlace “cuello de botella”<sup>17</sup>, configurado como sigue:

- Tasa de transmisión del enlace de 100 Mbps.
- Cola tipo *Drop-Tail*, con capacidad suficiente como para no provocar pérdidas de paquetes por congestión.
- Se introduce un módulo de pérdidas, que fuerza el estado de pérdida de los paquetes que lo atraviesan según un modelo tipo Bernoulli o Gilbert simplificado.
- Retardo fijo.

El retardo en los enlaces de acceso de la fuente y del destino TCP se calcula para que, sumados con el retardo en el “cuello de botella”, den un RTT de 75 ms.

Se evalúan la tasa de transmisión efectiva y la tasa de transmisión en estado estable del flujo TCP NewReno de larga duración que atraviesa este enlace. Ambas tasas se calculan mediante el cociente de una cantidad de segmentos transmitidos en un intervalo de tiempo sobre dicho

---

<sup>17</sup>Es necesario precisar que en ns-2, la congestión se presenta en los enlaces y no en los nodos. Es decir, cada enlace tiene asociada una cola, cuya saturación es la que provoca la pérdida de los paquetes.

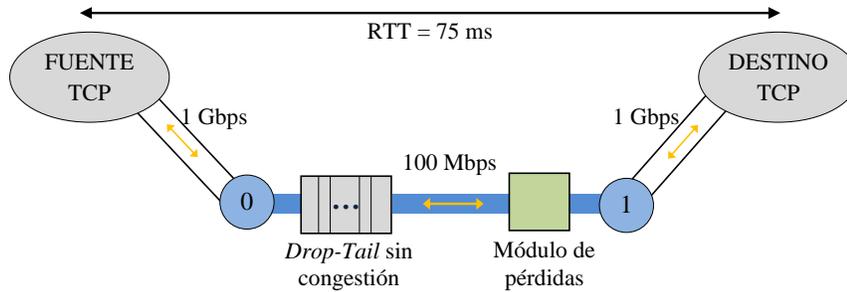


Figura 3.5: Topología usada para la validación del algoritmo.

intervalo. Para la tasa de transmisión efectiva, esa cantidad se refiere al número de segmentos recibidos exitosamente por el receptor TCP. Para la tasa de transmisión, el conteo es del total de segmentos enviados por el transmisor TCP, sin importar si llegaron a su destino o se perdieron. Para ambas cantidades se usan dos variables preestablecidas en el simulador ns-2 y actualizadas en el transmisor TCP: *nackpack\_* y *ndatapack\_*, respectivamente [55]. La variable *nackpack\_* cuenta el número de ACKs (RAs o DAs) recibidos en el transmisor TCP. Por otro lado, la variable *ndatapack\_* cuenta el número de segmentos enviados por el transmisor TCP. Como se mencionó en la discusión sobre la lectura de los parámetros a partir de la simulación, presentada en la sección 2.4.5, en un caso como éste, en donde no hay más flujos atravesando el enlace “cuello de botella”, estas cantidades equivalen a las que arrojan los monitores de flujos y de la cola en el enlace.

Se ejecutaron las pruebas para distintos tamaños del intervalo de observación, a partir de  $1e5$  s y hasta  $1e7$  s, encontrándose una diferencia poco significativa. Los resultados reportados corresponden a intervalos de  $1e7$  s.

La tasa de transmisión efectiva,  $THR_E$ , que resulta de aplicar el algoritmo se compara tanto con la  $THR_E$  medida directamente de la simulación ns-2 como con la  $THR_E$  calculada usando el modelo en [37], de los autores Parvez *et al.*, por ser el modelo analítico más reciente y completo del  $THR_E$  en estado estable de una conexión TCP NewReno de larga duración. La tasa de transmisión, THR, que resulta de aplicar el algoritmo se compara únicamente con la THR medida directamente de la simulación ns-2, pues esta medida de desempeño no aparece en los modelos para el THR de NewReno actuales. Para cada punto sobre las curvas de ambas aproximaciones se calcula el error como está definido en [37], es decir, como  $|\text{simulación} - \text{modelo}| / \text{simulación}$ .

Como se menciona en la sección 2.4.5, los parámetros del modelo en [37] se obtienen a partir de las siguientes mediciones de la simulación: tasa promedio de LEs, tasa promedio de segmentos perdidos y RTT promedio. La tasa promedio de LEs se calcula como el cociente de las variables  $E$  sobre  $ndatapack_.$ ,  $p = E/ndatapack_.$  La variable  $E$  está definida en la sección 2.4.5 como el contador de LEs en el flujo TCP NewReno de estudio durante el intervalo de observación. Específicamente, es una variable definida en el módulo TCP NewReno de ns-2 que se actualiza en cada inicio de una FR o en cada TO, incrementándose en uno. Por otro lado, para la tasa promedio de segmentos perdidos se usa la variable  $q\_red$ , que es la tasa de segmentos perdidos reportada por el monitor de flujos de ns-2. El RTT es un parámetro de la simulación, según el cual se calculan los retardos en los enlaces de la red; sin embargo, para corroborar, se calcula su promedio a lo largo de la simulación ns-2. La tasa promedio de segmentos perdidos dentro de cada  $LE(q)$ , que se obtiene mediante (2.67), (2.66) y (2.57) usando  $q' = q\_red$ . La  $E[RTO]$  se aproxima como  $3E[RTT]$ .

Respecto de los parámetros de entrada del algoritmo, el RTT promedio,  $R_.$ , es el mismo que el usado para el modelo en [37] y corresponde al RTT impuesto en el escenario de simulación. A lo largo de la simulación se calcula el promedio de la RTO, y su división entre  $R_.$  se toma como el parámetro RTO. Como  $ssthresh$  inicial, se considera un valor arbitrariamente alto:  $\sigma_{ini} = 500$  y como duración típica de una conexión TCP NewReno de larga duración se usa  $T = 1e6$  s. El tamaño máximo de segmento es  $SMSS = 960$  y  $RTO_{ini} = 3s$ .

### 3.5.1. Módulo de pérdidas con distribución Bernoulli.

El primer caso que se somete a validación usa un modelo de pérdidas con distribución tipo Bernoulli, cuya tasa promedio de paquetes perdidos,  $p$ , se varía en un intervalo del 0.05 % al 10 %. En la gráfica de la figura 3.6 se muestran las curvas de la  $THR_E$ , en Mbps, para este rango de  $p$ . Aunque es notorio que ambas propuestas para la evaluación del  $THR_E$  son capaces de seguir de cerca al  $THR_E$  que se mide de la simulación, el 1.4 % de error promedio correspondiente al algoritmo es significativamente menor que el 7 % correspondiente al modelo reportado en [37].

### 3.5.2. Módulo de pérdidas con distribución Gilbert Simplificado.

Para la validación del algoritmo se emplea el modelo simplificado de Gilbert, con  $p_{e0} = 0$ ,  $p_{e1} = 1$  y los parámetros  $p_{01}$  y  $p_{10}$  variando en un rango dentro de los valores calculados de

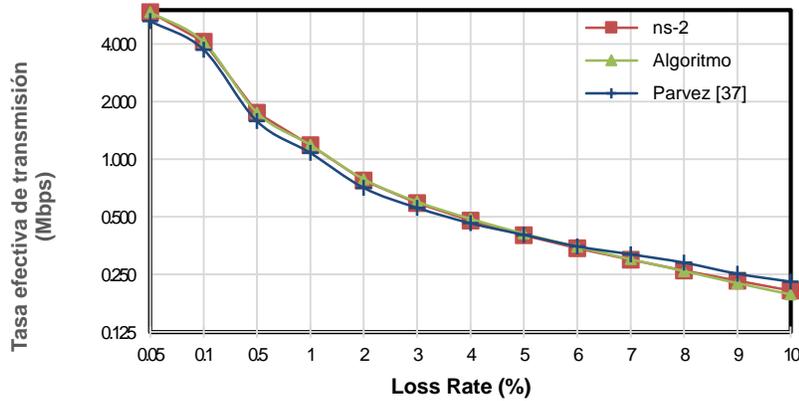


Figura 3.6: Tasa de transmisión efectiva,  $THR_E$ , en Mbps de una conexión TCP NewReno de larga duración con un modelo de pérdidas en la red tipo Bernoulli.

mediciones en [74] y [66], y que han sido usados en trabajos como [67] y [68].  $p_{01}$  se varía en un rango desde 0.005 hasta 0.08 y  $p_{10}$  toma valores entre 0.1 y 0.9; por lo que la longitud promedio de las ráfagas de paquetes perdidos,  $B_L$ , corresponde con valores desde 1.1 hasta 10. En la figura 3.7 se muestran las curvas de la  $THR_E$ , en Mbps, para este rango de  $p_{10}$  y para tres valores de  $p_{01}$ : 0.04, 0.06 y 0.08. En este caso, la tasa promedio de pérdidas es 0.14. El error promedio del algoritmo propuesto resulta en 8.9 %, mientras que el error promedio del modelo de Parvez en [37] asciende hasta el 1900 %. La razón detrás de tan significativo incremento en el error asociado con el modelo en [37] parece estar relacionada con  $B_L$ , pues para  $B_L \leq 1,3$ , el error cae en el rango reportado en [37]. Sin embargo, conforme el valor de  $B_L$  aumenta, es decir, conforme más severa es la naturaleza de ráfaga de las pérdidas en la red, el modelo en [37] tiende a sobrestimar el  $THR_E$ . Una posible explicación de esta tendencia en los resultados del modelo en [37] es que éste se basa en una simplificación optimista de los mecanismos de NewReno de acuerdo con lo siguiente:

- El SS siempre termina en una CA, es decir, no hay posibilidad de pérdida en esta fase.
- La condición para activar la fase FR es incompleta: siempre que existe un TD se transita a la FR, sin considerar a la variable *recover*.
- Se excluye del modelo el efecto del mecanismo de *back-off*, es decir, no se introduce una ponderación creciente de forma exponencial de la RTO en TOs sucesivos para el mismo LE.

De lo anterior, es claro que una representación más fiel de los mecanismos del protocolo en la herramienta de evaluación se traduce en una mayor precisión de sus resultados. Adicionalmente, de la figura 3.7 es posible notar que el  $THR_E$  predicho por el algoritmo subestima

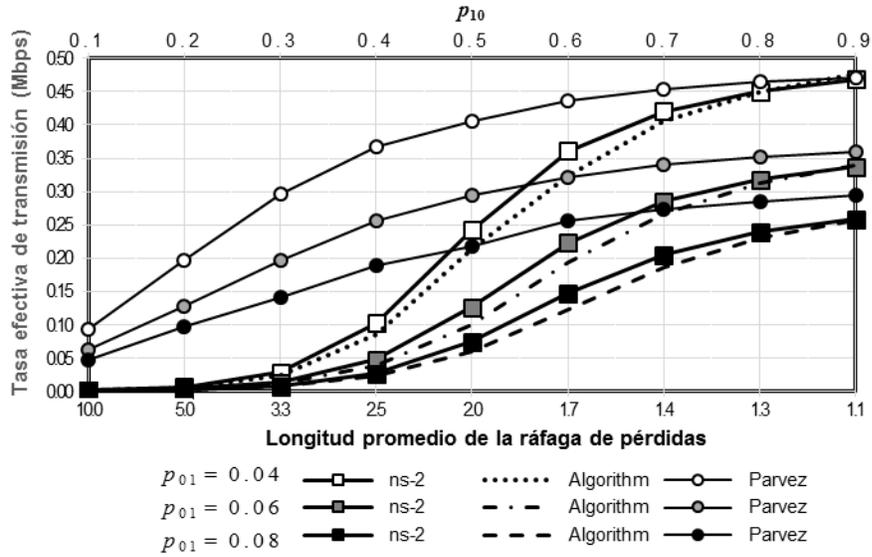


Figura 3.7: Tasa de transmisión efectiva,  $THR_E$ , en Mbps de una conexión TCP NewReno de larga duración con un modelo de pérdidas en la red tipo Gilbert Simplificado. Caso A.

ligeramente al  $THR_E$  medido de la simulación. Lo anterior puede explicarse en función de la simplificación del mecanismo FR de NewReno que se realiza para el diseño del algoritmo y que se hereda del modelo en [37]. Esta simplificación se enuncia como que cualquier LE dentro de la FR resulta en un TO e implica que los segmentos enviados durante la FR incompleta se desprecian, lo que se traduce en una subestimación de  $X$  y de  $Y$ . Además, implícitamente, esta suposición resulta en una mayor cantidad de TOs que la real, pues en principio sólo la pérdida de una retransmisión en una FR puede llevar a un TO, mientras que la pérdida de cualesquiera otros segmentos en la FR se puede recuperar mediante una FR siguiente (FRs sucesivos). Ya que el mecanismo de control de congestión relacionado con un TO reduce la tasa de transmisión más drásticamente que el asociado con un TD, el resultado es una subestimación del  $THR_E$ .

En la figura 3.8 se presentan las curvas de la  $THR_E$ , en Mbps, para el mismo rango de  $p_{10}$  y para cuatro valores más de  $p_{01}$ : 0.02, 0.009, 0.007 y 0.005. En este caso, la tasa promedio de pérdidas es 0.03. El error promedio del algoritmo resulta en 16.2 %, mientras que el error promedio del modelo de Parvez en [37] asciende hasta el 677 %. De nuevo, la aproximación que se logra con este algoritmo supera por mucho a la del modelo en [37]. Sin embargo, su error promedio se incrementa en casi el doble respecto de los casos en la figura 3.7. La diferencia entre ambos escenarios radica en el valor de  $p_{01}$ . Un valor menor de  $p_{01}$ , como en las curvas de la figura 3.8, significa que existe una menor probabilidad de que estando en un

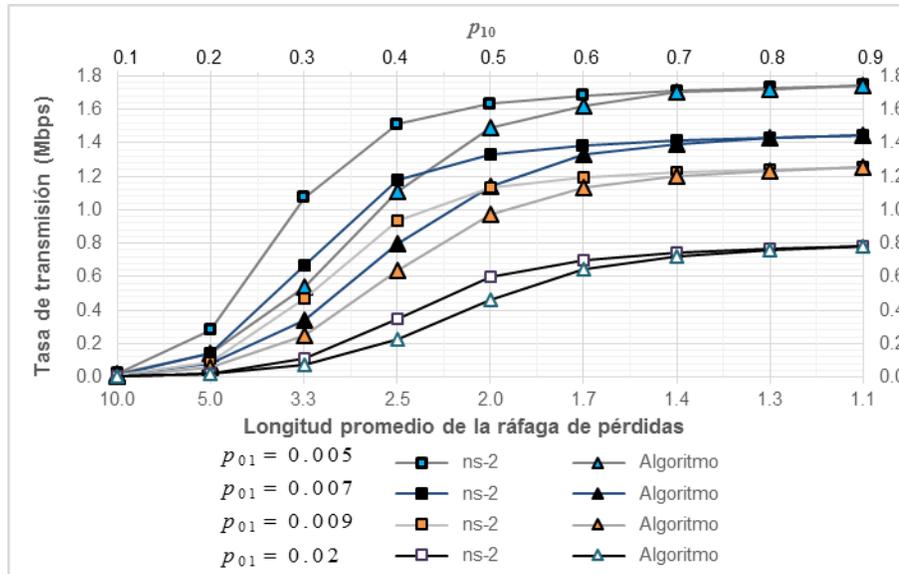


Figura 3.8: Tasa de transmisión efectiva,  $THR_E$ , en Mbps de una conexión TCP NewReno de larga duración con un modelo de pérdidas en la red tipo Gilbert Simplificado. Caso B.

estado libre de pérdidas se produzca una, de forma que la longitud promedio de las secuencias de segmentos sin pérdida es mayor, lo cual también se refleja en un valor menor de la tasa promedio de pérdidas. Es difícil hallar con precisión la razón de esta degradación en el grado de aproximación, sin embargo, es claro que las simplificaciones a las se recurre para su diseño son la razón general de las imprecisiones que resultan del algoritmo.

En la figura 3.9 se presentan las curvas de la THR, en Mbps, para los mismos valores de  $p_{10}$  y  $p_{01}$  que en la figura 3.7. El error promedio del algoritmo resulta en 12.9 %, mayor que el 8.9 % correspondiente al cálculo de la  $THR_E$ . La sub-estimación de la THR puede radicar en que la cantidad de segmentos transmitidos se subestima o en que el tiempo de la conexión se sobrestima. Según las simplificaciones a las que se recurre para el diseño de este algoritmo, ambas cosas suceden. La subestimación de los segmentos principalmente se debe, como se mencionó anteriormente, a la simplificación del mecanismo FR. Por otro lado, también se usa una simplificación del mecanismo de recuperación de pérdidas activado mediante un TO: éste sigue un esquema *Go-Back-N*. Es decir, todos los segmentos a partir del que se detectó como perdido se consideran como perdidos y se retransmiten. Con esta suposición se sobrestima la cantidad de segmentos perdidos, pero también se sobrestima, y en mucho mayor proporción, el tiempo empleado para esta retransmisión, debido a lo siguiente: existe una condición necesaria para que se pueda activar una recuperación de pérdidas mediante la FR:  $a > r$ . Es decir, si existe una pérdida, el transmisor TCP se asegura que ésta no pertenezca

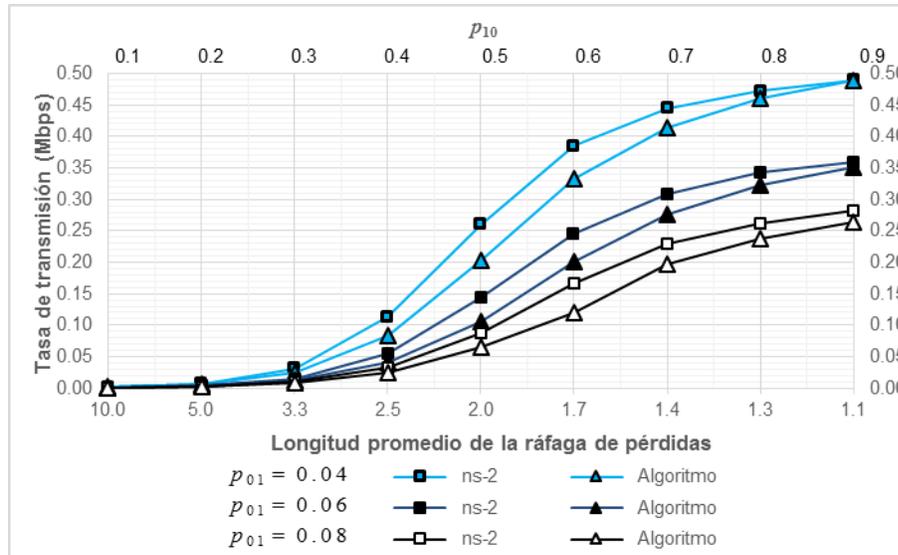


Figura 3.9: Tasa de transmisión, THR, en Mbps de una conexión TCP NewReno de larga duración con un modelo de pérdidas en la red tipo Gilbert Simplificado.

a un LE previo para poder recuperarla con una FR. La suposición de un mecanismo *Go-Back-N* implica que el número de retransmisiones supuestas por el algoritmo es mayor que el real. Si una de éstas se pierde, según el algoritmo, forzosamente se recuperará a partir de un TO. Esto se traduce en una reducción de la  $THR_E$  predicha por el algoritmo.

### 3.6. Conclusiones del algoritmo.

En este capítulo se presenta un algoritmo como herramienta de evaluación del THR (y del  $THR_E$ ) en estado estable de una conexión TCP NewReno de larga duración. El algoritmo usa como parámetros de entrada el RTT promedio, la RTO promedio, el valor inicial del *ssthresh*, la duración típica de una conexión TCP de larga duración y los parámetros del modelo de pérdidas empleado para caracterizar el comportamiento de pérdidas de la red. Sus resultados presentan un mucho mejor grado de aproximación a los de la simulación respecto de los modelos analíticos existentes. En particular se compara con el modelo desarrollado por Parvez *et al.* en [37], por ser el modelo más reciente y completo en esta área. El incremento en la precisión de los resultados arrojados por este algoritmo respecto de los modelos existentes se debe principalmente a dos factores:

- La incorporación de una representación más cercana y detallada de los mecanismos del protocolo, en particular del SS y la CA.

- La posibilidad de usar un modelo de pérdidas adecuado para caracterizar distintos escenarios de operación en la red.

Además, en comparación con la simulación como herramienta de evaluación del THR, con este algoritmo se evitan la complejidad y el consumo de tiempo y recursos asociados con ella.

El algoritmo realiza un seguimiento de la dinámica presente en la transmisión TCP, la cual está determinada por el comportamiento de pérdidas de la red. Específicamente, el algoritmo avanza a través de los distintos mecanismos (llamados fases) del protocolo. La transición entre fases es guiada según la secuencia de pérdidas generada por el modelo de pérdidas. El algoritmo usa un conjunto de variables globales, las cuales se van actualizando con el paso por cada fase. Entre éstas se encuentra el contador de segmentos enviados, el contador de segmentos enviados exitosamente y la duración de la conexión. Una vez que la variable de la duración alcanza un valor representativo de una conexión de larga duración, el algoritmo se detiene y las tasas de transmisión, THR y  $THR_E$ , se calculan.

El algoritmo se valida a través de su comparación con un conjunto de pruebas de simulación ns-2. Además, sus resultados se comparan con los correspondientes del modelo en [37]. Se consideran dos escenarios: el primero con un modelo de pérdidas tipo Bernoulli y el segundo con un modelo simplificado de Gilbert. En todos los casos, el error promedio presentado por el algoritmo es significativamente menor que el resultante del modelo de Parvez. Sin embargo, a pesar de esta mejora notable respecto de las aproximaciones por modelado analítico, existen escenarios de prueba en los que el error promedio asciende por encima del 15 %. La explicación de este fenómeno se basa en la serie de simplificaciones de los mecanismos de recuperación de pérdidas a las que se recurre en el diseño de este algoritmo. Este conjunto de suposiciones se hereda de los trabajos previos, principalmente de [37] y [32], con el fin de hacer manejable el nivel de complejidad del análisis.

Entre las limitantes más severas que se pueden señalar de este trabajo se tienen las siguientes:

- Es válido sólo para flujos TCP NewReno de larga duración, de forma que quedan excluidos del análisis los flujos cortos que constituyen el mayor porcentaje de la cantidad de flujos transitando en Internet [38] [39] [41] [42] [47].
- En lugar de modelar explícitamente la pérdida de una retransmisión en la FR, se supone que cualquier LE dentro de la FR fuerza un TO.

- Se excluyen del análisis los casos en los que se presentan FRs sucesivas, en los que la pérdida de algunos segmentos transmitidos dentro de una FR se recupera con una siguiente FR.
- Se desprecian los segmentos enviados dentro de una FR incompleta, dejando fuera el análisis de la dinámica de la transmisión en la FR cuando hay pérdidas en ella.
- Al ocurrir un TO, se considera que son retransmitidos todos los segmentos a partir del asociado con el TO y hasta el último segmento transmitido hasta el momento en que se presenta el TO (*Go-Back-N*).

En el siguiente capítulo se presenta el desarrollo de una extensión de este algoritmo con el que se corrigen todas estas limitantes.



## Capítulo 4

# Algoritmo para la evaluación del desempeño de una conexión TCP NewReno.

En este capítulo se desarrolla un algoritmo para la evaluación del valor esperado del THR de una conexión TCP NewReno. Sus parámetros de entrada son el RTT promedio, la RTO promedio, el valor inicial del *ssthresh*, el tamaño del flujo TCP y los parámetros del modelo empleado para caracterizar el comportamiento de pérdidas de la red.

Este algoritmo se desarrolla como una extensión al algoritmo para la evaluación del THR en estado estable de una conexión TCP NewReno de larga duración presentado en el capítulo 3. Con esta extensión se busca eliminar las restricciones del algoritmo referido, a través de: generalizar esta herramienta de evaluación para flujos de cualquier tamaño; suprimir las simplificaciones referentes a la fase FR; e incluir una aproximación más adecuada del mecanismo de recuperación de pérdidas activado por un TO que las presentes en los modelos analíticos existentes.

Este algoritmo realiza un seguimiento de los mecanismos por los que atraviesa una conexión TCP NewReno. Cada mecanismo está representado por una fase. El orden en que se siguen las fases en una conexión está determinado por la secuencia de segmentos erróneos a lo largo de ella, la cual es generada por el modelo de pérdidas. El algoritmo ocupa un conjunto de variables globales, las cuales se van actualizando a su paso por las fases. Entre éstas variables se tienen las correspondientes a la cantidad de segmentos transmitidos y a la duración de la conexión. A diferencia del algoritmo del capítulo 3, que es para flujos TCP de larga duración, este algoritmo comienza por la fase inicial de una conexión ya establecida (el SS inicial) y termina cuando se ha concluido la transmisión confiable del flujo. El  $THR_E$  de la conexión se

calcula como el cociente del total de segmentos transmitidos exitosamente entre la duración de la conexión. El THR se calcula como el cociente del total de segmentos enviados durante la conexión entre su duración. El valor esperado del THR ( $THR_E$ ) de una conexión TCP de cierto tamaño se aproxima mediante el promedio de los THRs ( $THR_{ES}$ ) obtenidos de una muestra adecuada de conexiones de ese tamaño.

El algoritmo se valida a través de su comparación con los resultados de un conjunto de pruebas de simulación en ns-2. El modelo de pérdidas usado tanto en la simulación como en el algoritmo es de tipo Gilbert, obteniendo un error promedio del 2.15 %.

## **4.1. Suposiciones sobre las que se desarrolla el algoritmo.**

### **4.1.1. Suposiciones generales.**

Se conservan las suposiciones generales 3 - 10, 12, 13 y 20 - 21 de la sección 2.2.1, correspondientes al modelo en [32] y retomadas en [48], [49], [50], [37] y en el algoritmo del capítulo 3.

### **4.1.2. Suposiciones que se derivan del modelo en [37].**

Se retoman las suposiciones particulares 3, 5 - 8 de la sección 2.4.2, usadas también para el algoritmo del capítulo 3.

### **4.1.3. Suposiciones particulares que se derivan del algoritmo del capítulo 3.**

Se mantienen las suposiciones 1, 4 - 8 de la sección 3.1.3.

### **4.1.4. Suposiciones particulares usadas para este algoritmo.**

1. La suposición 1 de la sección 2.2.1 se modifica como sigue: se considera una fuente de datos continua pero finita. De hecho, el tamaño del flujo es un parámetro más del modelo.

2. La suposición 2 de la sección 2.2.1 se altera para quedar como: se ignora la fase de establecimiento de la conexión y se modelan todas las demás fases de NewReno “lento pero firme”, incluyendo el SS inicial (que es determinante para los flujos de corta duración, los cuales constituyen la mayor proporción de los flujos en Internet).
3. La suposición 14 de la sección 2.2.1 se cambia de la siguiente manera: la primera ronda ocurre al inicio de la fase SS inicial.
4. La suposición 2 de la sección 2.4.2 cambia a: una ventana de pérdida, LW, comienza con el primer segmento perdido en una fase y tiene longitud  $\max(w, m - a)$ , con  $a$ ,  $m$  y  $w$  actualizadas según la fase (ver definición de las variables en la sección de variables globales del algoritmo).
5. Se elimina la suposición 4 de la sección 2.4.2, que implica una simplificación al mecanismo FR de NewReno en el análisis.
6. La suposición 9 de la sección 2.4.2 se altera de forma que, para el mecanismo de recuperación de pérdidas activado por un TO se supone un esquema *Go-Back-N* pero no a partir del segmento detectado como perdido, sino desde un segmento posterior, de tal manera que se evita que todos los segmentos sean retransmitidos. El tamaño del salto en la retransmisión se calcula para que, en promedio, represente todas las posibles cantidades de segmentos realmente retransmitidos en una conexión (ver explicación de la fase TO del algoritmo).
7. La suposición 2 de la sección 3.1.3 cambia para considerar que una conexión TCP transita entre distintas fases comenzando con la fase SS inicial y terminando con la fase “fin de la conexión” (EC: *End of the Connection*).
8. Para efectos de las pérdidas en una FR, se distingue entre un segmento retransmitido (retransmisión) y un segmento transmitido por primera ocasión (transmisión nueva): una retransmisión perdida provoca forzosamente un TO y una transmisión nueva perdida podrá recuperarse por una FR siguiente o mediante un TO.
9. Se distingue entre una FR que concluye exitosamente la recuperación del último LE, llamada cFR (*complete FR*), y una FR que termina en un TO por causa de alguna retransmisión fallida (iFR: *incomplete FR*).
10. La suposición 9 de la sección 3.1.3 se modifica como: un evento de pérdida, LE, comienza con el primer segmento perdido en una fase y dura un RTT, excepto cuando se trate de una FR. En una cFR, un LE comienza con la primera transmisión nueva perdida

y se prolonga hasta el término de la CA posterior a la cFR. La LW mide  $\max(w, m - a)$ , con  $a$ ,  $m$  y  $w$  actualizadas según dicha fase CA. En una iFR, un LE comienza con la primera retransmisión fallida y se prolonga hasta que termina la iFR. La LW mide  $\max(w, m - a)$ , con  $a$ ,  $m$  y  $w$  actualizadas según la iFR.

## 4.2. Descripción del algoritmo.

Este algoritmo tiene como objetivo la evaluación del THR promedio de una conexión TCP NewReno. Sus parámetros de entrada son el RTT promedio,  $R_*$ ; la duración promedio, en rondas, del temporizador de pérdidas cuando el factor de *back-off* es unitario, RTO; la duración máxima de la RTO,  $RTO_{\max}$ ; el tamaño máximo de segmento en bits, SMSS; el valor inicial del *ssthresh*,  $\sigma_{ini}$ ; el tamaño, en segmentos, del flujo TCP,  $C_*$ ; y los parámetros del modelo de pérdidas empleado para caracterizar el comportamiento de pérdidas de la red.

Si una conexión TCP dura  $A$  rondas y el número de segmentos enviados en total es  $Y$ , entonces la tasa de transmisión de la conexión se calcula como:

$$THR = \frac{Y(SMSS)}{AR_*} \quad (4.1)$$

Si de los  $Y$  segmentos transmitidos, sólo  $X$  no sufrieron pérdida, la tasa de transmisión efectiva está dada por:

$$THR_E = \frac{X(SMSS)}{AR_*} \quad (4.2)$$

Cada fase del protocolo tiene asociada una parte del algoritmo, de forma que éste se mueve a través de las fases según lo determine el modelo que representa el comportamiento de pérdidas de la red. Al paso de cada fase, un conjunto de variables globales se actualiza. Dichas variables se enlistan en las tablas 4.1 y 4.2.

Las variables  $a$ ,  $m$ ,  $X$ ,  $Y$  y  $A$  son acumulativas.<sup>1</sup> Los valores de inicialización de las variables globales son:  $a = m = r = X = Y = 0$ ,<sup>2</sup>  $A = 0$ ,  $n_{TO} = 0$ ,  $\sigma = \sigma_{ini}$  y  $w = 1$ .

<sup>1</sup>En el modelo se considera que cada ACK apunta al último número de secuencia recibido exitosamente por el receptor, en lugar de al siguiente esperado, como suele emplearse en las implementaciones de TCP.

<sup>2</sup>En [61] se especifica que  $r$  se inicializa como  $r = SN$  inicial. El modelo se inicializa  $r = 0$ . El cambio en el modelo para incluir esta modificación es directo, sin embargo, si se considera  $iwnd = 1$ , este cambio en  $r$  inicial no afecta.

Tabla 4.1 Variables globales principales.

Variable	Descripción
$a$	SN recibido exitosamente máximo. <sup>1</sup>
$A$	Duración, en rondas, de la conexión.
$d$	Cantidad de DAs al salir de una fase.
$\Delta$	Primer segmento perdido en la fase. $\Delta \geq 1$ .
$\delta$	Cantidad de segmentos perdidos en la fase.
$m$	SN transmitido máximo.
$n_{TO}$	Es el exponente de la base 2 con que se pondera a la RTO.
$r$	Variable <i>recover</i> de TCP NewReno.
$s$	SN del último segmento transmitido.
$\sigma$	<i>ssthresh</i> .
$w$	Tamaño de la <i>cwnd</i> .
$X$	Cantidad de segmentos enviados exitosamente
$Y$	Es la cantidad de segmentos enviados en total.

Tabla 4.2 Variables globales auxiliares.

Variable	Descripción
$a_s$	Variable auxiliar en el cálculo de $a$ en un TO.
$A_{CA\_aux}$	Variable auxiliar en el cálculo de $A$ en la CA, cuyo valor se establece en la fase que precede a la CA.
$\beta$	$\beta \in [0, w - 1]$ es la posición del segmento que activa la transición a la siguiente fase dentro de la <i>cwnd</i> en la ronda $i$ de cada fase.
$i$	Índice de la ronda en la que se presenta el segmento que provoca la transición a otra fase. $i \geq 0$ .
$f$	Índice de las rondas dentro de cada fase. Al inicio de cada fase se inicializa como $f = 0$ , y termina con un valor $f = i$ .
$f_{FR}$	Equivale al número de rondas que duró la última fase FR.
$N[]$	Arreglo que contiene el número de segmentos transmitidos en cada ronda de la última FR. Es de tamaño $f_{FR}$ más uno.

En la figura 4.1 se presenta el diagrama de transiciones entre fases que define la dinámica de TCP NewReno a lo largo de una conexión. Junto a cada transición aparece el conjunto de condiciones que se requiere para que la transición se produzca. El THR ( $THR_E$ ) de una conexión particular depende de la secuencia de fases por las que transcurrió. La promediación de una muestra adecuada de conexiones TCP NewReno resultará en el THR ( $THR_E$ ) promedio de una conexión TCP.

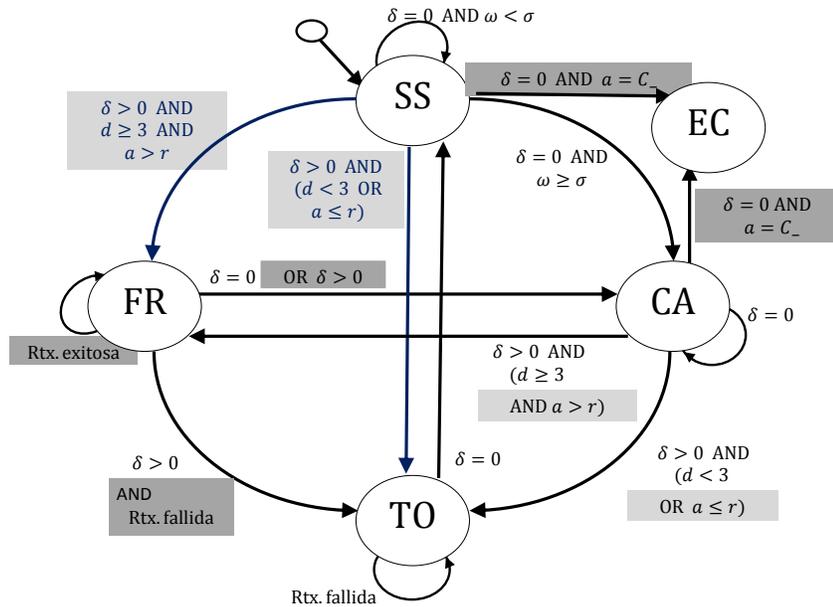


Figura 4.1: Fases de una conexión TCP NewReno y las transiciones entre ellas.

A continuación se presenta, para todas y cada una de las fases, una explicación breve de su funcionamiento, del conjunto completo de condiciones que provocan la transición hacia y desde ellas, y de cómo se ve reflejado el paso por la fase en el valor de las variables globales del algoritmo.

#### 4.2.1. Fase SS.

Esta fase se ocupa tanto al inicio de una conexión ( $SS_{ini}$ : SS inicial), como después de un TO. En el  $SS_{ini}$  la transmisión se inicia a partir de un tamaño de  $cwnd$  determinado por la  $iwnd$ , cuyo valor se considera de un SMSS.<sup>3</sup> Por otro lado, cuando un TO se presenta, se retransmite el segmento que provocó el TO. Si la retransmisión es exitosa, entonces ésta se considera como la primera transmisión dentro del SS que prosigue al TO (ver explicación de la *Aproximación TO - SS*). Un SS posterior a un TO comienza con un tamaño de ventana igual al valor de la  $lwnd$ , que según [15] es de un SMSS.

Durante el SS,  $w$  se incrementa en un SMSS por cada RA a partir de un SMSS (ya que

<sup>3</sup>En el estándar más reciente para TCP Reno, RFC 5681 [15], se establece que el tamaño de la  $iwnd$  debe ser 2, 3 ó 4 SMSS según el tamaño del SMSS. En el algoritmo no se incluye esta modificación del estándar. Sin embargo, ajustarlo no supone gran complicación.

$iwnd = lwnd = 1$  SMSS) y mientras que  $w < \sigma$ .<sup>4</sup> Si no existe ninguna pérdida en el SS, éste termina con una transición SS - CA. Por el contrario, y en contraste con los modelos actuales para el THR de TCP NewReno, la ejecución del modelo de pérdidas estando en el SS puede dar como resultado la pérdida del  $\Delta$ -ésimo segmento, dando origen a un LE. En este caso, el SS terminará con una transición SS - FR o SS - TO.

Con la recepción del RA del  $n$ -ésimo segmento enviado en esta fase, el valor de  $w$  se actualiza a  $n + 1$ . De forma que, si la transmisión de todos los segmentos dentro de la  $cwnd$  es exitosa, entonces  $w$  se duplica al inicio de la siguiente ronda. Es decir,  $w$  crece exponencialmente, con base 2, en cada ronda. Por lo tanto, la ronda en la que se transmite el  $n$ -ésimo segmento se puede calcular como:

$$f = \lfloor \log_2 n \rfloor \quad (4.3)$$

Esto queda claro si se examina la figura 4.2. En ella se muestra un conjunto de casos en los que la transmisión en el SS termina con un LE. En cada caso se puede observar el crecimiento de  $w$  a partir de la  $iwnd$ . El SS se inicia en la ronda  $f = 0$  con la transmisión de los  $w = 1$  segmentos, comenzando con el SN = 1. Al final de la ronda  $f = 0$  se recibe el RA 1, el cual provoca que la  $cwnd$  crezca como  $w = w + 1 = 2$  y se desplace para comenzar con el segmento 2. La ronda  $f = 1$ , comienza con la transmisión de los segmentos 2 y 3. Hacia el final de esta ronda, se recibe el RA 2 por el que, de nuevo, la  $cwnd$  crece como  $w = w + 1 = 3$  y se desplace para comenzar con el segmento 3, abarcando hasta el segmento 5. Como se considera transmisión en ráfaga (*back-to-back*), inmediatamente después del RA 2 se recibe el RA 3, con lo que la  $cwnd$  crece como  $w = w + 1 = 4$  y se desplace para comenzar con el segmento 4, abarcando hasta el segmento 7. Al inicio de la ronda  $f = 2$ , se transmiten los segmentos [4 - 7]. Hacia el final de esta ronda llegan en ráfaga los RAs 4 al 7, con lo que la  $cwnd$  se aumenta de uno en uno hasta  $w = 8$  y queda desplazada para abarcar los segmentos [8 - 15], los cuales se transmiten en la siguiente ronda  $f = 3$ . Así, generalizando es claro que, mientras no se presenta ninguna pérdida, la  $cwnd$  al inicio de cada ronda  $f$  es del doble del tamaño que en la ronda anterior. Además, el límite izquierdo de la  $cwnd$  coincide con el  $w$ -ésimo segmento enviado en el SS.

En el algoritmo se usan dos variables auxiliares descritas en la tabla 4.3.

---

<sup>4</sup>En el RFC 5681 [15] se menciona que este incremento es el máximo que se puede tener. Para efectos del algoritmo, sin embargo, no existen las condiciones que pudieran generar incrementos menores.

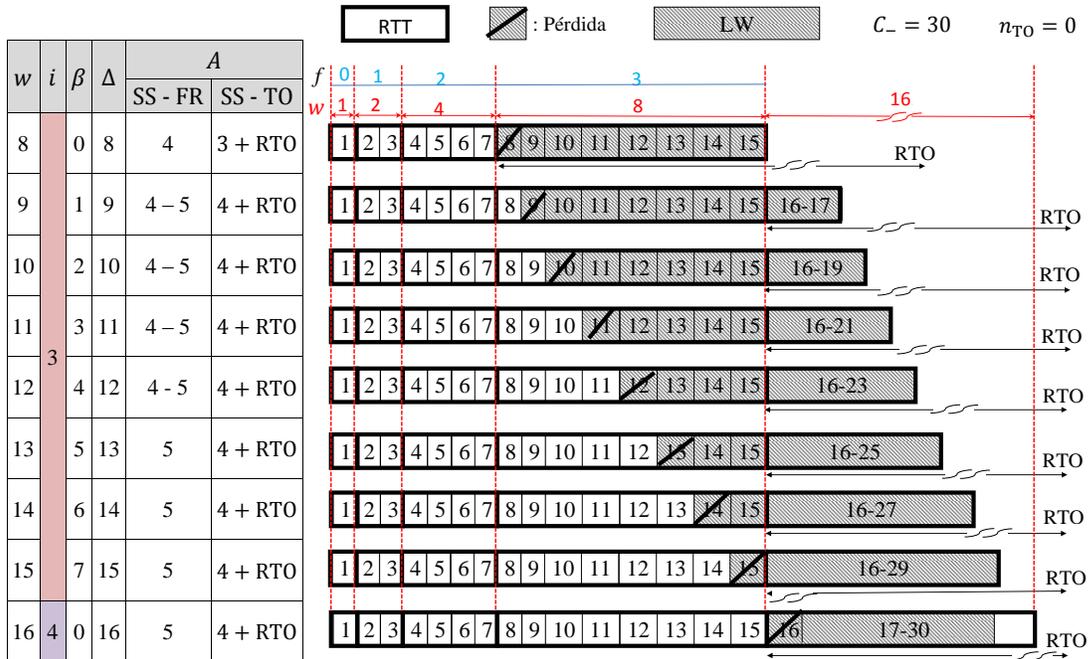


Figura 4.2: Casos de un SS que termina con un LE.

Tabla 4.3 Variables locales de la fase SS.

Variable	Descripción
$a_{SS}$	Representa la cantidad de segmentos enviados exitosamente en la fase antes de la primera pérdida
$m_{SS}$	Número de segmentos enviados en total en el paso por el SS.

### Transición SS - FR.

Si un LE se presenta en el SS, forzará la transición hacia una fase TO o FR. Considérense como ejemplo la figura 4.2. En cada caso presentado en la figura, el LE comienza con la pérdida del  $\Delta$ -ésimo segmento enviado en el SS.  $\Delta \geq 1$  si se trata del  $SS_{ini}$ ; mientras que en cualquier otro SS,  $\Delta > 1$ . Es decir, en el  $SS_{ini}$ , cualquier segmento transmitido es sujeto de pérdida. Sin embargo, en un SS que prosigue a un TO, el primer segmento corresponde con la retransmisión exitosa del segmento asociado con el TO; es decir, el primer segmento transmitido en un SS que sigue de un TO no es sujeto de pérdida (ver la explicación de la *Aproximación TO - SS*). Debido al crecimiento exponencial (con base 2) de  $w$  en cada ronda, la ronda en la que ocurre el LE,  $f = i$ , puede calcularse como:

$$i = \lfloor \log_2 \Delta \rfloor \tag{4.4}$$

El  $\Delta$ -ésimo segmento está ubicado en la posición  $\beta \geq 0$  de la *cwnd* transmitida al inicio de la ronda  $f = i$ , por lo que:

$$\beta = \Delta - 2^i \quad (4.5)$$

$w = \Delta$  en el momento en que el LE se presenta. Así, la expresión para  $w$  en términos de  $i$  y  $\beta$  es:

$$w = 2^i + \beta \quad (4.6)$$

La cantidad de segmentos enviados exitosamente en el SS antes del LE está dada por:

$$a_{SS} = \Delta - 1 = 2^i + \beta - 1 \quad (4.7)$$

En el momento en que el LE se presenta, la *cwnd*, de tamaño  $w$ , queda desplazada de tal forma que comienza con el primer segmento perdido. La LW (que en la figura aparece remarcada) abarca desde el  $\Delta$ -ésimo segmento y termina con el último segmento enviado en esa fase,  $s$ . Si el tamaño del flujo,  $C_-$ , es mayor que la cantidad de segmentos que se alcanzan a transmitir en el SS (i.e.  $C_- > s$ ), entonces LW mide  $w$ . Por ejemplo, en la figura 4.2 el tamaño del flujo es  $C_- = 30$ , de forma que en todos los casos, excepto en el último, el SS termina antes de transmitir el flujo completo y la LW mide  $w$ . En consecuencia, en estos casos, el total de segmentos enviados durante el SS,  $m_{SS}$ , queda como  $a_{SS} + w$ . Sin embargo, en el último caso, la transmisión del flujo termina antes de que la última *cwnd* se cierre, por lo que el tamaño de la LW es menor que  $w$  y el  $m_{SS}$  abarca todos los segmentos desde el inicio de la transmisión en el SS y hasta el último segmento del flujo. Por lo tanto, en cualquier caso, la cantidad de segmentos transmitidos en total en el SS está dada por:

$$m_{SS} = \min(C_- - a, a_{SS} + w) = \min(C_- - a, 2\Delta - 1) = \min(C_- - a, 2^{i+1} + 2\beta - 1) \quad (4.8)$$

con el valor de  $a$  al inicio de la fase, puesto que la transmisión en el SS comienza con el SN posterior al último segmento reconocido antes de entrar a la fase (ver explicación de la fase TO). La variable  $s$  representa el SN del último segmento transmitido. Al salir del SS,  $s$  apunta al último segmento de la LW, es decir,  $s$  se actualiza en el SS como:

$$s = a + m_{SS} \quad (4.9)$$

con el valor de  $a$  al inicio de la fase SS. La variable  $\delta$  almacena la cantidad de pérdidas en

la LW. En [15] se especifica que cuando un receptor TCP recibe un segmento fuera de orden debe emitir inmediatamente un DA, por lo que cada segmento de la LW que no se pierde generará un DA como respuesta del receptor. Por ello, el número de DAs que resulta al final del SS es:

$$d = m_{SS} - a_{SS} - \delta \quad (4.10)$$

La variable  $X$  se actualiza con la cantidad de segmentos enviados exitosamente, que es la suma de los  $a_{SS}$  segmentos antes del LE mas los  $d$  segmentos no perdidos de la LW, es decir:

$$X = X + a_{SS} + d \quad (4.11)$$

Para la variable  $Y$  se toman en cuenta todos los segmentos enviados durante el SS, de forma que:

$$Y = Y + m_{SS} \quad (4.12)$$

La variable  $a$ , que representa el máximo SN reconocido, termina apuntando al segmento anterior a la primera pérdida. Como  $a$  es acumulativa, al finalizar el SS se actualiza como:

$$a = a + a_{SS} \quad (4.13)$$

Según [14], una pérdida se detecta en TCP NewReno por cualquiera de dos eventos: TD o TO. Si se cumple que el número de DAs al finalizar la fase es  $d \geq 3$ , entonces un TD ocurre. Sin embargo, para que la recuperación de las pérdidas del LE sea efectuada a través de una FR, también es necesario que todos los LEs previos ya hayan sido recuperados completamente. En cada LE, la variable *recover*,  $r$ , se actualiza con el SN máximo que se ha transmitido hasta ese momento,  $m$ . Por lo tanto, la condición que asegura que ya se han recuperado por completo los LEs anteriores es que al presentarse el LE, el valor del máximo SN reconocido,  $a$ , cumpla con que  $a > r$ .<sup>5</sup>

La condición  $d \geq 3$  implica que la *cwnd* debe consistir mínimo de cuatro segmentos, es decir, que  $w \geq 4$ , lo que ocurre a partir de la ronda  $f = 2$ . Por otro lado, la mínima ubicación del

---

<sup>5</sup>En realidad, si  $a = r$ , ya se recuperaron todos los errores de las LWs anteriores (variante *Less careful*) [25], pero en la variante *Careful* del estándar RCF 2582 [25], incorporada como parte del nuevo estándar [61], se indica que la recuperación total de los LEs previos ocurre cuando se reconoce al menos un byte más que *recover*. Esta es la condición que se usa en el algoritmo. Adicionalmente, en el algoritmo se considera que la variable  $a$  indica el máximo SN recibido exitosamente por el receptor TCP, por lo que esta condición se traduce en  $a > r$ ; pero en el estándar [61]  $a$  apunta al siguiente SN esperado, por lo que si se usara así la definición de  $a$ , la condición sería  $a - 1 > r$ .

$\Delta$ -ésimo segmento que satisface la condición  $a > r$  corresponde a la ronda:

$$i_{SSr+} = \begin{cases} \lfloor \log_2(r - a + 2) \rfloor & ; r - a + 2 > 0 \\ 0 & ; r - a + 2 \leq 0 \end{cases} \quad (4.14)$$

en la posición dentro de la *cwnd* enviada al inicio de esa ronda:

$$\beta_{SSr+} = \begin{cases} r - a + 2 - 2^{i_{SSr+}} & ; r - a + 2 > 0 \\ 0 & ; r - a + 2 \leq 0 \end{cases} \quad (4.15)$$

con el valor de  $a$  al comienzo de la fase. De forma que los valores de  $i$  y  $\beta$  que corresponden al primer segmento que al perderse en un SS puede activar una FR son, respectivamente:

$$i_{\min SSFR} = \text{máx}(2, i_{SSr+}) \quad (4.16)$$

$$\beta_{\min SSFR} = \begin{cases} \beta_{SSr+} & ; i = i_{SSr+} \\ 0 & ; i > i_{SSr+} \end{cases} \quad (4.17)$$

Por otro lado, la máxima cantidad de segmentos que pueden enviarse sin pérdida antes de que el SS transite hacia la CA es igual a  $\sigma$  (ver la explicación de la transición SS - CA). Así que un LE que de lugar a una FR a lo más puede presentarse con la pérdida del  $\sigma$ -ésimo segmento transmitido en el SS (i.e.  $\Delta \leq \sigma$ ), siempre y cuando, esté a su vez alejado mínimo tres segmentos del final del flujo. Por lo tanto, los valores de  $i$  y  $\beta$  que corresponden al último segmento dentro del SS que al perderse puede activar una FR son, respectivamente:

$$i_{\max SSFR} = \lfloor \log_2 \text{mín}(\sigma, \text{máx}(0, C_- - a - 3)) \rfloor \quad (4.18)$$

$$\beta_{\max SSFR} = \text{mín}(\sigma, \text{máx}(0, C_- - a - 3)) - 2^{i_{\max SSFR}} \quad (4.19)$$

con el valor de  $a$  antes de actualizarse.

*Aproximación SS-FR.* La duración del SS depende de la fase a la que transita al presentarse el LE y de la posición de las pérdidas en la LW. En los casos de la figura 4.2, el inicio del LE está marcado con la pérdida del  $\Delta$ -ésimo segmento en la ronda  $f = i$ , pero los demás errores dentro de la LW no están marcados, de forma que se pueden pensar todas las posibles combinaciones de errores en ella. Para que el LE se recupere mediante una FR es necesario que se produzca un TD. Cuando  $\beta = 0$ , la LW se transmite completa en la ronda  $i$ , de forma

que forzosamente el TD ocurrirá al final de esta ronda y la duración de la fase es  $i + 1$ . Sin embargo, cuando  $\beta > 0$ , la LW se extiende hasta la ronda  $i + 1$ . Dependiendo de la posición de los errores en la LW será el momento en que se produzca el TD. Por ejemplo, cuando  $\beta = 1$ , si los errores ocurren de forma que el tercer segmento no perdido dentro de la LW es a lo más el 15, la duración de la fase es  $i + 1$ , pero si el TD se completa con el DA del segmento 16 o del 17, entonces la duración de la fase es  $i + 2$ . Por otro lado, si  $\beta \geq 5$  en esta figura, la duración de la fase es  $i + 2$  pues la cantidad de segmentos de la LW dentro de la ronda  $i$  no es suficiente para generar el TD aunque ningún segmento se pierda. El algoritmo usa como información la posición del  $\Delta$ -ésimo segmento, pero no la posición de los demás errores en el LE. Por ello, en el algoritmo se usa una aproximación que involucra todos los posibles casos, como sigue: la duración de un SS que transita a una FR es  $i + 1$  rondas si  $\beta = 0$  e  $i + 2$  rondas cuando  $\beta > 0$ .

Según el algoritmo de Karn [11], que dice que el ACK de un segmento retransmitido no se debe ocupar para el cálculo del RTT ni para la actualización de la RTO, el mecanismo de *back-off* se restablece cada vez que se reconoce un segmento con SN mayor que  $r$ . Es decir, si  $a > r$ , entonces  $n_{TO} = 0$ .<sup>6</sup> Por lo tanto, ya que la condición  $a > r$  es necesaria para entrar en una FR, siempre que una transición a una FR existe el mecanismo de *back-off* se restablece con  $n_{TO} = 0$ .

La variable  $m$  representa al máximo SN transmitido. Si no hay retransmisiones pendientes, es decir, si todos los LEs previos se recuperaron por completo, en cada instante, el valor de  $m$  equivale al SN del último segmento transmitido registrado en  $s$ . Es decir, la condición  $a > r$  también implica que la variable  $m$  se actualice como  $m = s$ , lo cual sucede siempre al entrar en una FR.

Como ejemplo de lo anterior se tiene la figura 4.3, en la cual se presenta un fragmento de una conexión TCP NewReno a partir del  $SS_{ini}$ . En la tabla se puede revisar cómo se van actualizando las variables globales a su paso por las fases. En el caso de las variables  $A$ ,  $X$  y  $Y$ , están reportados sus incrementos en cada fase en lugar de sus valores absolutos. Las variables se inicializan como  $a = m = r = X = Y = 0$ ,  $A = 0$ ,  $n_{TO} = 0$ ,  $\sigma = \sigma_{ini}$  y  $w = 1$ .  $\sigma_{ini}$  toma un valor arbitrariamente alto [15], que en este caso es de 500, con el fin de que la conexión se mantenga en la fase SS hasta que se presente un estado de congestión incipiente en la red, a partir del cual TCP regula la tasa de transmisión buscando no saturarla. El  $SS_{ini}$  termina con un LE que se presenta por la pérdida del segmento  $\Delta = 30$ , en la ronda  $i = 4$ , según (4.4).

---

<sup>6</sup>Un RA con  $a = r$  forzosamente corresponde con el RA que termina la recuperación de una ventana de pérdida y se genera por una retransmisión exitosa.

Variable\Fase	Inicio	SS <sub>ini</sub>	FR	CA	TO	SS	CA	TO	SS	TO	SS	CA	TO	TO	SS
$a$	0	29	83	83	83	90	90	96	98	100	102	105	107	✓	109
$A$	0	+6	+3	+RTO	+0	+3	+2 RTO	+0	+2+4 RTO	+0	+2	+1+RTO	+2 RTO	+0	+2
$A_{CA\_aux}$	-	-	13	-	-	0	-	-	-	1	-	-	-	-	1
$d$	-	27	0	0	-	0	6	-	2	-	0	2	0	0	0
$\delta$	-	3	1	15	-	0	1	0	1	0	0	1	1	0	0
$m$	0	59	84	98	-	✓	✓	-	101	-	102	108	-	-	109
$n_{TO}$	0	0	0	0	1	✓	✓	2	✓	3	0	0	1	2	0
$N[]$	-	-	12, 13, 13, 1	-	-	-	-	-	-	-	-	-	-	-	-
$r$	0	-	59	-	98	-	-	✓	-	101	-	-	108	✓	-
$s$	0	59	84	98	83	90	97	96	101	100	102	108	107	✓	109
$\sigma$	500	-	15	-	7	-	-	3	-	2	-	-	2	✓	-
$w$	1	30	15	✓	1	7	7	1	3	1	2	3	1	1	2
$X$	0	+56	+27	+0	+0	+7	+6	+0	+4	+0	+2	+5	+0	+0	+2
$Y$	0	+59	+28	+14	+0	+7	+7	+0	+5	+0	+2	+6	+1	+0	+2
$i$	-	4	2	0	-	2	0	-	1	-	1	1	-	-	1
$\beta$	-	14	12	-	-	3	0	-	1	-	0	1	-	-	0

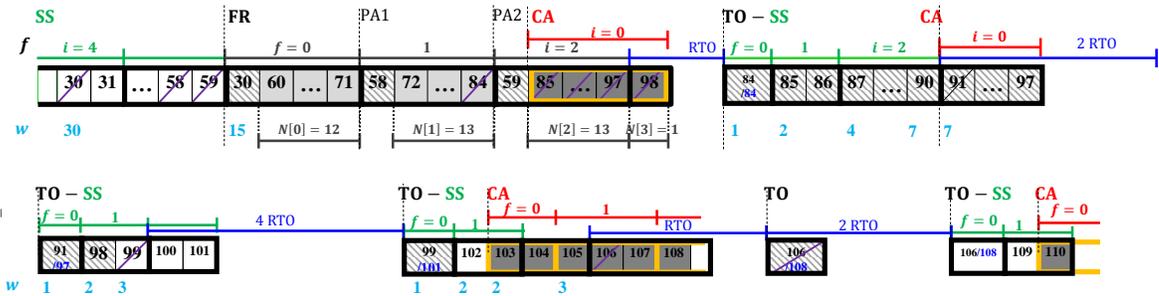


Figura 4.3: Fragmento de una conexión TCP NewReno. Caso A.

La ubicación del LE en la  $cwnd$  correspondiente al inicio de la ronda  $i$  es  $\beta = 14$ , de acuerdo con (4.5). La LW abarca toda la  $cwnd$  que comienza con el segmento  $\Delta$ , cuyo tamaño es  $w = \Delta = 30$ , lo que corresponde con (4.6). La cantidad total de pérdidas en esta fase es  $\delta = 3$ .

Al concluir el  $SS_{ini}$ , la cantidad de segmentos enviados exitosamente antes del LE es, según (4.7),  $a_{SS} = 29$ ; mientras que de acuerdo con (4.8), la cantidad de segmentos enviados en total durante la fase es  $m_{SS} = 59$ . Por lo tanto, las variables globales se actualizan como  $s = 59$ ,  $d = 27$ ,  $X = 56$ ,  $Y = 59$  y  $a = 29$  según (4.9), (4.10), (4.11), (4.12) y (4.13) respectivamente. Ambas condiciones necesarias para la recuperación del LE por medio de una FR se cumplen en este caso; es decir,  $a > r$  y  $d \geq 3$ , por lo que el SS transita hacia una FR. Como  $\beta > 0$ , la duración de la fase se calcula como  $i + 2$ . Dado que se cumple que  $a > r$ , entonces el mecanismo de *back-off* se restablece con  $n_{TO} = 0$  y la variable  $m$ , que representa al máximo SN transmitido, se actualiza como  $m = s$ . Las variables  $r$ ,  $\sigma$  y  $N$  no se modifican al pasar por la fase SS.

#### Transición SS - TO.

Si un SS termina con un LE y alguna de las condiciones que activan la FR no se cumple, es

decir, si sucede que  $d < 3$  ó  $a \leq r$ , entonces el SS transita a la fase TO para comenzar con la recuperación del LE. En este caso, la actualización de las variables se lleva a cabo de la misma forma que cuando una transición SS - FR ocurre, excepto en lo referente a su duración. Es decir, las variables  $s$ ,  $d$ ,  $X$ ,  $Y$  y  $a$  se actualizan según (4.9), (4.10), (4.11), (4.12) y (4.13) respectivamente, a partir de los valores de  $a_{SS}$  y  $m_{SS}$  obtenidos con (4.7) y (4.8). Las variables  $r$ ,  $\sigma$  y  $N$  no se modifican al pasar por la fase SS. La duración del SS equivale a la suma de la RTO, ponderada según el mecanismo de *back-off*, más el tiempo en el que se recibe el último RA. El factor  $n_{TO}$  se restablece cada que se reconoce un segmento con  $SN > r$ , por lo que el valor de  $n_{TO}$  que se usa para ponderar a la RTO es:

$$n_{TO} = \begin{cases} 0 & ; a > r \\ n_{TO} & ; a \leq r \end{cases} \quad (4.20)$$

con el valor de  $a$  ya actualizado al final del SS. La ponderación de la RTO queda como  $2^{n_{TO}}RTO$ , siempre y cuando este valor no exceda el máximo especificado para la duración del temporizador,  $RTO_{max}$ .<sup>7</sup> En cuanto a la recepción del último RA antes del LE, considérense como ejemplo los casos de la figura 4.2. Si el LE se presenta al inicio de la ronda  $f = i$  (i.e.  $\beta = 0$ ), entonces el último RA se recibió al finalizar la ronda anterior ( $i - 1$ ); en cualquier otro caso (es decir, cuando  $\beta > 0$ ), el último RA corresponde con un segmento enviado en la misma ronda  $i$ . Por lo tanto, la duración de la fase SS cuando transita a un TO es de  $i + \text{mín}(RTO_{max}, 2^{n_{TO}}RTO)$  si  $\beta = 0$  y de  $i + 1 + \text{mín}(RTO_{max}, 2^{n_{TO}}RTO)$  cuando  $\beta > 0$ .

Como se mencionó anteriormente, en un  $SS_{ini}$ , cualesquier segmento puede perderse y el LE se puede presentar a partir de la ronda  $i = 0$ . Sin embargo, según la *Aproximación TO - SS*, el LE se puede presentar sólo a partir de la ronda  $i = 1$  en cualquier otro SS (i.e.  $\Delta > 1$ ).

Por otro lado, la condición  $a \leq r$  significa que aún no se han recuperado por completo los LEs previos.<sup>8</sup> Según [61], esto implica que cualesquier pérdida que se produzca en una situación así, forzosamente se recuperará después de activarse un TO. Por lo tanto, la condición  $d < 3$  ó  $a \leq r$ , suficiente para que se produzca un TO, se puede dividir en dos casos:

SS-TO.A Si  $a \leq r$  se genera un TO sin importar la cantidad de DAs que se presentan al final de la fase.

SS-TO.B Si  $a > r$  entonces si se requiere que  $d < 3$  para que se genere un TO.

<sup>7</sup>Según el RFC 2988 [11], el temporizador está sujeto a un máximo de al menos 60 s.

<sup>8</sup>Ver nota al pie 5.

Los valores de  $i$  y  $\beta$  que corresponden a la mínima ubicación del  $\Delta$ -ésimo segmento que satisface la condición  $a > r$  están dados por (4.14) y (4.15), respectivamente. En consecuencia, el primer segmento que puede perderse y dar lugar a un TO por la condición SS-TO.B ocurre en la ronda:

$$i_{\min\text{SSTO.B}} = \begin{cases} i_{\text{SS}r+} & ; \text{SS}_{\text{ini}} \\ \text{máx}(1, i_{\text{SS}r+}) & ; \text{cualquier otro SS} \end{cases} \quad (4.21)$$

en la posición dentro de la  $cwnd$  enviada al inicio de esa ronda:

$$\beta_{\min\text{SSTO.B}} = \begin{cases} \beta_{\text{SS}r+} & ; i = i_{\text{SS}r+} \\ 0 & ; i > i_{\text{SS}r+} \end{cases} \quad (4.22)$$

En último lugar, un TO se puede generar por la pérdida del segmento con el que el SS terminaría para transitar hacia la CA o hacia el EC, lo que suceda primero, lo cual ocurriría con los valores de  $i$  y  $\beta$  como:

$$i_{\max\text{SSTO.B}} = \lfloor \log_2(\text{mín}(\sigma, C_- - a)) \rfloor \quad (4.23)$$

$$\beta_{\max\text{SSTO.B}} = \text{mín}(\sigma, C_- - a) - 2^{i_{\max\text{SSTO.B}}} \quad (4.24)$$

con el valor de  $a$  antes de actualizarse. Por otro lado, para que un TO se presente por la condición SS-TO.A, el  $\Delta$ -ésimo segmento puede ocurrir desde la ronda:

$$i_{\min\text{SSTO.A}} = \begin{cases} 0 & ; \text{SS}_{\text{ini}} \\ 1 & ; \text{cualquier otro SS} \end{cases} \quad (4.25)$$

En cualquier posición dentro de la  $cwnd$  enviada en esa ronda (i.e.  $\beta_{\min\text{SSTO.A}} = 0$ ). Los valores de  $i$  y  $\beta$  que corresponden al último segmento que al perderse cumple con la condición  $a \leq r$  son:

$$i_{\text{SS}r} = \lfloor \log_2(r - a + 1) \rfloor \quad (4.26)$$

$$\beta_{\text{SS}r} = r - a + 1 - 2^{i_{\text{SS}r}} \quad (4.27)$$

respectivamente. De forma que el último segmento que puede perderse y dar lugar a un TO por la condición SS-TO.A ocurriría en con:

$$i_{\max\text{SSTO}_A} = \text{mín}(i_{\max\text{SSTO}_B}, i_{\text{SS}r}) \quad (4.28)$$

$$\beta_{\max\text{SSTO}_A} = \begin{cases} \beta_{\text{SS}r} & ; i = i_{\text{SS}r} < i_{\max\text{SSTO}_B} \\ \beta_{\max\text{SSTO}_B} & ; i = i_{\max\text{SSTO}_B} < i_{\text{SS}r} \\ \text{mín}(\beta_{\max\text{SSTO}_B}, \beta_{\text{SS}r}) & ; i = i_{\max\text{SSTO}_B} = i_{\text{SS}r} \end{cases} \quad (4.29)$$

Como ejemplo de un SS que finaliza en un TO considérese el tercer SS en la figura 4.3, el cual comienza con los valores  $a = 96$ ,  $r = m = 98$ ,  $\sigma = 3$  y  $n_{\text{TO}} = 2$  (ver la explicación de la fase TO). Como es un SS que sigue de un TO, su primera transmisión corresponde a la retransmisión asociada con el TO, la cual es exitosa (ver *Aproximación TO - SS*). Ya que  $\sigma = 3$ , entonces el intervalo que puede dar lugar a un LE es  $1 < \Delta \leq 3$ . El LE ocurre con el segmento  $\Delta = 3$ , en la ronda  $i = 1$ , en la posición  $\beta = 1$  dentro de la *cwnd* enviada al inicio de esa ronda, de acuerdo con (4.4) y (4.5) respectivamente. El modelo de pérdidas produce una LW con  $\delta = 1$  pérdidas. Los segmentos 97 y 98 se enviaron sin pérdida antes del LE, de forma que  $a_{\text{SS}} = 2$  y la transmisión en el SS se extiende hasta el segmento 101, por lo que  $m_{\text{SS}} = 5$ , según (4.7) y (4.8) respectivamente. En consecuencia, el SS finaliza con  $w = 3$ ,  $s = 101$  y  $a = 98$  conforme a (4.6), (4.9) y (4.13), respectivamente. Los segmentos 100 y 101 resultan en un par de DAs, como se prevé con (4.10). La variable  $X$  se actualiza con (4.11) y  $Y$  con (4.12). En este caso, ambas condiciones necesarias para que se active una FR fallan, es decir,  $a = r = 98$  y  $d = 2 < 3$ , por lo que el LE se recupera a partir de un TO. Como  $a \leq r$ , esto implica que el mecanismo de *back-off* sigue activo y  $n_{\text{TO}}$  permanece sin cambio, de acuerdo con (4.20). Justo en esta fase la recuperación de las retransmisiones faltantes (provenientes del LE ocurrido en la primera CA de la figura) se concluye, en este caso con la retransmisión del segmento 98. Esto se refleja en la actualización de la variable  $m$ , del valor  $m = 98$  (que tenía desde la primera CA) a  $m = s = 101$ . El último RA en este SS corresponde al del segmento 98 y se recibe al final de la ronda  $i$  (i.e.  $\beta > 0$ ). Así que la duración de la fase es de  $i + 1 + 2^{n_{\text{TO}}}$  RTO rondas.

#### *Transición SS – CA.*

Durante el SS y mientras que  $w < \sigma$ ,  $w$  se incrementa en uno por cada RA, a partir de  $w = 1$  (ya que  $i_{\text{wnd}} = l_{\text{wnd}} = 1$  SMSS). El estándar marca que el SS se termina cuando se cumple que  $w \geq \sigma$ , dando paso a la fase CA. Específicamente, en el algoritmo se considera que esta transición ocurre al recibir al primer RA siendo  $w = \sigma$ . Considérese la figura 4.4, en la que se presenta un conjunto de casos en los que un SS transita a una CA para distintos valores del

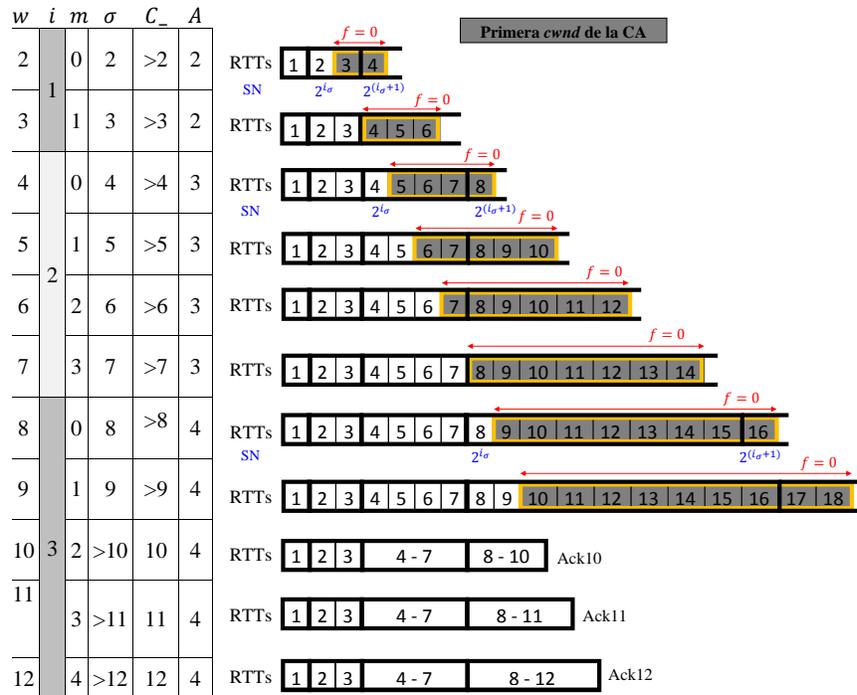


Figura 4.4: Casos de SS sin pérdidas.

umbral del SS,  $\sigma$ . En todos los casos, el SS comienza con una  $w = 1$  en la que se transmite el segmento 1. Con la llegada del RA del segmento 1, al final de la ronda  $f = 0$ , se aumenta  $w = w + 1 = 2$ , de forma que en la siguiente ronda,  $f = 1$ , la *cwnd* abarca los segmentos 2 y 3. Hacia el final de esta ronda, llega el RA del segmento 2. Cuando  $\sigma = 2$ , el RA del segmento 2 ya no provoca un crecimiento de la ventana y produce la transición SS - CA, de forma que la CA comienza con una *cwnd* de tamaño  $w = 2$  desplazada para iniciar con el segmento 3. Por otro lado, si  $\sigma > 2$ , el RA del segmento 2 provoca que  $w = w + 1 = 3$  y la *cwnd* se desplaza para comenzar con el segmento 3. Al recibir el RA del segmento 3, inmediatamente después del RA 2, la *cwnd* se desplaza para comenzar con el segmento 4 y la respuesta del transmisor TCP depende, de nuevo, de la relación  $w$  vs.  $\sigma$ . Si  $\sigma = 3$  (i.e.  $w = \sigma$ ) entonces el SS termina y la CA comienza con este tamaño de *cwnd*. Si  $\sigma > 3$  entonces el RA 3 provoca que  $w = w + 1 = 4$ . Generalizando, la respuesta del transmisor TCP a un RA, estando en SS, se puede describir como:

$w = \sigma$ . El RA del  $\sigma$ -ésimo segmento ya no provoca un crecimiento de la ventana y produce la transición SS - CA, de forma que la CA comienza con una *cwnd* de tamaño  $w = \sigma$  desplazada para iniciar con el segmento  $\sigma + 1$ .

$w < \sigma$ . El RA del  $w$ -ésimo segmento provoca que  $w = w + 1$ . La *cwnd* se desplaza de manera que el valor de  $w$  coincide con el número del segmento al inicio de la *cwnd*.

Usando (4.3), la ronda en la que se produce la transición SS - CA se puede calcular como:

$$i = i_\sigma = \lfloor \log_2 \sigma \rfloor \quad (4.30)$$

Considerando la *cwnd* al inicio de esa ronda, la posición del último segmento enviado antes de dicha transición es:

$$\beta = \beta_\sigma = \sigma - 2^i \quad (4.31)$$

A partir de la explicación anterior, queda claro que un SS sin pérdidas dura  $i + 1$  rondas y concluye con  $w = \sigma$ . Por ejemplo, en todos los casos de la figura 4.4, la *cwnd* transmitida al inicio de la ronda  $f = 2$  es de tamaño  $w = 4$  y abarca los segmentos [4 - 7]. Cuando  $\sigma = 5$ , el RA del segmento 5, que se recibe hacia el final de a ronda  $i = 2$ , provoca que el SS termine y que la *cwnd* se desplace para abarcar los segmentos [6 - 10]. El segmento 5 está ubicado en la posición  $\beta = 1$  de la *cwnd* transmitida al inicio de la ronda  $i$ . El SS dura 3 rondas, es decir, 3 RTTs.

*Aproximación SS - CA.* Como se supone una transmisión en ráfaga al inicio de cada ronda, todos los segmentos posteriores al  $\sigma$ -ésimo segmento transmitidos en la ronda  $i$  en realidad se envían antes de que el SS transite a la CA. Sin embargo, para el algoritmo éstos se consideran como segmentos dentro de la primera *cwnd* transmitida en la CA siguiente. Por ejemplo, en la figura 4.4, al inicio de la ronda  $f = 2$  todos los segmentos de la *cwnd* (i.e. del 4 al 7) se envían en ráfaga. La ronda termina con la recepción de los RAs de todos ellos. Si  $\sigma = 5$ , entonces el SS transita a la CA al finalizar la ronda  $i = 2$ , con la llegada del RA 5. Los RAs de los segmentos 6 y 7 también se reciben al finalizar la ronda  $i$ . Sin embargo, para el algoritmo, estos segmentos se consideran como parte de la CA siguiente. Es decir, se supone que la *cwnd* abarca los segmentos [6 - 10] al inicio de la CA, y que todos ellos se transmiten en ráfaga en su ronda  $f = 0$ . Esta suposición se realiza con el objetivo de que, desde el punto de vista del análisis, una fase CA siempre comience con la transmisión de  $w = \sigma$  segmentos en su ronda cero, y la transición SS - CA coincida siempre con el inicio de la primera ronda de la CA.<sup>9</sup> Como se puede notar en la figura, debido a esta aproximación, las rondas en

---

<sup>9</sup>Para el cálculo de la duración de la CA, en particular para para el caso CA - FR, esta aproximación se mantiene; sin embargo, para los casos CA - EC y CA - TO, esta aproximación se desglosa y se modelan adecuadamente las rondas. Para ello se usa la variable  $A_{CA\_aux}$ , la cual almacena la cantidad de segmentos enviados realmente dentro del SS pero considerados como parte del CA siguiente.

la CA no coinciden con los RTTs en el transcurso de la conexión. Como resultado de esta aproximación, al final de un SS sin pérdidas que transita a una CA se considera que el total de segmentos enviados, todos exitosamente, es igual a  $\sigma$ ; es decir:

$$a_{SS} = m_{SS} = \sigma \quad (4.32)$$

Usando estos valores, las variables  $s$ ,  $d$ ,  $X$ ,  $Y$  y  $a$  se actualizan al finalizar el SS según (4.9), (4.10), (4.11), (4.12) y (4.13), respectivamente. Las variables  $r$ ,  $\sigma$  y  $N$  no se modifican al pasar por la fase SS.

Como ejemplo de un SS sin pérdidas que transita a una CA, considérese el segundo SS en la figura 4.3, el cual comienza con los valores  $a = 83$ ,  $r = m = 98$ ,  $\sigma = 7$  y  $n_{TO} = 1$  (ver la explicación de la fase TO). Como es un SS que sigue de un TO, su primera transmisión, el segmento 84, corresponde a la retransmisión exitosa asociada con el TO (ver *Aproximación TO - SS*). El SS termina al recibir el RA del  $\sigma$ -ésimo segmento transmitido en la fase, en este caso el segmento 90, el cual está ubicado en la posición  $\beta = 3$  de la *cwnd* enviada al inicio de la ronda  $i = 2$ , valores acordes con (4.30) y (4.31). De manera que el SS dura  $i + 1 = 3$  rondas y termina con  $\sigma$  segmentos transmitidos, todos exitosamente, de forma que  $a_{SS} = m_{SS} = 7$ , según (4.32). En consecuencia, el SS finaliza con  $w = 7$  y  $s = a = 90$  conforme a (4.6), (4.9) y (4.13), respectivamente. Ya que es un SS sin pérdidas (i.e.  $\delta = 0$ ), no se producen DAs ( $d = 0$ ), como se prevé con (4.10). Ambas variables  $X$  y  $Y$  se incrementan en  $\sigma$  segmentos, de acuerdo con (4.11) y (4.12). En este caso, el LE anterior abarca hasta el segmento  $r = m = 98$  y el SS termina con el segmento  $s = 90$ , de forma que la transmisión en el SS no es suficientemente larga como para intentar recuperar por completo el último LE. Como consecuencia, la variable  $m$  no se puede actualizar y mantiene su mismo valor con el que entró a la fase. Generalizando, si al concluir una fase no se han retransmitido todas las retransmisiones faltantes (independientemente de si se vuelven a perder o no), entonces  $m$  no se modifica al pasar por esta fase. Es decir, al salir del SS, la variable  $m$  se actualiza como:

$$m = \text{máx}(m, s) \quad (4.33)$$

Puesto que no está recuperado por completo el LE anterior, es decir, como  $a \leq r$ , esto implica que el mecanismo de el mecanismo de *back-off* sigue activo y  $n_{TO}$  permanece sin cambio, de acuerdo con (4.20).

El cuarto SS de la figura 4.3, es otro ejemplo de un SS que termina con una transición SS - CA. Este SS comienza con  $a = 100$ ,  $r = m = 101$ ,  $\sigma = 2$  y  $n_{TO} = 3$  (ver la explicación de la

fase TO). El SS comienza con la retransmisión exitosa del segmento 101, en la ronda  $f = 0$ . Al inicio de la ronda  $f = 1$ , se envía la  $cwnd$  completa (es decir, los segmentos 102 y 103). Puesto que  $\sigma = 2$ , el SS concluye al recibir el RA 102, al final de la ronda  $i = 1$ . Debido a la *Aproximación SS - CA*, aunque el segmento 103 se envió en la última ronda del SS, se considera como si perteneciera a la  $cwnd$  en la ronda  $f = 0$  del CA siguiente. De forma que el SS termina con  $a_{SS} = m_{SS} = 2$  y en consecuencia, con  $w = 2$  y  $s = a = 102$ . En este caso, la transmisión en el SS abarca todas las retransmisiones pendientes (hasta el segmento 101) y nuevas transmisiones (el segmento 102), por lo que  $m = s$  al salir de la fase, como lo establece (4.33). Además, no sólo se realizan todas las retransmisiones faltantes, sino que éstas son exitosas (i.e.  $a > r$  al concluir la fase), lo que permite que el mecanismo de *back-off* se restablezca con  $n_{TO} = 0$ .

#### *Transición SS – EC.*

En la figura 4.4 se muestran varios casos de SSs sin pérdidas. Si el tamaño del flujo es mayor que la transmisión realizada en el SS, entonces el SS termina con una transición hacia la fase CA (en la figura, esto sucede cuando  $\sigma \leq 9$ ). Pero si el flujo termina de transmitirse durante el SS, entonces una transición SS - EC se produce, y la conexión concluye (en la figura, esto ocurre cuando  $C_- = 10, 11$  y  $12$ ). La máxima cantidad de segmentos enviados en un SS que transita a una CA es  $\sigma$ . Por otro lado, si en un SS se termina la transmisión del flujo, la cantidad de segmentos enviados en dicho SS, todos exitosamente, está dada por:

$$a_{SS} = m_{SS} = C_- - a \quad (4.34)$$

con el valor de la variable  $a$  al inicio de la fase, puesto que el SS comienza su transmisión con el segmento siguiente al último segmento reconocido antes de entrar a la fase, registrado en  $a$ . El último segmento del flujo ocurre en la ronda:

$$i_{SSSEC} = \begin{cases} \lceil \log_2(C_- - a) \rceil & ; C_- - a > 0 \\ 0 & ; C_- - a = 0 \end{cases} \quad (4.35)$$

Considerando la  $cwnd$  al inicio de esa ronda, la posición del último segmento del flujo es:

$$\beta_{SSSEC} = \begin{cases} C_- - a - 2^{i_{SSSEC}} & ; C_- - a > 0 \\ 0 & ; C_- - a = 0 \end{cases} \quad (4.36)$$

Usando estos valores, las variables  $w$ ,  $s$ ,  $d$ ,  $X$ ,  $Y$  y  $a$  se actualizan al finalizar el SS según (4.6), (4.9), (4.10), (4.11), (4.12) y (4.13), respectivamente. Las variables  $r$ ,  $\sigma$  y  $N$  no se modifican

al pasar por la fase SS.

La fase concluye con la recepción del RA del último segmento del flujo, al final de la ronda  $i$ , por lo que su duración es de  $i + 1$  rondas. En la figura 4.4, en los casos donde  $C_- = 10, 11$  y  $12$ , como el último segmento del flujo ocurre en la ronda  $i = 3$ , y el SS dura 4 rondas.

Combinando las posibles transiciones que pueden ocurrir desde un SS, la variable  $A$  se actualiza al pasar por esta fase como:

$$A = A + \begin{cases} i + 1 & ; (SS - CA) \text{ OR } (SS - EC) \text{ OR } ((SS - FR) \text{ AND } \beta = 0) \\ i + 2 & ; (SS - FR) \text{ AND } \beta > 0 \\ i + \text{mín}(RTO_{\text{max}}, 2^{n_{\text{TO}}} RTO) & ; (SS - TO) \text{ AND } \beta = 0 \\ i + 1 + \text{mín}(RTO_{\text{max}}, 2^{n_{\text{TO}}} RTO) & ; (SS - TO) \text{ AND } \beta > 0 \end{cases} \quad (4.37)$$

Por último, la variable  $A_{CA\_aux}$  es una variable auxiliar en el cálculo de  $A$  en la CA, cuyo valor se establece en la fase previa a la CA. De forma que esta variable toma sentido en un SS que transita hacia una CA, mientras que en cualquier otra posible terminación del SS, esta variable no se usa. La explicación de cómo se determina su valor se presenta en la siguiente sección.

## 4.2.2. Fase CA

De acuerdo con [15], a partir de  $w = \sigma$ ,  $w$  crece en un SMSS por cada RTT durante la CA, debido a la recepción de  $w$  RAs. Esto sucede desde la ronda  $f = 0$  y hasta la ronda  $f = i$ , en la que una de dos cosas sucede: a) ocurre la primera pérdida en esta fase, o b) se termina el flujo TCP. De forma que el valor de  $w$  se actualiza al pasar por el CA como:

$$w = \sigma + i \quad (4.38)$$

En el algoritmo se usan tres variables auxiliares descritas en la tabla 4.4.

Una CA ocurre después de un SS sin pérdidas o de una cFR (con o sin transmisiones nuevas perdidas). Para fines de análisis, se diferencian dos casos: cuando una CA se inicia estando en un estado sin pérdidas pendientes de recuperar (i.e. con  $\delta = 0$  al entrar a la CA), y el caso contrario (es decir, cuando  $\delta > 0$  al entrar a la fase CA). Como se muestra en la figura 4.1,

Una CA con  $\delta = 0$  a su entrada sigue de un SS o de una cFR sin pérdida de transmisiones nuevas (SS/FR  $\stackrel{\delta=0}{\text{CA}}$ ). Una CA con  $\delta > 0$  inicial sigue de una cFR con transmisiones nuevas perdidas (FR  $\stackrel{\delta>0}{\text{CA}}$ ).

Tabla 4.4 Variables locales de la fase CA.

Variable	Descripción
$a_{CA}$	Representa la cantidad de segmentos enviados exitosamente en la fase antes de la primera pérdida
$\delta_{CA}$	Número de segmentos perdidos en la CA.
$m_{CA}$	Número de segmentos enviados en total en el paso por la CA.

*Transición SS/FR  $\stackrel{\delta=0}{\text{CA}}$  - FR.*

En la figura 4.5 se muestra cómo se actualizan las variables  $w$ ,  $a_{CA}$  y  $m_{CA}$  si un LE se presenta y el fin del flujo está alejado, de forma que la CA termina antes de que el flujo se transmita por completo. La transmisión en la CA comienza en la ronda  $f = 0$  con la emisión en ráfaga de los  $w = \sigma$  segmentos de la  $cwnd$ . Si no hay pérdidas, hacia el final de esta ronda se reciben los  $w$  RAs de todos los segmentos de la  $cwnd$ , lo que provoca el crecimiento de la  $cwnd$  a  $w + 1$ . Esto se repite en cada ronda hasta la ronda  $i$ , en la que un LE se presenta con la pérdida del  $\Delta$ -ésimo segmento transmitido en la fase. De la figura 4.5, es claro que:

$$\sum_{f=0}^{i-1} (\sigma + f) < \Delta \leq \sum_{f=0}^i (\sigma + f) \quad (4.39)$$

De esta desigualdad se puede despejar el valor de  $i$ , que queda como:

$$i = \lceil -\sigma - \frac{1}{2} + \sqrt{(\sigma - \frac{1}{2})^2 + 2\Delta} \rceil \quad (4.40)$$

Y la posición del LE en la  $cwnd$  enviada al inicio de la ronda  $i$ , como:

$$\beta = \Delta - 1 - i(\sigma + \frac{i-1}{2}) \quad (4.41)$$

En la figura 4.5a el segmento  $\Delta$  está ubicado en la posición  $\beta = 0$  de la  $cwnd$  enviada al inicio de la ronda  $i$ , cuyo tamaño es de  $w = \sigma + i$ . En la figura 4.5b,  $0 < \beta < w$ , lo que implica que al finalizar la ronda  $i$  se reciben  $\beta$  RAs correspondientes a los segmentos enviados en esta ronda previos al LE. Puesto que  $\beta < w$ , estos RAs no son suficientes como para provocar un crecimiento en  $w$ .<sup>10</sup> Como resultado, la LW comienza con el segmento  $\Delta$  y abarca  $\beta$  segmentos más enviados en la siguiente ronda. Generalizando, de la figura se puede verificar que, cuando

<sup>10</sup>Esto es válido sólo por la suposición de que  $w$  es entero.

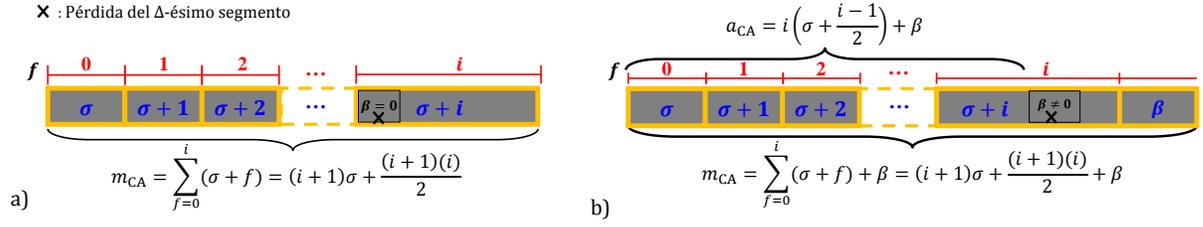


Figura 4.5: CA que termina con un LE. El  $\Delta$ -ésimo segmento ocurre en la *cwnd* enviada al inicio de la ronda  $i$ , ubicado en la posición: a)  $\beta = 0$ , b)  $\beta > 0$ .

el final del flujo está muy alejado, la cantidad de segmentos enviados en total durante una CA que termina en un LE está dada por:

$$m_{CA} = \sum_{f=0}^i (\sigma + f) + \beta = (i+1)\left(\sigma + \frac{i}{2}\right) + \beta = \Delta - 1 + \sigma + i \quad (4.42)$$

También es claro que la cantidad de segmentos enviados sin pérdida antes del LE equivale a:

$$a_{CA} = \sum_{f=0}^{i-1} (\sigma + f) + \beta = (i)\left(\sigma + \frac{i-1}{2}\right) + \beta = \Delta - 1 \quad (4.43)$$

Como se explicó en la fase SS, en el momento en que el LE se presenta, la *cwnd*, de tamaño  $w$ , queda desplazada de tal forma que comienza con el primer segmento perdido. La LW abarca desde el  $\Delta$ -ésimo segmento y termina con el último segmento enviado en esa fase. Si el tamaño del flujo,  $C_-$ , es mayor que la cantidad de segmentos que se alcanzan a transmitir en la CA, entonces la LW mide  $w = \sigma + i$ . En consecuencia, el total de segmentos enviados durante la CA,  $m_{CA}$ , queda como en (1.42). Sin embargo, si la transmisión del flujo termina antes de que la última *cwnd* en la CA se cierre, el tamaño de la LW es menor que  $w$  y el  $m_{CA}$  abarca todos los segmentos desde el inicio de la transmisión en la CA y hasta el último segmento del flujo. Por lo tanto, en cualquier caso, la cantidad de segmentos transmitidos en total en la CA está dada por:

$$m_{CA} = \min(C_- - a, (i+1)\left(\sigma + \frac{i}{2}\right) + \beta) = \min(C_- - a, \Delta - 1 + \sigma + i) \quad (4.44)$$

con el valor de  $a$  al inicio de la fase, puesto que la transmisión en la fase comienza con el SN posterior al último segmento reconocido antes de entrar a la fase (ver explicación de las fases SS y FR). La variable  $s$  representa el SN del último segmento transmitido. Al salir de la CA,  $s$  apunta al último segmento de la LW, es decir,  $s$  se actualiza en la CA como:

$$s = a + m_{CA} \quad (4.45)$$

con el valor de  $a$  al inicio de la fase CA. La variable  $\delta_{CA}$  almacena la cantidad de pérdidas en la LW, por lo que el número de DAs que resulta al final de la CA es:

$$d = m_{CA} - a_{CA} - \delta_{CA} \quad (4.46)$$

El número total de pérdidas en la fase es equivalente a

$$\delta = \delta + \delta_{CA} \quad (4.47)$$

La variable  $X$  se actualiza con la cantidad de segmentos enviados exitosamente, que es la suma de los  $a_{CA}$  segmentos antes del LE mas los  $d$  segmentos no perdidos de la LW, es decir:

$$X = X + a_{CA} + d \quad (4.48)$$

Para la variable  $Y$  se toman en cuenta todos los segmentos enviados durante la CA, de forma que:

$$Y = Y + m_{CA} \quad (4.49)$$

La variable  $a$ , que representa el SN máximo reconocido, termina apuntando al segmento anterior a la primera pérdida. Como  $a$  es acumulativa, al finalizar la CA se actualiza como:

$$a = a + a_{CA} \quad (4.50)$$

La variable  $m$  corresponde al SN transmitido máximo y se actualiza como  $m = s$ , con el valor de  $s$  ya actualizado, siempre y cuando este valor exceda al valor actual de la variable. Si no lo excede, entonces el valor de  $m$  se conserva. Este caso significa que la fase termina sin intentar la recuperación completa de un LE anterior, es decir, que aún existen retransmisiones pendientes. Por lo tanto, usando el valor de  $s$  ya actualizado, se tiene que:

$$m = \text{máx}(m, s) \quad (4.51)$$

Como se mencionó en el caso del SS, una FR se activa si existe al menos una pérdida ( $\delta > 0$ ) y si se cumple con:  $d \geq 3$  (i.e. existe un TD) y  $a > r$ . La condición  $a > r$  asegura que los LEs previos se han recuperado por completo, por lo que se permite la actualización de  $m$ . La condición  $a > r$  también es necesaria para restablecer el mecanismo de *back-off*. Por lo tanto, siempre que existe una transición a una FR se tiene que  $n_{TO} = 0$  y  $m = s$ . Las variables  $A_{CA\_aux}$ ,  $N$ ,  $r$  y  $\sigma$  no se modifican al pasar por la fase CA.

La condición  $d \geq 3$  tiene varias implicaciones:

- El último segmento que al perderse puede activar una FR tiene  $SN = C_- - 3$ . Usando el valor de  $a$  al inicio de la fase, la ronda en donde esto ocurre se calcula como :

$$i_{\max\text{CAFR}} = \begin{cases} i_{\text{CAEC}} & ; \beta_{\text{CAEC}} > 3 \\ i_{\text{CAEC}} - 1 & ; \beta_{\text{CAEC}} \leq 3 \end{cases} \quad (4.52)$$

y la posición del segmento con  $SN = C_- - 3$  dentro de la  $cwnd$  enviada al inicio de esta ronda está dada por:

$$\beta_{\max\text{CAFR}} = \begin{cases} \beta_{\text{CAEC}} - 4 & ; \beta_{\text{CAEC}} > 3 \\ \beta_{\text{CAEC}} - 5 + \sigma + i_{\text{CAEC}} & ; \beta_{\text{CAEC}} \leq 3 \end{cases} \quad (4.53)$$

donde  $i_{\text{CAEC}}$  es la ronda en donde ocurriría el final del flujo si no hubiera pérdidas en la CA, y está dada por:

$$i_{\text{CAEC}} = \lceil -\sigma - \frac{1}{2} + \sqrt{(\sigma + \frac{1}{2})^2 - 2(\sigma - C_- + a)} \rceil \quad (4.54)$$

y  $\beta_{\text{CAEC}}$  es la posición del segmento con  $SN = C_-$  dentro de la  $cwnd$  enviada al inicio de la ronda  $i_{\text{CAEC}}$ . A diferencia de la variable  $\beta$ ,  $\beta_{\text{CAEC}} \in [1, \sigma + i_{\text{CAEC}}]$ , y se calcula como:

$$\beta_{\text{CAEC}} = C_- - a - i_{\text{CAEC}} \left( \sigma + \frac{i_{\text{CAEC}} - 1}{2} \right) \quad (4.55)$$

- $w \geq 4$ , lo que ocurre a partir de la ronda:

$$i_{\text{wmin}} = \text{máx}(0, 4 - \sigma) \quad (4.56)$$

Por otro lado, la condición  $a > r$  implica que el segmento perdido que active la FR tenga un  $SN \geq r + 2$ . Ésto ocurre a partir de la ronda:

$$i_{\text{CAr+}} = \lceil -\sigma - \frac{1}{2} + \sqrt{\sigma^2 - \sigma + \frac{9}{4} + 2 \text{máx}(0, r + 1 - a)} \rceil \quad (4.57)$$

en la posición dentro de la  $cwnd$  enviada al inicio de esta ronda:

$$\beta_{CAr+} = \text{máx}(0, r + 1 - a) - i_{CAr+}(\sigma + \frac{i_{CAr+} - 1}{2}) \quad (4.58)$$

donde  $a$  tiene su valor a la entrada de la fase en ambas expresiones (4.57) y (4.58). Combinando las implicaciones de ambas condiciones sobre los valores de  $i$  y  $\beta$ , a partir de (4.56) - (4.58), cualquier LE en una CA que se pueda recuperar con una FR puede ocurrir a partir de:

$$i_{\min\text{CAFR}} = \text{máx}(i_{w\min}, i_{CAr+}) \quad (4.59)$$

$$\beta_{\min\text{CAFR}} = \begin{cases} \beta_{CAr+} & ; i = i_{CAr+} \\ 0 & ; i \neq i_{CAr+} \end{cases} \quad (4.60)$$

*Aproximación SS/FR $\stackrel{\delta=0}{\rightleftharpoons}$ CA-FR.* Considerando que  $\delta = 0$  al inicio de la CA, la duración de esta fase cuando transita a una FR se aproxima como  $i + 1$  rondas. La duración real oscila entre  $[i, i + 2]$  rondas, dependiendo de la ubicación de las pérdidas en la LW (información no disponible en el algoritmo), del tamaño de la LW y del valor de la variable  $A_{CA\_aux}$ , como se explica con el siguiente ejemplo. En la figura 4.6 se muestran tres casos SS - CA - FR y la CA tiene una duración distinta en cada uno de ellos. La ronda en la que ocurre la transición SS - CA y la posición del segmento que la provoca dentro de la  $cwnd$  enviada al inicio de esa ronda están dados por  $i_\sigma$  y  $\beta_\sigma$ , según (4.30) y (4.31) respectivamente. En todos los casos,  $i_\sigma = 3$ , por lo que el SS dura 4 RTTs. Al inicio de la última ronda del SS,  $i_\sigma$ , todos los segmentos de la  $cwnd$ , de tamaño  $2^{i_\sigma}$ , que abarca del segmento 8 al 15, se transmiten en ráfaga. Con la llegada del RA del segmento  $\sigma$  se produce la transición SS - CA. La variable  $A_{CA\_aux}$  representa la cantidad de segmentos que se enviaron en la última ronda de un SS sin pérdidas pero que, por la *Aproximación SS - CA*, se consideran como enviados al inicio de la fase CA siguiente. En cada caso, la CA tiene una duración distinta dependiendo del valor de  $A_{CA\_aux}$  y de la ubicación del LE y de los segmentos que provocan el TD, como sigue:

- $\sigma = 10$  en el primer caso, por lo que  $A_{CA\_aux} = 5$  representa los segmentos 11 - 15. En la CA, el LE se presenta por la pérdida del segmento 12. El TD se logra con los ACKs de los segmentos 13, 14 y 15. Tanto la pérdida que activa la transición a la FR como el TD se generan por segmentos que realmente fueron transmitidos en el SS anterior. La FR comienza en el quinto RTT. Es decir, la duración de la CA intermedia es de  $i$  rondas.
- En el segundo caso también  $\sigma = 10$  y  $A_{CA\_aux} = 5$ . Sin embargo, en este caso el LE se presenta por la pérdida del segmento 17, y el tercer DA recibido en la CA (correspon-

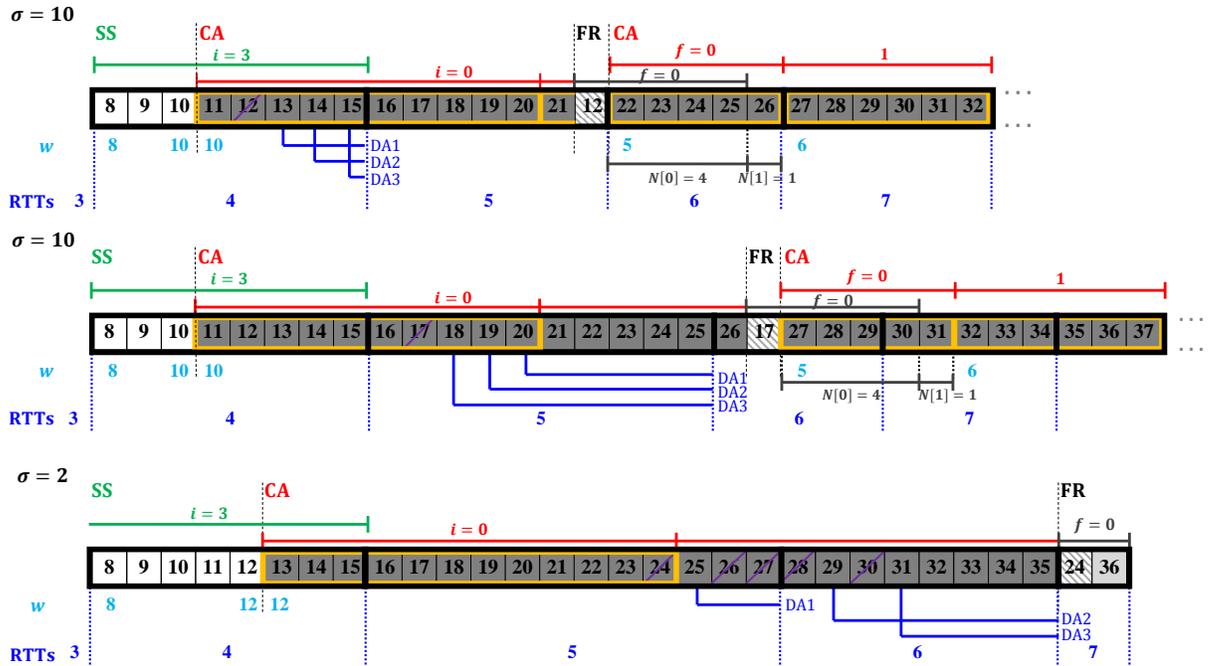


Figura 4.6: Aproximación  $SS/FR \stackrel{\delta=0}{\approx} CA-FR$  Una CA puede durar  $i, i+1$  ó  $i+2$  rondas cuando proviene de una fase sin errores y transita a una FR.

diente al ACK del segmento 20) se genera al finalizar el quinto RTT. De manera que el FR comienza en el RTT 6. Así que la CA dura  $i + 1$  rondas.

- $\sigma = 12$  en el tercer caso, y  $A_{CA\_aux} = 3$  representa a los segmentos 13, 14 y 15. En la CA, el triple DA, correspondiente a los ACKs de los segmentos 25, 29 y 31, se obtiene hasta que el sexto RTT concluye. El FR comienza en el RTT 7. Por lo tanto, la CA dura  $i + 2$  rondas.

Generalizando, la variable  $A_{CA\_aux}$  se puede calcular como el tamaño de la  $cwnd$  al inicio de su ronda  $i_\sigma$  menos los segmentos hasta el segmento  $\sigma$ , es decir:

$$A_{CA\_aux} = 2^{i_\sigma} - \beta_\sigma - 1 \quad (4.61)$$

Como ejemplo de lo anterior se tiene la figura 4.7, en la cual se presenta un fragmento de una conexión TCP NewReno a partir del  $SS_{ini}$ . En la tabla se puede revisar cómo se van actualizando las variables globales a su paso por las fases. En el caso de las variables  $A, X$  y  $Y$ , están reportados sus incrementos en cada fase en lugar de sus valores absolutos. Las variables se inicializan como  $a = m = r = X = Y = 0, A = 0, n_{TO} = 0, \sigma = \sigma_{ini}$  y  $w = 1. \sigma_{ini} = 10$ .

$C_ = 97$	Inicio	SS <sub>ini</sub>	CA	FR	CA	FR	CA	FR	CA	TO	SS	CA
$a$	0	10	16	26	36	43	72	81	81	84	86	97
$A$	0	+4	+1	+1	+2	+3	+6	+5	+RTO	+0	+2	+4
$A_{CA_{aux}}$	-	5	-	4	-	2	-	2	-	-	0	-
$d$	-	0	9	0	3	0	3	0	2	0	0	0
$\delta$	-	0	1	0	3	0	5	1	1	0	0	0
$m$	0	10	26	26	42	43	80	83	85	-	86	97
$n_{RTO}$	0	0	0	0	0	0	0	0	0	1	0	0
$N[\ ]$	-	-	-	{4,1}	-	{0,1,2,1}	-	{0,0,1,2,2,1}	-	-	-	-
$r$	0	-	-	26	-	42	-	80	-	85	-	-
$s$	0	10	26	26	42	43	80	83	85	84	86	97
$\sigma$	10	-	-	5	-	3	-	4	-	2	-	-
$w$	1	10	10	5	6	3	8	4	4	1	2	5
$X$	0	+10	+15	+1	+13	+4	+32	+7	+2	+0	+2	+11
$Y$	0	+10	+16	+1	+16	+4	+37	+8	+2	+0	+2	+11
$i$	-	3	0	0	1	2	5	4	0	-	1	3
$\beta$	-	2	6	-	5	-	4	-	-	-	1	2

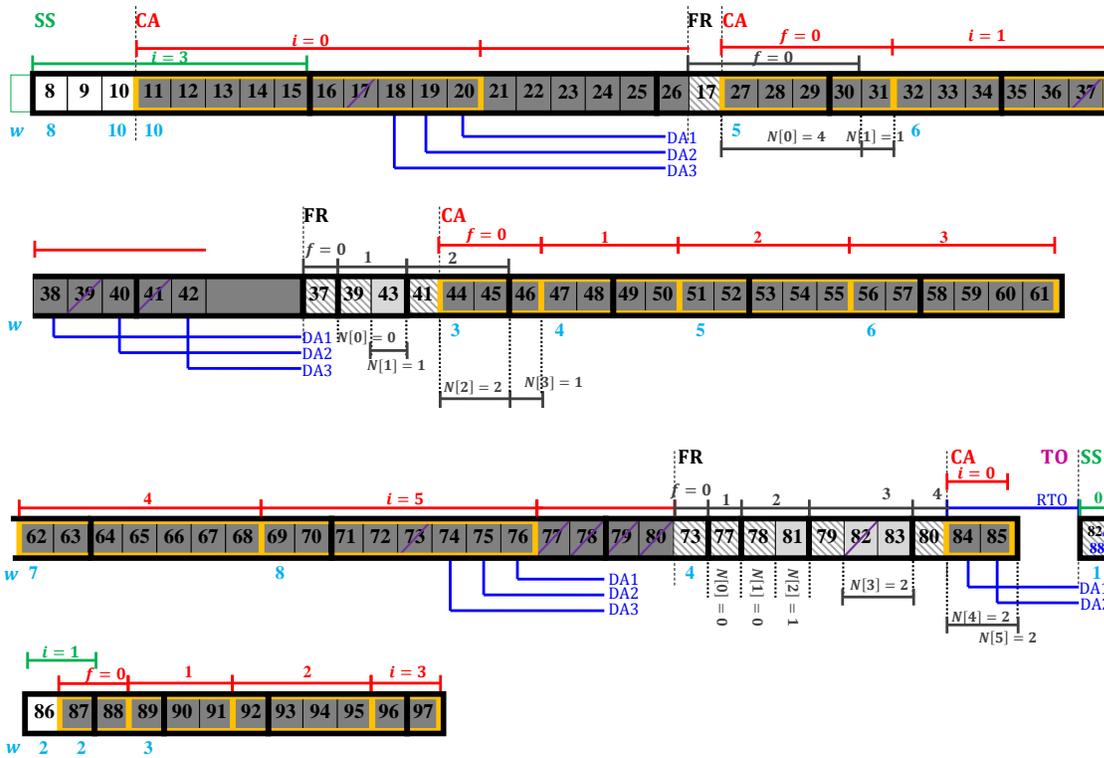


Figura 4.7: Fragmento de una conexión TCP NewReno. Caso B.

El  $SS_{ini}$  termina sin pérdidas, por lo que transita a una CA al recibir el RA del  $\sigma$ -ésimo segmento, enviado en su ronda  $i_\sigma = 3$ , en la posición  $\beta_\sigma = 2$ .  $A_{CA.aux} = 5$  de acuerdo con (4.61). El  $SS_{ini}$  finaliza con los valores  $m = s = a = 10$ . La CA siguiente realmente comienza en el RTT 5, con la transmisión del segmento 16 y termina un RTT después con la recepción del TD correspondiente a los ACKs de los segmentos 18, 19 y 20. La transmisión del segmento 26 es inmediatamente antes de recibir el TD. Sin embargo, debido a la *Aproximación SS - CA*, se considera que la ronda cero de la CA comienza con la transmisión a partir del segmento 11. Como se puede notar de la figura, debido a esta aproximación, las rondas en la CA no coinciden con los RTTs en el transcurso de la conexión. El LE se presenta con la pérdida del  $\Delta$ -ésimo segmento enviado durante la CA ( $\Delta = 7$ ), en la ronda  $i = 0$ , según (4.40). La posición de este segmento en la *cwnd* enviada al inicio de esa ronda es  $\beta = 6$ , según (4.41). La LW, de tamaño  $w = 10$  de acuerdo con (4.38), comienza con el segmento 17 y termina con el 26 y sólo contiene  $\delta = \delta_{CA} = 1$  pérdidas. En esta CA, la cantidad de segmentos enviados exitosamente antes del LE es, según (4.43),  $a_{CA} = 6$ ; mientras que de acuerdo con (4.44), la cantidad de segmentos enviados en total durante la fase es  $m_{CA} = 16$ . Por lo tanto, las variables globales se actualizan como  $s = 26$ ,  $d = 9$ ,  $X = X + 15$ ,  $Y = Y + 16$  y  $a = 16$  según (4.45), (4.46), (4.48), (4.49) y (4.50) respectivamente. Ambas condiciones necesarias para la recuperación del LE por medio de una FR se cumplen en este caso; es decir,  $a > r$  y  $d \geq 3$ , por lo que la CA transita hacia una FR. En este caso, el TD se produce con el ACK del segmento 20, un RTT después de que el SS previo termina, de forma que la duración real de la CA es de una ronda, por lo que la *Aproximación SS/FR $\stackrel{\delta=0}{=}CA-FR$*  es adecuada. Dado que se cumple que  $a > r$ , entonces el mecanismo de *back-off* se restablece con  $n_{TO} = 0$ . Las variables  $A_{CA.aux}$ ,  $N$ ,  $r$  y  $\sigma$  no se modifican al pasar por la fase CA. En este caso y de acuerdo con (4.51), la variable  $m$ , que representa al máximo SN transmitido, toma el valor de  $s$ , pues no hay LEs previos.

La segunda CA en esta figura es otro ejemplo de una transición *SS/FR $\stackrel{\delta=0}{=}CA - FR$* . Este CA se presenta después de una cFR que concluye con los valores  $w = 5$ ,  $m = s = a = 26$  y  $A_{CA.aux} = 4$  (ver la explicación de la fase FR). El LE ocurre con la pérdida del  $\Delta$ -ésimo segmento enviado en la CA ( $\Delta = 11$ ), en la ronda  $i = 1$ , según (4.40). La LW comienza con el segmento 37 y termina con el 42. Hay  $\delta = 3$  pérdidas en la LW, ubicadas de tal forma que el TD se produce al finalizar la ronda  $i + 1$  gracias al ACK del segmento 42. La duración real de esta CA es de tres rondas, a partir de la transmisión del segmento 30 (ver la explicación de la fase FR). Sin embargo, su duración se aproxima como  $i + 1 = 2$  rondas, de acuerdo con la *Aproximación SS/FR $\stackrel{\delta=0}{=}CA-FR$* . En este caso y de acuerdo con (4.51), la variable  $m$  toma el valor de  $s$ , pues no hay retransmisiones pendientes por reintentar.

### Transición FR $\delta > 0$ CA - FR.

En este caso, la transición CA - FR se activa por la pérdida de cualesquier nueva transmisión dentro de la cFR anterior, excepto la primera (para cumplir con la condición  $a > r$ ). Por lo tanto, en la CA no se reciben RAs, es decir, todas las transmisiones exitosas en la CA generarán sólo DAs. Según la suposición 10 de la sección 4.1.4, en una CA en estas condiciones no se presenta un LE y las  $\delta_{CA}$  pérdidas que existan durante su transmisión forma parte del LE que comenzó en la cFR previa. Para que una FR se active, se debe cumplir que  $d \geq 3$  y  $a > r$ , lo cual implica que:

- El cFR anterior termina con la recepción de un RA total (ver la explicación de la fase FR). Por lo tanto, el TD se conforma por los ACKs de los tres primeros segmentos no perdidos enviados en la ronda  $i = 0$  de la CA. Consecuentemente, esta fase dura sólo una ronda.
- Como no se recibe ningún RA dentro de la CA, no existe crecimiento de  $w$ , lo que coincide con (4.38).
- Todas las transmisiones en esta CA se consideran como parte del LE que comenzó en la cFR anterior, por lo que no hay ningún segmento transmitido antes del LE, es decir:

$$a_{CA} = 0 \quad (4.62)$$

y  $a$  queda sin cambio al pasar por esta fase, como lo prevé (4.50).

- La cantidad de segmentos enviada en esta CA es suficiente para producir el TD, de forma que  $m_{CA} \geq 3$ . Como se explica en la fase FR, específicamente, el valor de  $m_{CA}$  en una condición así está dado por:

$$m_{CA} = N[f_{FR} - 1] + N[f_{FR}] \quad (4.63)$$

- El SN del último segmento transmitido en la cFR previa a la CA está guardado en la variable  $m$  (ver la explicación de la fase FR). La transmisión en la CA prosigue a dicho segmento, por lo que el último segmento transmitido en la CA,  $s$ , se calcula como:

$$s = m + m_{CA} \quad (4.64)$$

donde la variable  $m$  tiene su valor a la entrada de la CA.

- Las  $\delta_{CA}$  pérdidas en la CA se suman a las  $\delta$  pérdidas del LE anterior de acuerdo con (4.47).
- La condición  $a > r$  provoca el restablecimiento del mecanismo de *back-off*. Por lo tanto, siempre que existe una transición a una FR se tiene que  $n_{TO} = 0$ .
- Como se menciona en la explicación de la fase FR, siempre se cumple que  $m = s$  al finalizar una cFR puesto que no quedan retransmisiones pendientes por enviar. Como la fase CA sigue de una cFR, esta condición se mantiene, por lo que al finalizar la CA también se tiene que  $m = s$ , lo cual coincide con lo predicho con (4.51).

Usando (4.62) y (4.63), las variables  $d$ ,  $X$  y  $Y$  se actualizan al finalizar la CA según (4.46), (4.48) y (4.49), respectivamente. Las variables  $A_{CA\_aux}$ ,  $N$ ,  $r$  y  $\sigma$  no se modifican al pasar por la fase CA.

*Aproximación  $FR \xrightarrow{\delta > 0} CA-FR$ .* Considerando que  $\delta > 0$  al inicio de la CA, la duración de esta fase cuando transita a una FR se calcula desglosando los casos de la *Aproximación FR-CA* (ver su explicación en la fase FR) como sigue: los  $N[f_{FR} - 1]$  segmentos en realidad se transmiten en la última ronda de la cFR anterior, mientras que los  $N[f_{FR}]$  segmentos se transmiten en la ronda siguiente al término de la cFR, es decir, dentro de la CA. Si  $N[f_{FR}] = 0$ , forzosamente el triple DA se genera por los  $N[f_{FR} - 1]$  segmentos transmitidos antes de comenzar la CA y su duración equivale a  $i$  rondas. Sin embargo, si  $N[f_{FR}] > 0$ , entonces el triple DA se puede generar por segmentos exitosos dentro de los  $N[f_{FR} - 1]$  o de los  $N[f_{FR}]$ , y por ello la duración del CA varía entre  $[i, i + 1]$  rondas. Como esta información no se tiene en el algoritmo, se aproxima la duración de la CA en este caso como  $i + 1$  rondas (que bien podría sustituirse por la aproximación alternativa de  $i$  rondas). Por lo tanto, en una situación así, la variable  $A$  se actualiza a su paso por esta fase como:

$$A = A + \begin{cases} i & ; N[f_{FR}] = 0 \\ i + 1 & ; N[f_{FR}] > 0 \end{cases} \quad (4.65)$$

Un ejemplo de una transición  $FR \xrightarrow{\delta > 0} CA - FR$  se presenta en la figura 4.8. La FR que sigue del SS es una cFR que termina con tres transmisiones nuevas perdidas ( $\delta = 3$ ), correspondientes a los segmentos 69 - 71. La primera transmisión nueva en la FR (el segmento 60) no se pierde, lo que permite que  $a > r$  al término de la cFR. En este caso,  $N[f_{FR} - 1] = 11$  y  $N[f_{FR}] = 0$  (ver la explicación de la fase FR). Debido a la *Aproximación FR-CA*, se supone que los segmentos que constituyen la *cwnd* en la primera ronda de la CA posterior a la cFR,

Variable\Fase	SS	FR	CA	FR	CA	TO	SS
$a$	29	68	✓	92	93	94	96/97
$A$	+6	+3	+0	+4	+1 + RTO	+0	+2
$A_{CA,aux}$	-	11	-	0	-	-	0
$d$	27	0	10	0	1	-	0
$\delta$	3	3	4	0	6	0	0
$m$	59	81	92	92	100	-	-
$\pi_{TO}$	0	0	0	0	✓	1	✓
$N[\cdot]$	-	[12, 10, 11, 0]	-	{0, 0, 0, 7}	-	-	-
$r$	0	59	-	92	-	100	-
$s$	59	81	92	92	100	94	97
$\sigma$	500	15	-	7	-	3	-
$w$	30	15	✓	7	7	1	3
$X$	+56	+22	+10	+4	+2	+0	+3
$Y$	+59	+25	+11	+4	+8	+0	+3
$i$	4	2	0	3	0	-	1
$\beta$	14	9	-	-	1	-	1

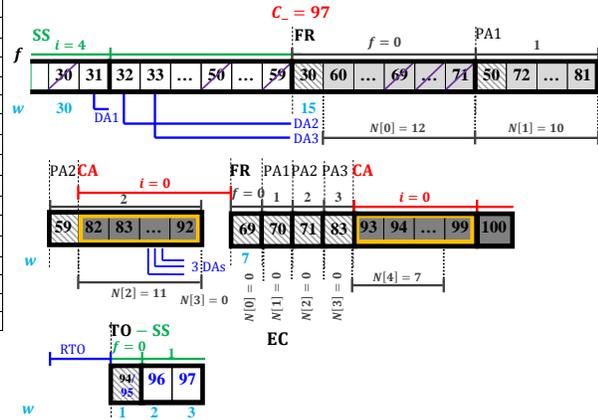


Figura 4.8: Fragmento de una conexión TCP NewReno. Caso C.

en este caso los segmentos [82 - 92], abarcan desde los  $N[f_{FR} - 1]$  segmentos que realmente se transmitieron en la última ronda de la cFR hasta los  $N[f_{FR}]$  segmentos que si se transmiten después de la transición FR - CA. Por ello,  $m_{CA} = 11$ , de acuerdo con (4.63). Todos los segmentos considerados como transmitidos en la CA forman parte del LE que comenzó en la cFR previa, por lo que  $a_{CA} = 0$ , como lo prevé (4.62) y  $a$  no cambia, como lo establece (4.50). El segmento 83 se pierde ( $\delta_{CA} = 1$ ), lo que causa que se actualice  $\delta = 4$  con (4.47). Por lo tanto, y de acuerdo con (4.46), se generan  $d = 10$  DAs. Es importante notar que si hubiera sucedido que  $\delta_{CA} \geq 9$ , causaría un TO en lugar de una nueva FR. Puesto que el triple DA se produce dentro de los  $N[f_{FR} - 1]$  segmentos, las transiciones consecutivas FR -CA y CA - FR ocurren en el mismo RTT. Es decir, al finalizar la última ronda de la cFR se recibe el RA total con el que concluye, pero casi inmediatamente después se produce el TD. Por lo tanto, la CA dura  $i = 0$  rondas, como lo indica (4.65). Las variables  $s$ ,  $X$  y  $Y$  se actualizan según (4.64), (4.48) y (4.49), respectivamente y  $m = s$ .

Otro ejemplo se muestra en la figura 4.9 en el cual se requieren de los  $N[f_{FR}] > 0$  segmentos para generar el triple DA, un RTT después de que la cFR termina. De los  $N[f_{FR} - 1] = 7$  segmentos enviados en la última ronda de la cFR, los segmentos del 85 al 89 se pierden, por lo que no se generan suficientes DAs para obtener el TD. Éste se completa con el ACK del segmento 92, el cual forma parte de los  $N[f_{FR}]$  segmentos enviados después de la cFR. En este caso, la CA dura  $i + 1 = 1$  rondas como lo establece (4.65). Si, por ejemplo, alguno de los segmentos entre el 85 y el 89 no se hubiera perdido, el triple DA se hubiera generado al finalizar la cFR y la duración de la CA hubiera sido de  $i = 0$  rondas, es decir, la Aproximación  $FR \xrightarrow{\delta > 0} CA$ -FR resultaría sobrada.

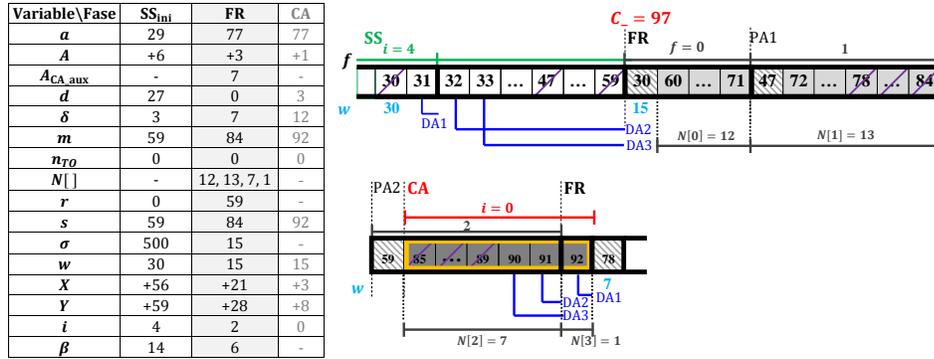


Figura 4.9: Fragmento de una conexión TCP NewReno. Caso D.

$SS/FR \stackrel{\delta=0}{=} CA - TO$ .

Si una CA termina con un LE y alguna de las condiciones que activan la FR no se cumple, es decir, si sucede que  $d < 3$  ó  $a \leq r$ , entonces la CA transita a la fase TO para comenzar con la recuperación del LE. Al igual que cuando se transita a una FR, se distingue entre dos casos: cuando  $\delta = 0$  al inicio de la fase y el caso contrario.

Si  $\delta = 0$  al inicio de la CA, entonces la CA prosigue a una fase sin pérdidas. Es decir, la CA sigue de un SS cuya transmisión fue de  $\sigma$  segmentos sin pérdida, o la CA sucede a una cFR que no tuvo transmisiones nuevas perdidas. En esta situación, la actualización de las variables se lleva a cabo de la misma forma que cuando una transición  $SS/FR \stackrel{\delta=0}{=} CA - FR$  ocurre, excepto en lo referente a su duración. Es decir, las variables  $w$ ,  $d$ ,  $\delta$ ,  $X$ ,  $Y$ ,  $a$  y  $m$  se actualizan según (4.38), (4.46), (4.47), (4.48), (4.49), (4.50) y (4.51), respectivamente, a partir de los valores  $i$ ,  $a_{CA}$  y  $m_{CA}$  obtenidos según (4.40), (4.43) y (4.44), respectivamente. En este caso, la variable  $s$  se actualiza de acuerdo con (4.45). Las variables  $A_{CA\_aux}$ ,  $N$ ,  $r$  y  $\sigma$  no se modifican al pasar por la fase CA.

La duración de la CA equivale a la suma de la RTO, ponderada según el mecanismo de *back-off*, más el tiempo en el que se recibe el último RA. El factor  $n_{TO}$  se restablece cada que se reconoce un segmento con  $SN > r$ , por lo que el valor de  $n_{TO}$  que se usa para ponderar a la RTO está dado como en (4.20) con el valor de  $a$  ya actualizado al final de la CA. La ponderación de la RTO queda como  $2^{n_{TO}} RTO$ , siempre y cuando este valor no exceda el máximo

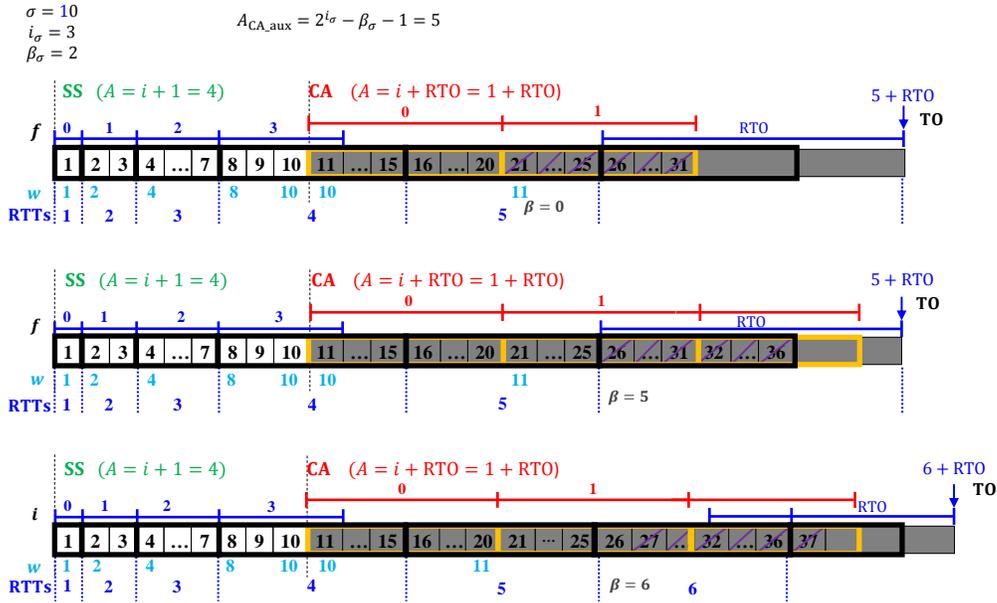


Figura 4.10: Transiciones SS - CA - TO.

especificado para la duración del temporizador,  $RTO_{max}$ .<sup>11</sup> Para calcular el lapso de tiempo en el que se recibe el último RA, las aproximaciones *Aproximación SS-CA* y *Aproximación FR-CA* se desglosan como sigue:

La *Aproximación SS-CA* establece que los segmentos transmitidos en la ronda  $i$  del SS, después del  $\sigma$ -ésimo segmento, se tomen en cuenta como parte de la CA siguiente, y supone que la transición SS - CA coincide con el inicio de la primera ronda de la CA. De esta forma, existe un traslape entre parte del último RTT del SS y la ronda cero de la CA. Este desfase entre los RTTs y las rondas de la CA se mantiene durante toda la fase. Consecuentemente, los primeros  $A_{CA\_aux}$  segmentos en cada ronda se transmiten un RTT antes que los segmentos posteriores en la misma ronda. Por ejemplo, en la figura 4.10 el SS termina en el cuarto RTT, con la recepción del segmento  $\sigma = 10$ . Los segmentos 11 al 15 se transmiten en la última ronda del SS,  $i_\sigma$ . Sin embargo, debido a la *Aproximación SS-CA*, se consideran como transmitidos en la primera ronda de la CA. Estos segmentos se contabilizan en la variable  $A_{CA\_aux} = 5$ . Como se puede notar en la figura, debido a esta aproximación, las rondas en la CA no coinciden con los RTTs en el transcurso de la conexión y de ahí la necesidad de separar los distintos casos en los que la duración de la CA, cuando termina en un TO, es distinta. En el primer escenario presentado en la figura, el LE con el que termina la CA ocurre en su ronda  $i = 1$  y se debe a la pérdida del segmento 21. El último RA (correspondiente al SN = 20) se recibe

<sup>11</sup>Ver la nota al pie de página 7.

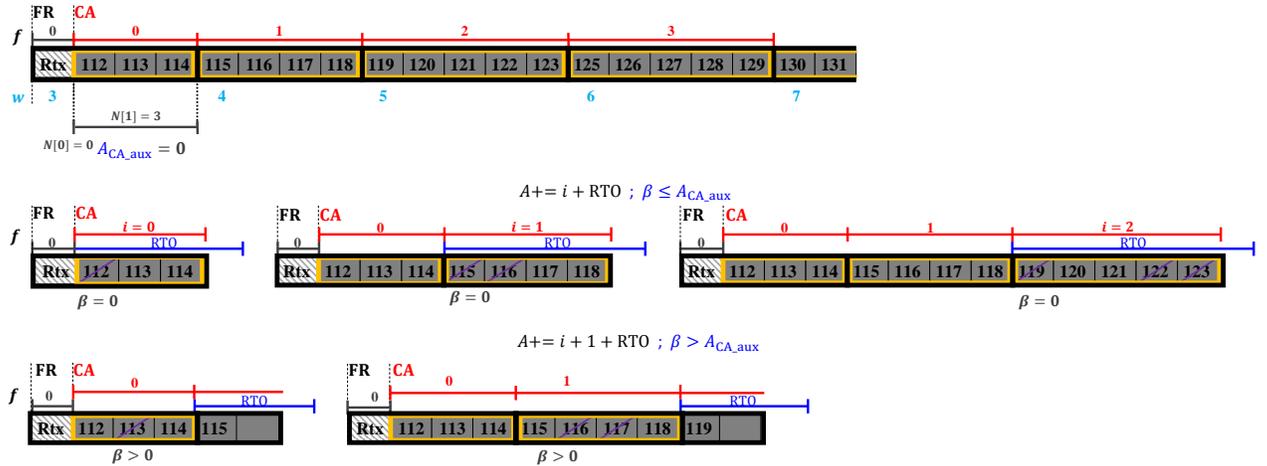


Figura 4.11: Transiciones FR - CA - TO.

al finalizar el RTT 5, es decir,  $i$  rondas después de terminado el SS. En el segundo escenario, sucede lo mismo pero el último RA corresponde al segmento 25. En ambos casos  $\beta \leq A_{CA\_aux}$ . Por el contrario, en el tercer escenario, aunque el LE también se presenta en la ronda  $i = 1$ ,  $\beta > A_{CA\_aux}$ , y el último RA, que corresponde al SN = 26, se recibe  $i + 1$  rondas después del SS. De estos ejemplo es claro que la duración de la conexión, en una transición SS - CA - TO, se actualiza la salir de la CA como:

$$A = A + \begin{cases} i + \min(\text{RTO}_{\max}, 2^{n_{\text{TO}}} \text{RTO}) & ; \beta \leq A_{CA\_aux} \\ i + 1 + \min(\text{RTO}_{\max}, 2^{n_{\text{TO}}} \text{RTO}) & ; \beta > A_{CA\_aux} \end{cases} \quad (4.66)$$

La *Aproximación FR-CA* establece que los  $A_{CA\_aux} = N[f_{FR} - 1]$  segmentos transmitidos en la última ronda de la cFR se tomen en cuenta como parte de la CA siguiente y supone que la transición FR - CA coincide con el inicio de la primer ronda de la CA. De esta forma, la ronda cero de la CA se traslapa en parte con el último RTT de la FR y este desfase entre los RTTs y las rondas de la CA se mantiene durante toda la fase, de manera que los primeros  $A_{CA\_aux}$  segmentos en cada ronda se transmiten un RTT antes que los segmentos posteriores en la misma ronda. Por ejemplo, en la figura 4.11 se muestran varios casos de transición FR  $\delta=0$  CA - TO. En todos ellos, la transición FR - CA ocurre de la misma forma, como se muestra en la parte a). Se trata de una cFR que se activa para recuperar sólo una pérdida, es decir,  $\delta = 1$  al iniciar la fase. La FR dura una ronda y tiene  $A_{CA\_aux} = N[f_{FR} - 1] = 0$ . En estos casos en particular, la ronda cero de la CA sí coincide con el inicio de un RTT debido a que  $A_{CA\_aux} = 0$ , es decir, no hay segmentos que se hayan transmitido en la cFR que se consideren dentro de la CA. En la parte b) de la figura 4.11 se muestran tres casos en los que la CA

termina con un LE. El LE ocurre en una posición  $\beta \leq A_{CA\_aux}$  dentro de la *cwnd* transmitida al inicio de la ronda  $i$ . En el primer caso  $i = 0$ , en el segundo  $i = 1$  y en el tercer caso  $i = 2$ . En los tres casos, el último RA corresponde a un segmento que se envió en el RTT anterior a la ocurrencia del LE, es decir  $i$  rondas después de la cFR. En la parte c) de la figura 4.11 aparecen dos casos más en los que la CA termina con un LE. Sin embargo, en éstos, el LE ocurre en una posición  $\beta > A_{CA\_aux}$  dentro de la *cwnd* transmitida al inicio de la ronda  $i$ . En ambos casos, el último RA corresponde a un segmento que se envió en el mismo RTT en donde el LE sucede, es decir  $i + 1$  rondas después de la cFR. De estos ejemplo es claro que la duración de la conexión se actualiza según (4.66) a su paso por la CA, en una transición FR  $\delta=0$  CA - TO.

De forma semejante a como se analizó en el SS, la condición  $d < 3$  ó  $a \leq r$ , suficiente para que se produzca un TO, se puede dividir en dos casos:

CA-TO.A Si  $a \leq r$  entonces no importa la cantidad de DAs que se presentan al final de la fase.

CA-TO.B Si  $a > r$  entonces sí se requiere que  $d < 3$  para que se genere un TO.

El primer segmento que puede perderse y dar lugar a un TO por la condición CA-TO.B ocurre con  $i_{\min\text{CATO.B}}$  y  $\beta_{\min\text{CATO.B}}$  dados por (4.57) y (4.58), respectivamente. Por otro lado, un TO se puede generar en último lugar por la pérdida del segmento con  $\text{SN} = C_-$ , lo cual ocurriría con los valores de  $i$  y  $\beta$  dados por (4.54) y (4.55),<sup>12</sup> respectivamente.

Por otro lado, para que un TO se presente por la condición CA-TO.A, el  $\Delta$ -ésimo segmento puede ocurrir desde la ronda  $i_{\min\text{CATO.A}} = 0$ , en la primera posición dentro de la primera *cwnd* (i.e.  $\beta_{\min\text{CATO.A}} = 0$ ). Los valores de  $i$  y  $\beta$  que corresponden al último segmento que al perderse cumple con la condición  $a \leq r$  son:

$$i_{CAr} = \lceil -\sigma - \frac{1}{2} + \sqrt{\sigma^2 - \sigma + \frac{9}{4} + 2 \text{máx}(0, r - a)} \rceil \quad (4.67)$$

en la posición dentro de la *cwnd* enviada al inicio de esta ronda:

$$\beta_{CAr} = \text{máx}(0, r - a) - i_{CAr+}(\sigma + \frac{i_{CAr+} - 1}{2}) \quad (4.68)$$

respectivamente. De forma que el último segmento que puede perderse y dar lugar a un TO por la condición CA-TO.A ocurriría en con:

<sup>12</sup>Es preciso recordar que  $\beta_{CAEC} \in [1, \sigma + i_{CAEC}]$ .

$$i_{\max\text{CATO\_A}} = \min(i_{\text{CAEC}}, i_{\text{CAr}}) \quad (4.69)$$

$$\beta_{\max\text{CATO\_A}} = \begin{cases} \beta_{\text{CAr}} & ; i = i_{\text{CAr}} < i_{\text{CAEC}} \\ \beta_{\text{CAEC}} & ; i = i_{\text{CAEC}} < i_{\text{CAr}} \\ \min(\beta_{\text{CAEC}}, \beta_{\text{CAr}}) & ; i = i_{\text{CAEC}} = i_{\text{CAr}} \end{cases} \quad (4.70)$$

Como ejemplo de una transición SS - CA  $\underline{a \leq r}$  TO considérese el fragmento de una conexión TCP NewReno mostrado en la figura 4.3. La segunda CA prosigue de una fase SS que concluye con los valores  $s = a = 90$ ,  $r = m = 98$ ,  $n_{\text{TO}} = 1$  y  $w = \sigma = 7$  (ver la explicación de la fase SS).  $A_{\text{CA\_aux}} = 0$ , lo que significa que en este caso sí coincide la primera ronda de la CA con los RTTs. El LE se presenta con la pérdida del primer segmento enviado en la CA ( $\Delta = 1$ ), lo que implica que  $i = 0$  y  $\beta = 0$ , de acuerdo con (4.40) y (4.41), respectivamente. De la figura es claro que  $a_{\text{CA}} = 0$  y  $m_{\text{CA}} = 7$ , como lo establecen (4.43) y (4.44). La LW queda con un tamaño dado por  $w = 7$ , según (4.38). No hay pérdidas adicionales en el LE, por lo que  $\delta_{\text{CA}} = 1$ . Por lo tanto, según (4.46), al finalizar la ronda  $i$  se reciben  $d = 6$  DAs. Como la CA sigue de un SS sin pérdidas,  $\delta = \delta_{\text{CA}} = 1$  al finalizar la fase, de acuerdo con (4.47).  $a = 90$  y  $s = 97$  como lo determinan (4.50) y (4.45) respectivamente. Las variables  $X$  y  $Y$  se actualizan con (4.48) y (4.49). En este caso, la transmisión en la CA abarcó únicamente retransmisiones pendientes del último LE, y una de ellas se perdió para dar lugar a un nuevo LE. Por lo tanto, el valor de  $m$  quedó sin cambio, como lo indica (4.51). Las variables  $A_{\text{CA\_aux}}$ ,  $N$ ,  $r$  y  $\sigma$  no se modifican al pasar por la fase CA. Aunque un TD se produce, la condición  $a \leq r$  evita que una FR se active para la recuperación del nuevo LE. Adicionalmente, la condición  $a \leq r$  evita que el mecanismo de *back-off* se restablezca y el  $n_{\text{TO}}$  se mantiene con su mismo valor. En este caso no se recibe ningún RA en la CA, puesto que el LE se da con el primer segmento transmitido en la fase, por lo que la CA dura 2RTO rondas, de acuerdo con (4.66).

Como ejemplo de una transición SS - CA  $\underline{a > r \text{ AND } d < 3}$  TO considérese la tercera CA en la figura 4.3, que también prosigue de una fase SS. En este caso, el segmento 103 se transmite en la última ronda del SS pero se considera como transmitido en la ronda cero de la CA, según la *Aproximación SS-CA*, por lo que  $A_{\text{CA\_aux}} = 1$ . Esto implica que las rondas en la CA no coinciden con los RTTs, como se muestra en la figura. El LE se presenta por la pérdida del  $\Delta$ -ésimo segmento ( $\Delta = 4$ ), el cual está ubicado en la posición  $\beta = 1$  de la *cwnd* enviada al inicio de la ronda  $i = 1$ . La LW abarca los segmentos [106 - 108]. Al finalizar esta fase,  $a = 105$ , de forma que la condición  $a > r$  se cumple, lo que permite que se restablezca  $n_{\text{TO}} = 0$  y se actualice el valor de  $m$  con el SN del último segmento transmitido en la CA,  $s = 108$ . De

nuevo, sólo hay  $\delta_{CA} = 1$  pérdidas en la LW y los segmentos 107 y 108 resultan en dos DAs (i.e.  $d = 2$ ). Ya que no se logra un TD, la CA termina con un TO. De la figura se observa que aunque los segmentos 103 y 104 pertenecen a la misma ronda en la CA, en realidad se envían en distintos RTTs. El segmento 104 se envía un RTT después que el segmento 103. El ACK del segmento 103 provoca la emisión del segmento 105 y el ACK del segmento 104 genera el permiso para emitir los segmentos 106 y 107. En consecuencia, el segmento 105 se envía un RTT antes que los segmentos 106 y 107. El último RA corresponde al segmento 105, el cual se recibe  $i$  RTTs después de que el SS terminó. Como se explicó anteriormente, esta situación se sintetiza con la relación  $\beta \leq A_{CA.aux}$ , con la que la duración de la conexión se actualiza a su paso por esta fase como en (4.66).

Como ejemplo de una transición FR  $\delta=0$  - CA  $a>r$ AND $d<3$  TO, considérese el fragmento de una conexión TCP NewReno mostrado en la figura 4.8. La segunda CA prosigue de una fase FR que concluye sin pérdida de nuevas transmisiones ( $\delta = 0$ ), con los valores  $m = s = a = 92$ ,  $r = 92$ ,  $n_{TO} = 0$  y  $w = \sigma = 7$  (ver la explicación de la fase FR).  $A_{CA.aux} = 0$ , de manera que las rondas en la CA coinciden con los RTTs. El LE se presenta con la pérdida del segmento 94, que es el segundo segmento enviado en la CA ( $\Delta = 2$ ), lo que implica que  $i = 0$  y  $\beta = 1$ , de acuerdo con (4.40) y (4.41), respectivamente. De la figura es claro que  $a_{CA} = 1$  y  $m_{CA} = 8$ , como lo establecen (4.43) y (4.44). La LW abarca los segmentos [94 - 100], es decir, su tamaño está dado como  $w = 7$ , según (4.38). Los segmentos 94 al 99 se pierden (i.e.  $\delta_{CA} = 6$ ), por lo que, según (4.46), sólo hay  $d = 1$  DA. Las variables  $\delta$ ,  $X$  y  $Y$  se actualizan con (4.47), (4.48) y (4.49).  $a = 93$  y  $s = 100$  como lo determinan (4.50) y (4.45) respectivamente. Se cumple la condición  $a > r$ , lo que permite la actualización de  $m = s = 100$  y  $n_{TO} = 0$ . Las variables  $A_{CA.aux}$ ,  $N$ ,  $r$  y  $\sigma$  no se modifican al pasar por la fase CA. La cantidad de DAs no alcanza para producir un TD, por lo que la CA termina en un TO. El último RA, correspondiente al segmento 93, se recibe en el mismo RTT en el que el LE se presenta, lo que se refleja en que  $\beta > A_{CA.aux}$ , por lo que, de acuerdo con (4.66), la duración de la fase es de  $i + 1 + RTO$  rondas.

Un ejemplo de una transición FR  $\delta=0$  - CA  $a=r$  TO se presenta en la figura 4.12. La segunda CA prosigue de una fase FR sin transmisiones nuevas, de forma que  $m = s = a = r$ ,  $A_{CA.aux} = 0$ ,  $\delta = 0$  y  $n_{TO} = 0$  al salir de la FR (ver la explicación de la fase FR). El LE se presenta por la pérdida del primer segmento transmitido en la CA, por lo que se mantiene  $a = r$ .<sup>13</sup> La LW abarca los segmentos [93 - 99] y de ellos sólo se pierde el 93, por lo que

<sup>13</sup>La única forma en que se puede producir una transición FR  $\delta=0$  - CA  $a=r$  TO es porque no haya transmisiones nuevas en la FR y se pierda el primer segmento transmitido en la CA.

Variable\Fase	SS	FR	CA	FR	CA	TO	TO
$a$	29	68	✓	92	92	98	✓
$A$	+6	+3	+0	+4	+RTO	+RTO	+0
$A_{CA\_aux}$	-	11	-	0	-	-	-
$d$	27	0	10	0	6	-	-
$\delta$	3	3	4	0	1	-	0
$m$	59	81	92	92	99	-	-
$n_{TO}$	0	0	0	0	✓	1	2
$N $	-	{12, 10, 11, 0}	-	{0,0,0,7}	-	-	-
$r$	0	59	-	92	-	99	✓
$s$	59	81	92	92	99	98	✓
$\sigma$	500	15	-	7	-	3	2
$w$	30	15	✓	7	7	1	1
$X$	+56	+22	+10	+4	+6	+0	+0
$Y$	+59	+25	+11	+4	+7	+11	+0
$i$	4	2	0	3	0	-	-
$\beta$	14	9	-	-	0	-	-

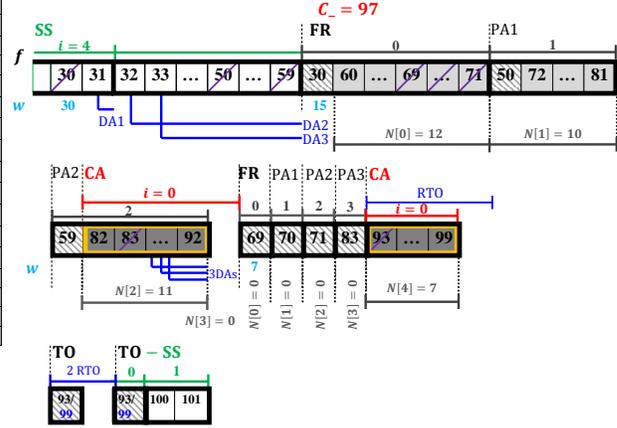


Figura 4.12: Fragmento de una conexión TCP NewReno. Caso E.

hay  $d = 6$  DAs. Aunque se presenta un TD con los ACKs de los segmentos 94, 95 y 96, la condición  $a = r$  obliga a que este nuevo LE se recupere forzosamente después de un TO. No hay RAs en esta fase, así que su duración es de RTO rondas.

$FR \xrightarrow{\delta > 0} CA - TO$ .

Si  $\delta > 0$  al inicio de la CA, entonces la CA sucede a una fase cFR con algunas transmisiones nuevas perdidas. En esta situación, la actualización de las variables se lleva a cabo de la misma forma que cuando una transición  $FR \xrightarrow{\delta > 0} CA - FR$  ocurre, excepto en lo referente a su duración. Es decir, las variables  $w$ ,  $d$ ,  $\delta$ ,  $X$ ,  $Y$ ,  $a$  y  $m$  se actualizan según (4.38), (4.46), (4.47), (4.48), (4.49), (4.50) y (4.51), respectivamente, a partir de los valores  $i = 0$  y  $a_{CA}$  y  $m_{CA}$  de acuerdo con (4.62) y (4.63), respectivamente. En este caso,  $s$  se actualiza de acuerdo con (4.64). Las variables  $A_{CA\_aux}$ ,  $N$ ,  $r$  y  $\sigma$  no se modifican al pasar por la fase CA.

El factor  $n_{TO}$  se restablece cada que se cumple que  $a > r$ . Como esta condición es necesaria para activar una FR, entonces  $n_{TO} = 0$  siempre que una FR comienza. El factor  $n_{TO}$  se modifica durante la fase TO, por lo que después de una FR,  $n_{TO} = 0$  se mantiene hasta que se presenta el siguiente TO. Debido a esto, en una transición  $FR \xrightarrow{\delta > 0} CA - TO$  siempre se cumple que  $n_{TO} = 0$  al finalizar la CA posterior a la FR. De cualquier forma, este valor de  $n_{TO}$  corresponde con (4.20). Además, en este caso, ningún segmento transmitido en la CA servirá para restablecer el temporizador puesto que los segmentos que no se pierdan serán reconocidos mediante DAs. Así que el último RA recibido es el RA total con el que la FR anterior termina. Por lo tanto, la duración de la CA es solamente de RTO rondas y  $A$  se actualiza al

final de la CA como:

$$A = A + \text{RTO} \quad (4.71)$$

Se puede desglosar la condición para que se presente un TO en dos casos:

- $FR \overset{\delta > 0}{}$  -  $CA \overset{a=r}{}$   $TO$ . Como se menciona en la explicación de la fase FR, siempre al salir de una FR se cumple que  $a \geq r$ . Por lo tanto, la condición  $a \leq r$ , suficiente para que se produzca un TO se limita a la condición  $a = r$ . La única forma en que se produce esta transición es porque se perdió la primera transmisión nueva en la cFR.
- $FR \overset{\delta > 0}{}$   $CA \overset{a > r \text{ AND } d < 3}{}$   $TO$ : En una CA que prosigue a una cFR con transmisiones nuevas perdidas ( $\delta > 0$ ), todos los segmentos enviados en la CA serán reconocidos mediante DAs. Si hay más de tres segmentos sin pérdida en la CA, entonces se produce un TD. Por ello, la transición  $FR \overset{\delta > 0}{}$   $CA \overset{a > r \text{ AND } d < 3}{}$   $TO$  se produce cuando se pierde al menos una transmisión nueva dentro de la cFR (excepto la primera) y hay menos de tres transmisiones exitosas en la CA siguiente.

Como ejemplo de una transición  $FR \overset{\delta > 0}{}$   $CA \overset{a > r \text{ AND } d < 3}{}$   $TO$ , considérese el fragmento de una conexión TCP NewReno mostrado en la figura 4.3. La primera CA prosigue de una fase FR que concluye con  $\delta = 1$  transmisión nueva perdida. La FR finaliza con  $m = s = 84$ ,  $a = 83$ ,  $r = 59$ ,  $A_{CA.aux} = N[f_{FR} - 1] = 13$ ,  $N[f_{FR}] = 1$ ,  $n_{TO} = 0$  y  $w = \sigma = 15$  (ver la explicación de la fase FR). La suma de los  $N[f_{FR} - 1] + N[f_{FR}]$  segmentos constituyen los  $m_{CA}$  segmentos que se consideran enviados durante la única ronda  $i = 0$  que dura la CA, como se establece en (4.63). De éstos, ninguno se presenta antes del LE, pues éste comenzó desde la FR anterior, por lo que  $a_{CA} = 0$ , como lo indica (4.62) y  $a$  no se actualiza, de acuerdo con (4.50). En este caso, todos los  $m_{CA}$  segmentos enviados en la CA se pierden, de forma que  $\delta_{CA} = 14$  y, según (4.46) y (4.47), se actualizan las variables  $d = 0$  y  $\delta = 15$ .  $w$  no se modifica al pasar por esta fase y las variables  $X$  y  $Y$  se actualizan según (4.48) y (4.49), respectivamente. El último segmento transmitido,  $s$ , se obtiene mediante (4.64), con el cual se actualiza  $m$ . En este caso, se cumple la condición  $a > r$ , sin embargo, puesto que  $d = 0$ , no se produce un TD. La CA termina con un TO una RTO después de iniciarse.

Otro ejemplo de una transición  $FR \overset{\delta > 0}{}$   $CA \overset{a > r \text{ AND } d < 3}{}$   $TO$  se presenta en la figura 4.7. La cuarta CA sigue de una fase FR que concluye con  $\delta = 1$  transmisión nueva perdida, y con los valores  $m = s = 83$ ,  $a = 81$ ,  $r = 80$ ,  $A_{CA.aux} = N[f_{FR} - 1] = 2$ ,  $N[f_{FR}] = 0$ ,  $n_{TO} = 0$  y  $w = \sigma = 4$  (ver la explicación de la fase FR). El LE presentado en la FR, a partir de la pérdida del segmento 82, determina el valor de  $a = 81$ , el cual no se altera al pasar por la fase CA siguiente.

Variable\Fase	SS <sub>ini</sub>	FR	CA
$a$	29	59	✓
$A$	+6	+3	+RTO
$A_{CA\ aux}$	-	12	-
$d$	27	0	12
$\delta$	3	2	2
$m$	59	82	94
$n_{TO}$	0	✓	✓
$N[\ ]$	-	12, 11, 12, 0	-
$r$	0	59	-
$s$	59	82	94
$\sigma$	500	15	-
$w$	30	15	14
$X$	+56	+24	+12
$Y$	+56	+26	+12
$i$	4	2	0
$\beta$	14	0	-

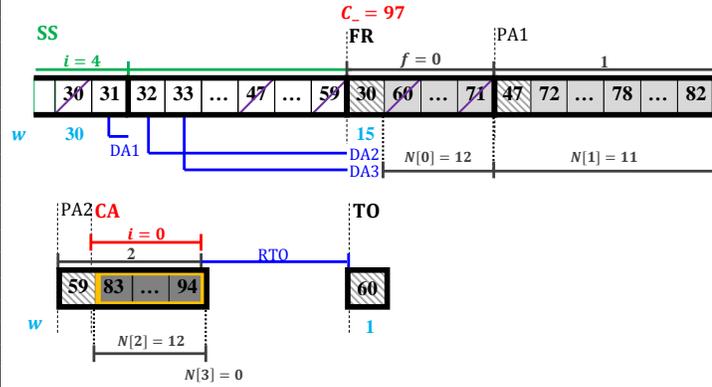


Figura 4.13: Fragmento de una conexión TCP NewReno. Caso F.

Ya que esta pérdida no corresponde a la primera nueva transmisión en la FR, se cumple con que  $a > r$  al salir de la FR y esta condición se mantiene en la CA siguiente. En la CA se transmiten los segmentos 84 y 85 sin pérdida. Sin embargo, sus DAs no son suficientes para generar el TD. Por lo tanto, la CA termina con un TO una RTO después de que la FR termina.

Como ejemplo de una transición FR  $\xrightarrow{\delta > 0}$  CA  $\xrightarrow{a=r}$  TO considérese el fragmento de una conexión TCP NewReno mostrado en la figura 4.13. La CA prosigue de una fase FR que concluye con  $\delta = 2$  transmisiones nuevas perdidas, y con los valores  $m = s = 82$ ,  $a = 59$ ,  $r = 59$ ,  $A_{CA\ aux} = N[f_{FR} - 1] = 12$ ,  $N[f_{FR}] = 0$ ,  $n_{TO} = 0$  y  $w = \sigma = 15$  (ver la explicación de la fase FR). La pérdida del primer segmento nuevo transmitido en la FR provoca que  $a = r$  al salir de la FR. Los valores de ambas variables ( $a$  y  $r$ ) se mantienen durante una CA que sigue de una cFR con pérdidas. Esta condición obliga a la CA a terminar con un TO independientemente de la cantidad de DAs recibidos, que en este caso son  $d = 12$ , correspondientes a los  $m_{CA}$  segmentos transmitidos en la CA. El TO se presenta RTO rondas después del final de la FR.

CA - EC.

Esta transición sólo ocurre después de una transición SS/FR  $\xrightarrow{\delta=0}$  CA. Considerando al valor de la variable  $a$  al inicio de la fase, esta transición se genera cuando se tienen  $(C_- - a)$  segmentos transmitidos exitosamente (i.e. se mantiene  $\delta = 0$ ) en la CA, es decir, cuando:

$$a_{CA} = m_{CA} = C_- - a \quad (4.72)$$

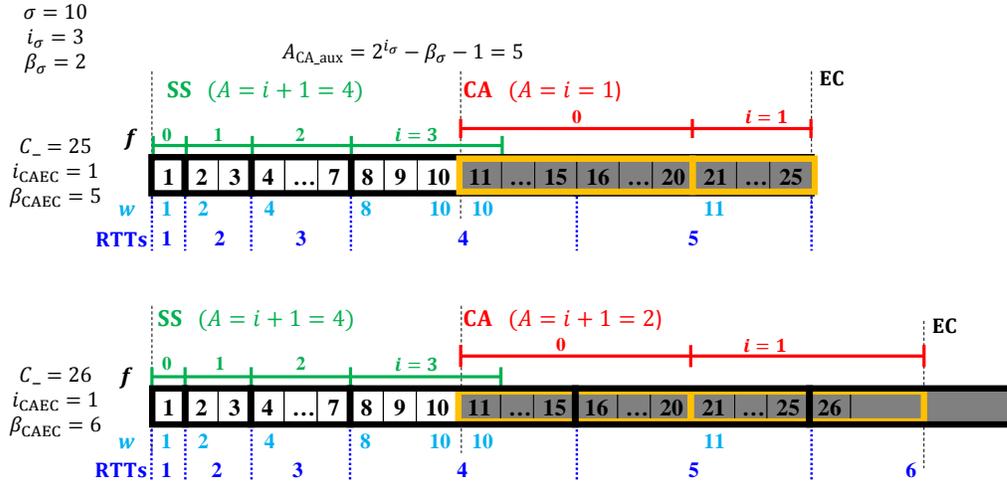


Figura 4.14: SS - CA - EC.

El último segmento del flujo ocurre en  $i$  y  $\beta$  dadas por (4.54) y (4.55),<sup>14</sup> respectivamente. Usando estos valores, las variables  $w$ ,  $s$ ,  $d$ ,  $X$ ,  $Y$ ,  $a$  y  $m$  se actualizan al finalizar la CA según (4.38), (4.45), (4.46), (4.48), (4.49), (4.50) y (4.51), respectivamente.

La fase concluye con la recepción del RA del último segmento del flujo, es decir, del segmento con  $SN = C_-$ . La cantidad de RTTs que transcurren durante la CA depende, sin embargo, de la posición relativa del segmento  $C_-$  en la  $cwnd$  enviada en la ronda  $i = i_{CAEC}$ ,  $\beta = \beta_{CAEC}$ , respecto del desfase que exista entre las rondas de la CA y los RTTs, reflejado en la variable  $A_{CA\_aux}$ . Considérense los casos de la figura 4.14. En ambos casos, son  $A_{CA\_aux} = 5$  los segmentos que se consideran dentro de la primera ronda de la CA pero que en realidad se enviaron en la última ronda del SS (los segmentos del 11 al 15). En la  $cwnd$  correspondiente al inicio de cada ronda de la CA, por tanto, los primeros  $A_{CA\_aux}$  segmentos se envían un RTT antes que los siguientes segmentos. Por lo tanto, si  $\beta_{CAEC} \leq A_{CA\_aux}$ , entonces la CA dura  $i = i_{CAEC}$  RTTs. En el caso contrario, la CA dura  $i + 1$  RTTs.

Como ejemplo de una transición CA - EC, considérese la figura 4.7. En ella,  $C_- = 97$ . La última CA prosigue de un SS, el cual termina con  $m = s = a = 86$ . La CA comienza su transmisión con el segmento posterior al último segmento reconocido antes de entrar a su fase, cuyo SN es igual a  $a$ . La fase concluye con la transmisión del flujo al transmitir  $C_- - a$  segmentos sin pérdidas, lo que corresponde con (4.72). La CA termina en su ronda  $i = 3$ , de acuerdo con (4.54). La posición del segmento  $C_-$  en la  $cwnd$  enviada al inicio de esta ronda

<sup>14</sup>Es preciso recordar que  $\beta_{CAEC} \in [1, \sigma + i_{CAEC}]$ .

es  $\beta_{CAEC} = 2$ , según (4.55).<sup>15</sup> Como  $\beta_{CAEC} > A_{CA.aux}$ , entonces la CA dura  $i + 1$  RTTs.

Combinando las posibles transiciones que pueden ocurrir desde una CA, la variable  $A$  se actualiza al pasar por esta fase como:

$$A = A + \left\{ \begin{array}{ll} i & ; (FR^{\delta \geq 0} CA - FR \text{ AND } N[f_{FR}] = 0) \\ & \text{OR } (CA - EC \text{ AND } \beta_{CAEC} \leq A_{CA.aux}) \\ i + 1 & ; (SS/FR^{\delta = 0} CA - FR) \\ & \text{OR } (FR^{\delta \geq 0} CA - FR \text{ AND } N[f_{FR}] > 0) \\ & \text{OR } (CA - EC \text{ AND } \beta_{CAEC} > A_{CA.aux}) \\ i + \text{mín}(RTO_{max}, 2^{n_{TO}} RTO) & ; (FR^{\delta \geq 0} CA - TO) \\ & \text{OR } (SS/FR^{\delta = 0} CA - TO \text{ AND } \beta \leq A_{CA.aux}) \\ i + 1 + \text{mín}(RTO_{max}, 2^{n_{TO}} RTO) & ; SS/FR^{\delta = 0} CA - TO \text{ AND } \beta > A_{CA.aux} \end{array} \right. \quad (4.73)$$

### 4.2.3. Fase FR

En [15] se especifica que cuando un receptor TCP recibe un segmento fuera de orden debe emitir inmediatamente un DA, y cuando recibe un segmento que completa parcial o totalmente un espacio en la secuencia de segmentos recibidos debe responder con un RA. Si un transmisor TCP recibe un TD,<sup>16</sup> lo interpreta como una indicación de un LE, el cual comienza con la pérdida del segmento indicado por los DAs. Si se cumple que el máximo SN reconocido abarca más que el *recover* [61], es decir, si  $a > r$ ,<sup>17</sup> entonces las  $\delta$  pérdidas del LE se intentarán recuperar con la FR de NewReno.

Así, el algoritmo transita a la fase FR desde una fase SS o CA que termina con un LE, es decir, que concluye con la emisión de una LW, con  $\delta$  pérdidas y  $w - \delta$  segmentos no perdidos (cada uno de los cuales provocará la emisión de un DA en el receptor). Como un requisito

<sup>15</sup>Es preciso recordar que  $\beta_{CAEC} \in [1, \sigma + i_{CAEC}]$ .

<sup>16</sup>En [15] existe la sugerencia de enviar un segmento al recibir el primer y el segundo DA. Éstos segmentos pueden servir para completar los tres DAs, si es que no se tenían. Por el momento, esta sugerencia no se incluye en el análisis.

<sup>17</sup>Ver nota al pie 5.

para activar la fase FR se debe cumplir que  $a > r$ , lo que implica que siempre  $n_{TO} = 0$  al entrar a una FR. Como el factor  $n_{TO}$  se actualiza solamente en la fase TO (ver la explicación de la fase TO), este valor se conserva durante la FR y  $n_{TO} = 0$  al salir de la FR.

Tabla 4.5 Variables locales de la fase FR.

Variable	Descripción
$\beta_\delta$	Es la posición de la primera transmisión nueva perdida en una cFR dentro de la $cwnd$ enviada al inicio de la ronda $f_\delta, \beta_\delta \in [0, N[f_\delta] - 1]$ .
$\delta_{FR}[]$	Arreglo que contiene la cantidad de transmisiones nuevas perdidas en cada ronda de la FR, $f = 0, \dots, f_{FR} - 1$
$\delta_{FRCA}$	Es el total de transmisiones nuevas perdidas en la cFR antes de su transición a la CA.
$\delta_{FRTO}$	Es el total de transmisiones nuevas perdidas en la iFR antes de su transición al TO.
$f_\delta$	Es la ronda en la que ocurre la primera transmisión nueva perdida en una cFR. $f_\delta \in [0, \delta - 2]$ , con $\delta$ al inicio de la fase.
$f_{TO}$	Es la ronda en la que ocurre la primera retransmisión perdida en una iFR. $f_{TO} \in [0, \delta - 1]$ , con $\delta$ al inicio de la fase.
$N_{FR}$	Es la suma del total de transmisiones nuevas en la FR.

Se define un conjunto de variables locales en la fase FR como se describe en la tabla 4.5. El mecanismo FR representado en el algoritmo consta de los siguientes pasos:

FR.1. Según [15], se retransmite el segmento supuesto como perdido y se actualizan las variables:<sup>18</sup>

$$r = m \quad (4.74)$$

$$\sigma = \text{máx}(2, \lfloor \frac{w}{2} \rfloor) \quad (4.75)$$

$$w = \sigma \quad (4.76)$$

FR.2.  $w$  se aumenta temporalmente en un SMSS por cada DA recibido (incluyendo el TD que activó la FR). Al entrar a la FR y como resultado de los pasos FR.1 y FR.2,  $w$

<sup>18</sup>En [15] se especifica que la actualización de  $\sigma$  debe ser como  $\sigma = \text{máx}(FlightSize/2, 2 * SMSS)$ . Sin embargo, en el modelo se usa el valor de  $w$  en lugar del  $FlightSize$ , por la dificultad de saber este valor. Bajo las suposiciones en las que se basa el modelo, particularmente: transmisión continua, transmisión en ráfaga en cada ronda y  $rwnd \gg cwnd$ , se considera que esta aproximación es aceptable. Otra aproximación podría ser  $\sigma = \text{máx}(2, \lfloor \frac{m-a}{2} \rfloor)$ , con los valores de  $a$  y  $m$  al inicio de la FR; sin embargo, es posible que esta aproximación también sobrestime el valor del  $FlightSize$ .

se actualiza como  $w = \sigma + d$ . Si hay más datos por transmitir (que es el caso en el algoritmo, por la suposición de transmisión continua) y si la  $cwnd$  lo permite, se transmiten [15], comenzando por el segmento con  $SN = m+1$ , es decir, con el segmento siguiente al último (y también más grande) SN enviado, registrado en  $m = s$ . Para calcular la cantidad de segmentos susceptibles de transmitirse en la primera ronda de la FR,  $f = 0$ , se toma el nuevo valor de  $w$  y se le restan los  $m - a$  segmentos de la LW (i.e. los segmentos que ya se han enviado pero que aún no se reconocen mediante un RA). Por lo tanto, suponiendo que  $C_- \gg m$ , la cantidad de segmentos susceptibles de transmitirse en la primera ronda de la FR queda dada por  $\sigma + d - (m - a)$ . Si este resultado es negativo, significa que hay más transmisiones en curso de las que la  $cwnd$  actualizada lo permite. Por lo tanto:

$$N[f = 0] = \text{máx}(0, \sigma + d - (m - a)) \quad (4.77)$$

En la figura 4.8 se muestran dos instancias de la fase FR. La primera sucede a una fase SS, en la que el LE se debe a la pérdida del segmento 30 y la LW abarca los segmentos [30 - 59]. En este caso, el tamaño de la LW corresponde con el valor de  $w = 30$  al salir de la fase SS (ya que  $C_-$  está muy alejado del SN con el que el SS termina). Además, se cumple que  $w = m - a = 59 - 29$ . El decir, las únicas transmisiones en curso corresponden a la última  $cwnd$  en el SS. Como resultado del paso [FR.1.], al entrar a la FR se actualizan  $r = 59$ ,  $\sigma = 15$  y  $w = 15$  de acuerdo con (4.74), (4.75) y (4.76), respectivamente. Sin embargo, debido a [FR.2.], al inicio de la ronda  $f = 0$  de la FR,  $w$  se aumenta temporalmente como  $w = \sigma + d = 15 + 27$  y la  $cwnd$  abarca los segmentos desde el 30 al 71. De éstos, los segmentos [30 - 59] corresponden a la LW, por lo que ya estaban en curso desde la fase anterior a la FR. Así que, después de la retransmisión del segmento 30, en esta ronda se transmiten  $N[f = 0] = 12$  segmentos, como lo establece (4.77). La FR transita a una CA cuando concluye con la recuperación del LE anterior. Sin embargo, un nuevo LE ocurrió dentro de dicha FR, con la pérdida del segmento 69, abarcando todos los segmentos transmitidos en la FR y en la CA siguiente, es decir, hasta el segmento 92. En consecuencia, la LW tiene un tamaño de  $m - a = 92 - 68 = 24$ , que supera al valor de  $w = 15$  con el que la CA termina. Las transmisiones exitosas dentro de la CA se contabilizan como  $d = 10$  DAs. Se activa una nueva FR para recuperar este LE. Como resultado del paso [FR.1.], al entrar a la segunda FR se actualizan  $r = 92$ ,  $\sigma = 7$  y  $w = 7$  de acuerdo con (4.74), (4.75) y (4.76), respectivamente. Sin embargo, debido a [FR.2.], al inicio de la ronda  $f = 0$  de la segunda FR,  $w$  crece temporalmente como  $w = \sigma + d = 7 + 10$  y la  $cwnd$  abarca los segmentos desde el 69 al

85. Esto implica que hay más segmentos en transmisión (hasta el segmento  $m = 92$ ) de los que la *cwnd* permite transmitir. En consecuencia, después de la retransmisión del segmento 69 no hay transmisiones nuevas en la ronda  $f = 0$  de la FR y  $N[0] = 0$  según (4.77), puesto que  $\sigma + d - (m - a) = -14 < 0$ .

En la figura 4.7 se puede hallar un caso en el que la *cwnd* en la primera ronda de la FR, después de su disminución por (4.76) y su crecimiento temporal indicado en el paso [FR.2.], queda justo del mismo tamaño que la LW, por lo que  $N[0] = 0$ . La segunda FR en esta figura sigue de una CA que concluye por la pérdida del segmento 37, de manera que la LW abarca los segmentos [37 - 42], es decir,  $m = 42$ ,  $a = 36$  y  $w = 6$ . Los segmentos 38, 40 y 42 de la LW no se pierden, por lo que  $d = 3$ . Como resultado de los pasos [FR.1.] y [FR.2.], al inicio de la ronda  $f = 0$  de la FR,  $w$  queda temporalmente como  $w = \sigma + d = 3 + 3$ . De forma que todos los segmentos dentro de la *cwnd* al inicio de la FR ya están en curso y  $N[0] = 0$  según (4.77), puesto que  $\sigma + d - (m - a) = 0$ .

Adicionalmente, en la figura 4.7 también se presenta un caso en el que la *cwnd* en la primera ronda de la FR, después de su disminución por (4.76) y su crecimiento temporal indicado en el paso [FR.2.], queda de un tamaño menor que el tamaño de la LW, por lo que  $N[0] = 0$ . La tercera FR en esta figura sigue de una CA que concluye por la pérdida del segmento 73. La LW abarca los segmentos [73 - 80], de los cuales los segmentos 74, 75 y 76 no se pierden, por lo que,  $m = 80$ ,  $a = 72$ ,  $w = 8$  y  $d = 3$  al entrar a la tercera FR. Como resultado de los pasos [FR.1.] y [FR.2.], al inicio de la ronda  $f = 0$  de la FR,  $w$  queda temporalmente como  $w = 7$  y la *cwnd* abarca los segmentos [73-79], por lo que  $N[0] = 0$  según (4.77), puesto que  $\sigma + d - (m - a) = -1$ .

De las de las  $N[f = 0]$  transmisiones nuevas en la ronda  $f = 0$  de la FR, los segmentos perdidos se contabilizan en  $\delta_{FR}[f]$  y los que lleguen exitosamente provocarán la emisión de DAs, los cuales servirán para aumentar  $w$  en la siguiente ronda de la FR.

FR.3. En cada ronda subsecuente dentro de la FR,  $f = 1, \dots, f_{FR} - 1$ , si se recibió un PA al finalizar la ronda anterior:

FR.3.1. Se retransmite el primer segmento no reconocido aún.

FR.3.2. La *cwnd* se incrementa en un SMSS (lo que refleja la retransmisión exitosa).

FR.3.3. La *cwnd* se disminuye por la cantidad de datos nuevos reconocidos, es decir, desde el segmento que al retransmitirse generó el PA actual, hasta antes del siguiente segmento sin reconocer.

FR.3.4. La *cwnd* se desplaza hacia la derecha para comenzar con el siguiente segmento sin reconocer [61].

Como resultado de los pasos [FR.2.] y [FR.3.], es posible comprender el efecto global en la *cwnd* como sigue: una vez que  $w$  crece por en uno por cada DA ([FR.2.]) y por el PA ([FR.3.2.]), el límite derecho de la *cwnd* no se altera por efecto de los pasos [FR.3.3.] y [FR.3.4.], ya que el recorte que  $w$  sufre debido a [FR.3.3.] equivale al desplazamiento que se indica en [FR.3.4.]. El análisis de la cantidad de transmisiones nuevas que se permiten en cada ronda de una cFR se muestra en la figura 4.15. En esta figura, un LE que abarca los segmentos  $[a + 1 - m]$  es indicado por un TD, por lo que una FR se activa. En la ronda  $f = 0$  de la FR, se retransmite el segmento que generó el LE, Rtx1, seguido de  $N[0]$  segmentos nuevos, según (4.77). De estos segmentos, las pérdidas se cuentan en  $\delta_{FR}[0]$ . Al finalizar la ronda  $f = 0$ , se reciben el PA1, correspondiente a la Rtx1 exitosa, y  $N[0] - \delta_{FR}[0]$  DAs. Todos ellos provocan un crecimiento en  $w$  de un SMSS cada uno. Consecuentemente, la *cwnd* otorga el permiso de transmisión a  $N[1] = N[0] - \delta_{FR}[0] + 1$  segmentos nuevos al iniciar la ronda  $f = 1$ . Mientras no se pierda ninguna retransmisión, este proceso se repite en cada ronda hasta la  $f = \delta - 1$ . En esta ronda se envía la retransmisión correspondiente a la última pérdida del LE seguida de los  $N[\delta - 1]$  segmentos nuevos. Generalizando este análisis, la cantidad de segmentos nuevos que se transmiten en cada ronda de la cFR está dada por  $N[f] = \sigma + d - m + a + f - \sum_{g=0}^{f-1} \delta_{FR}[g]$ ,  $f = 0, \dots, \delta - 1$ . Si  $N[f]$  calculado de esta forma resulta negativo, significa, como se vio en el caso de  $N[0]$ , que hay más segmentos en transmisión de los que la *cwnd* permite, por lo que es necesario complementar este cálculo con un  $\text{máx}(0, N[f])$ , como se hizo para  $f = 0$  en (4.77), es decir:

$$N[f] = \text{máx}(0, \sigma + d - m + a + f - \sum_{g=0}^{f-1} \delta_{FR}[g]); f = 0, \dots, \delta - 1 \quad (4.78)$$

Si el final del flujo está cercano, entonces esta cantidad puede verse limitada de la siguiente forma:

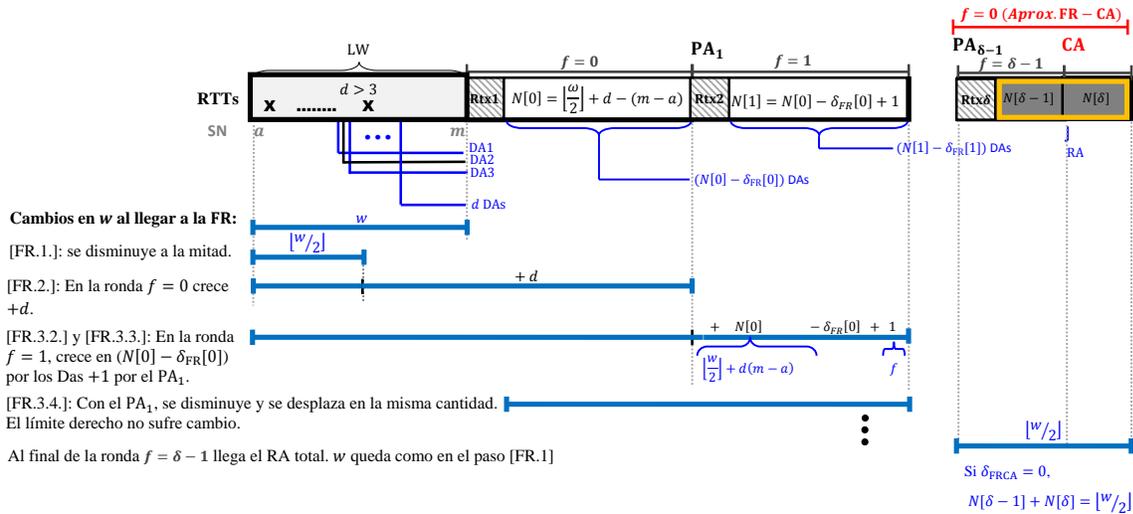


Figura 4.15: Nuevas transmisiones en una cFR.

$$N[f] = \min(C_- - m - \sum_{g=0}^{f-1} N[g], \max(0, \sigma + d - m + a + f - \sum_{g=0}^{f-1} \delta_{FR}[g])); f = 0, \dots, \delta - 1 \quad (4.79)$$

Continuando con los ejemplos de FR en la figura 4.8, de los  $N[0] = 12$  segmentos nuevos transmitidos en la ronda  $f = 0$  de la primera FR, los segmentos [69 - 71] se pierden, por lo que  $\delta_{FR}[0] = 3$ . Esta primera ronda concluye con la recepción del PA generado por la retransmisión exitosa del segmento 30 y con  $N[0] - \delta_{FR}[0] = 9$  DAs. El SN indicado por el PA corresponde al segmento 49, previo a la segunda pérdida del LE. Los pasos [FR.2.] y [FR.3.2.] resultan en un crecimiento de  $w$  en 10 SMSSs, de manera que la  $cwnd$  se abre para cubrir los segmentos [72 - 81]. En consecuencia y debido al paso [FR.3.3.], la  $cwnd$  se disminuye en 20, pues con este PA se reconocen los segmentos [30 - 49]. Por lo tanto, al inicio de la ronda  $f = 1$ ,  $w = w + 9 + 1 - 20 = 32$ . Como resultado del paso [FR.3.4.], también la  $cwnd$  se desplaza para iniciar con la segunda pérdida del LE, el segmento 50. Es decir, la  $cwnd$  queda abarcando los segmentos del 50 al 81. Es decir, los pasos [FR.3.3.] y paso [FR.3.4.] sólo desplazan el límite izquierdo de la  $cwnd$ , pero no alteran su límite derecho resultado de los pasos [FR.2.] y paso [FR.3.2.]. Al inicio de la ronda  $f = 1$  se retransmite el segmento 50 seguido de los los segmentos del 72 al 81, es decir,  $N[1] = 10$ , como lo prevé (4.79). No hay ninguna pérdida en la transmisión de la ronda  $f = 1$  (i.e.  $\delta_{FR}[0] = 0$ ), por lo que al final de la misma se recibe el PA correspondiente a la retransmisión del segmento 50 y  $N[1]$  DAs.

En consecuencia, al inicio de la ronda  $f = 2$  se envían  $N[2] = 11$  segmentos nuevos, lo que coincide con lo establecido por (4.79).

La segunda FR de la figura 4.8 no tiene transmisiones nuevas en la ronda  $f = 0$  puesto el LE abarca hasta el segmento 92 y en la primera ronda de la FR, la  $cwnd$  abarca sólo hasta el segmento 85. Esto se refleja en el resultado negativo de  $\sigma + d - (m - a) = -14 < 0$ . El paso [FR.2.] no surte efecto en este caso, pues no hay DAs. Por causa del paso [FR.3.2], la  $w$  se aumenta en 1 para abarcar hasta el segmento 86. Como se mencionó anteriormente, los pasos [FR.3.3.] y [FR.3.4.] solamente afectan al límite izquierdo de la  $cwnd$ , el cual se desplaza para comenzar con el segmento 70. Al inicio de la ronda  $f = 1$ , de nuevo  $\sigma + d - (m - a) + f = -13 < 0$ , por lo que  $N[1] = 0$ . Este proceso se repite en las  $\delta = 4$  rondas que dura la cFR, de forma que al inicio de la ronda  $f = \delta - 1 = 3$ , la  $cwnd$  se ha abierto para abarcar hasta el segmento 88. Es decir, en este caso, incluso en la última ronda de la FR siguen habiendo más segmentos en tránsito que los que la  $cwnd$  permite. Por lo tanto, este es un caso en el que no hubo transmisiones nuevas en ninguna ronda de la FR inclusive en la ronda  $f = \delta - 1$ .

*cFR: Transición FR - CA.*

FR.4. Si la retransmisión de las  $\delta$  pérdidas del último LE es exitosa (i.e. se trata de una cFR), la última retransmisión es reconocida mediante un RA total, es decir, un RA con  $a \geq r$ , el cual se recibe hacia el final de la ronda  $f = \delta - 1$  provocando la transición FR - CA. Una vez en la CA, se concluye con la transmisión en ráfaga de la  $cwnd$ . Por lo tanto, la cFR dura un número de rondas igual a:

$$f_{FR} = \delta \quad (4.80)$$

y la duración de la conexión al salir de la cFR se actualiza como:

$$A = A + \delta \quad (4.81)$$

*Aproximación FR-CA.* En el momento de la transición, los  $N[\delta - 1]$  segmentos transmitidos en la última ronda de la FR se convierten segmentos dentro de la  $cwnd$  en la primera ronda de la CA. Al inicio de la CA se termina la transmisión de la  $cwnd$  con los  $N[\delta]$  segmentos faltantes. Así, aunque los  $N[\delta - 1]$  segmentos se transmiten en la

última ronda de la FR y los  $N[\delta]$  se transmiten al iniciarse la CA, todos estos segmentos se contabilizan dentro de la fase CA. La cantidad de segmentos transmitidos en la fase previa a la CA pero considerados como parte de la CA está dada por:

$$A_{CA\_aux} = N[\delta - 1] \quad (4.82)$$

Por lo tanto, la cantidad de transmisiones nuevas realizadas en la cFR se cuantifican con  $N_{FR}$  definida como:

$$N_{FR} = \sum_{g=0}^{\delta-2} N[g] \quad (4.83)$$

Y la cantidad de segmentos transmitidos en la primer ronda de la CA siguiente se calcula como  $N[\delta - 1] + N[\delta]$ , como se especifica en (4.63) con  $f_{FR}$  dada por (4.80), y se muestra en la figura 4.15. La pérdida de alguno de estos segmentos se tratará dentro de la CA siguiente. La cantidad de transmisiones nuevas perdidas durante la cFR se contabiliza en:

$$\delta_{FRCA} = \sum_{g=0}^{\delta-2} \delta_{FR}[g] \quad (4.84)$$

La cantidad de segmentos enviados sin pérdida en una cFR incluye tanto a las  $\delta$  retransmisiones exitosas como a todas las transmisiones nuevas sin pérdida que se consideran dentro de la FR ( $N_{FR} - \delta_{FRCA}$ ), de manera que  $X$  se actualiza al final de la cFR como:

$$X = X + \delta + N_{FR} - \delta_{FRCA} \quad (4.85)$$

Y la cantidad de segmentos enviados en total se actualiza al paso por una cFR como:

$$Y = Y + N_{FR} + \delta \quad (4.86)$$

La variable  $\delta$  se actualiza al salir de la cFR con el número de transmisiones nuevas perdidas en esta fase, es decir:

$$\delta = \delta_{FRCA} \quad (4.87)$$

Puesto que la cFR termina con un RA, al salir de esta fase no hay ningún DA, es decir:

$$d = 0 \quad (4.88)$$

Como se ha mencionado con anterioridad, la condición  $a > r$ , necesaria para activar una fase FR, implica que todos los LEs previos se han recuperado por completo, de forma que no existen retransmisiones pendientes. Esta condición también es suficiente para restablecer el mecanismo de *back-off* y asegura que el máximo SN enviado corresponde con el SN del último segmento transmitido. De manera que siempre, al entrar a una fase FR, se cumple que  $n_{TO} = 0$  y  $m = s$ . El factor  $n_{TO}$  se incrementa únicamente en la fase TO, por lo que al salir de la FR se mantiene con el valor  $n_{TO} = 0$ . Puesto que la transmisión de segmentos nuevos en la FR comienza a partir del último (y más grande) SN transmitido, registrado en el valor de  $m = s$  al entrar a la fase, el SN del último segmento transmitido en la FR corresponde a:

$$s = m + N_{FR} \quad (4.89)$$

Según [61], la recepción del RA total provoca el decremento de  $w$  al valor establecido por (4.76), es decir,  $w = \sigma$ .<sup>19</sup> Si no hubo pérdida de transmisiones nuevas en la FR, es decir, si  $\delta_{FRCA} = 0$ , entonces el SN del último segmento transmitido también corresponde al máximo SN reconocido, y la variable  $a$  se actualiza como:

$$a = s \quad (4.90)$$

Además, si no hay pérdidas en la cFR, entonces en el instante en que se produce la transición FR - CA, los  $N[\delta - 1]$  segmentos se convierten en los primeros segmentos de la *cwnd*. Al inicio de la CA se transmiten los  $N[\delta]$  segmentos con los que se completa la *cwnd*, que equivalen a  $N[\delta] = \sigma - N[\delta - 1]$ . Sin embargo, si el final del flujo está cercano, esta cantidad puede verse limitada como:

$$N[\delta] = \text{mín}(C_- - m - \sum_{g=0}^{\delta-1} N[g], \sigma - N[\delta - 1]) \quad (4.91)$$

Sin embargo, si hubo una o más transmisiones nuevas perdidas en la cFR (i.e.  $\delta_{FRCA} > 0$ ), al salir de la FR la *cwnd* queda desplazada de forma que comienza con la primera de

---

<sup>19</sup>En [61] se definen dos opciones para la actualización de  $w$  al salir del FR. En el algoritmo se usa la segunda de ellas.

ellas. En este caso, el valor de  $N[\delta]$  dependerá de la relación entre la cantidad de transmisiones nuevas posteriores a la primera pérdida y el tamaño actual de  $w$ . Considérese la figura 4.16 como ejemplo. La primera pérdida de un segmento nuevo ocurre en la ronda  $f_\delta = 1$ . La posición de esta pérdida en la  $cwnd$  enviada al inicio de esa ronda es  $\beta_\delta$ . Como consecuencia de la pérdida, el límite izquierdo de la  $cwnd$  queda detenido en ella al salir de la FR, por lo que:

$$a = m + \sum_{g=0}^{f_\delta-1} N[g] + \beta_\delta \quad (4.92)$$

Debido a los pasos [FR.2.] y [FR.3.],  $w$  se aumenta temporalmente durante la FR y todos los segmentos nuevos posteriores a la primera pérdida que fueron transmitidos en esta fase (hasta la ronda  $\delta - 1$ ) quedan dentro de la  $cwnd$ . Como se mencionó anteriormente, el valor de  $w$  al salir de la FR está dado por (4.76),  $w = \sigma$ . AL comenzar la CA se transmiten en ráfaga los  $N[\delta]$  segmentos que faltan para completar la  $cwnd$ , los cuales se calculan como  $N[\delta] = \sigma - (\sum_{g=f_\delta}^{\delta-1} N[g] - \beta_\delta)$ . Si este resultado es negativo, significa que la cantidad de segmentos en tránsito (transmitidos pero sin reconocer con un RA) es mayor que  $w$ , de forma que hay segmentos fuera del límite derecho de la  $cwnd$  que ya se han transmitido. En este caso, la CA no se tiene permiso para transmitir ningún segmento más y  $N[\delta]$  debe ser cero. Es decir, se requiere de un  $\max(0, N[\delta])$ . Generalizando, en una cFR con transmisiones nuevas perdidas,  $N[\delta]$  se define como:

$$N[\delta] = \min(C_- - m - \sum_{g=0}^{\delta-1} N[g], \max(0, \sigma - \sum_{g=f_\delta}^{\delta-1} N[g] - \beta_\delta)) \quad (4.93)$$

A lo largo de la FR, siempre el último segmento transmitido también es el segmento con SN máximo, por lo que al salir de esta fase  $m$  se actualiza como:

$$m = s \quad (4.94)$$

Por las características de operación de una cFR se concluye que:

- $n_{TO=0}$  siempre al entrar y salir de una FR.
- Al salir de una cFR, siempre se cumple que  $m = s$ .
- En una cFR en donde no hay transmisiones nuevas ( $N_{FR} = 0$ ),  $m = s = a = r$  al salir de la fase.
- Si  $N_{FR} > 0$  al término de una cFR,  $a \geq r$  al salir de la fase.

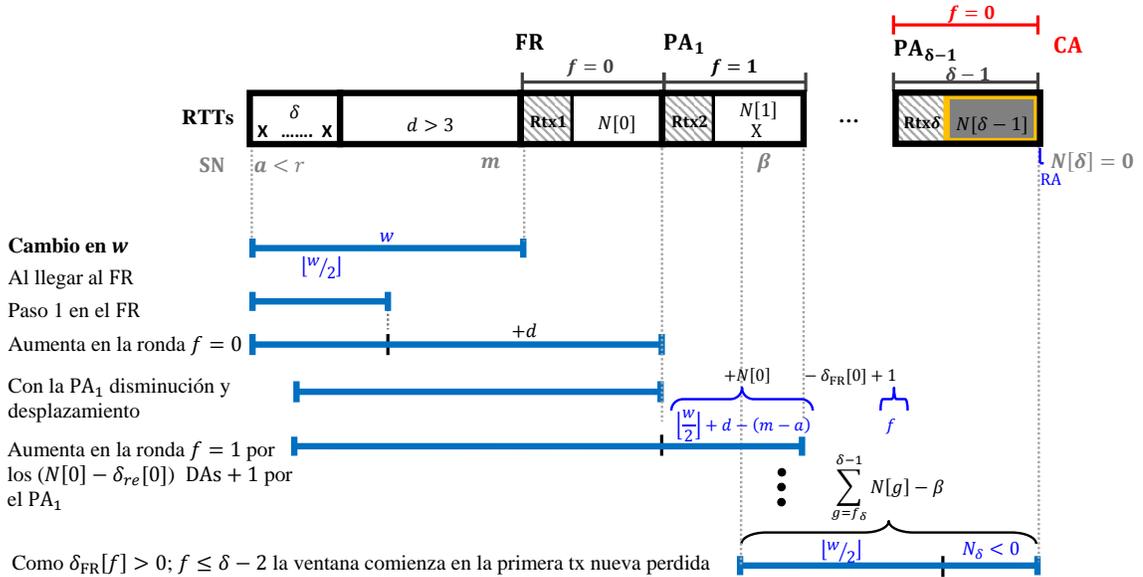


Figura 4.16: cFR con transmisiones nuevas perdidas.

- Si una cFR tuvo transmisiones nuevas y ninguna se perdió, es decir, si  $N_{FR} > 0$  y  $\delta_{FRCA} = 0$  al concluir una cFR, entonces  $m = s = a > r$ .
- Si  $C_- \gg m$ , entonces el tamaño de la  $cwnd$  al finalizar una cFR sin pérdidas ( $\delta_{FRCA} = 0$ ) es  $N[\delta - 1] + N[\delta] = \sigma$ .
- Si una cFR tuvo transmisiones nuevas, forzosamente la FR se activó para recuperar un LE con  $\delta > 1$  pérdidas.
- Si una cFR tuvo transmisiones nuevas y algunas de ellas se perdieron, es decir, si  $N_{FR} > 0$  y  $\delta_{FRCA} > 0$  al concluir una cFR, entonces  $m = s > a \geq r$ .
- Si  $C_- \gg m$ , entonces el tamaño de la  $cwnd$  al finalizar una cFR con pérdidas ( $\delta_{FRCA} > 0$ ) es  $N[\delta - 1] + N[\delta] < \sigma$ .
- Si la primera transmisión nueva dentro de la cFR se pierde, entonces  $a = r$  al finalizar la fase, lo que implicará la activación de un TO para recuperar este nuevo LE.
- Si  $m - a > w$  al salir de la cFR, y si el LE presentado en la FR se recupera mediante una FR siguiente, puede suceder que no haya transmisiones nuevas en ella o que las haya hasta después de varias rondas..

*iFR: Transición FR - TO.*

FR.5. El temporizador se restablece con cada una de las  $\delta$  retransmisiones (variante *Slow-but-Steady* [14]), de forma que éste expira con la primer retransmisión perdida, produciendo una transición FR - TO, con lo que la FR se transforma en una iFR. Se considera que la ronda en donde se envió la retransmisión causante del TO es  $0 \leq f_{TO} \leq \delta - 1$ . Como resultado de los pasos [FR.2.] y [FR.3.], dentro de las primeras  $f_{TO}$  rondas la cantidad de transmisiones nuevas que se pueden enviar está dada por (4.79) con  $f = 0, \dots, f_{TO}$ . Sin embargo, a partir de la ronda  $f_{TO}$  dejan de recibirse PAs, por lo que sólo se ejecuta el paso [FR.2.]. Como se muestra en la figura 4.17, esto continua hasta que el TO se activa. Por lo tanto, a partir de la ronda  $f_{TO}$  habrá  $\lceil RTO \rceil$  rondas más. Es decir, el número de rondas que dura una iFR es:

$$f_{FR} = f_{TO} + \lceil RTO \rceil \quad (4.95)$$

Ya que el mecanismo de *back-off* se restablece antes de entrar a una FR,  $n_{TO} = 0$ , por lo que la duración de la conexión al salir de la iFR se actualiza como:

$$A = A + f_{TO} + RTO \quad (4.96)$$

La cantidad de transmisiones nuevas que se pueden enviar en las rondas posteriores a la retransmisión fallida está dada por:

$$N[f] = \min(C_- - m - \sum_{g=0}^{f-1} N[g], \max(0, \sigma + d - m + a + f_{TO} - \sum_{g=0}^{f-1} \delta_{FR}[g])); f = f_{TO} + 1, \dots, f_{FR} \quad (4.97)$$

con  $f_{FR}$  definida en (4.95).

Por lo tanto, usando (4.95), la cantidad de transmisiones nuevas realizadas en la iFR se cuantifican como:

$$N_{FR} = \sum_{g=0}^{f_{FR}} N[g] \quad (4.98)$$

y la cantidad de transmisiones nuevas perdidas durante la iFR se contabiliza en:

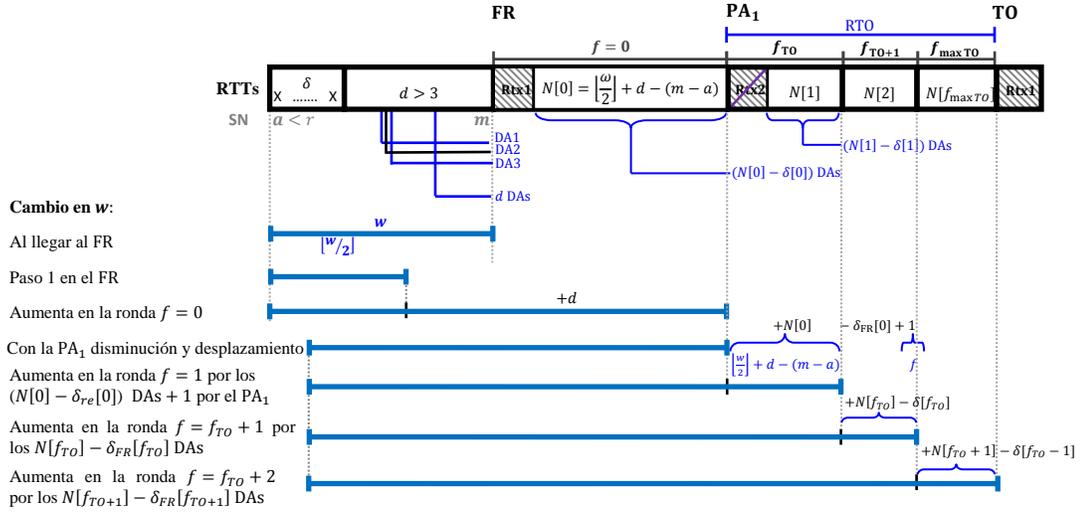


Figura 4.17: iFR provocada por la pérdida de una retransmisión.

$$\delta_{FRTO} = \sum_{g=0}^{f_{FR}} \delta_{FR}[g] \quad (4.99)$$

La cantidad de segmentos enviados sin pérdida en una iFR incluye tanto a las  $f_{TO}$  retransmisiones exitosas previas a la retransmisión fallida, como a todas las transmisiones nuevas sin pérdida que se consideran dentro de la FR, de manera que  $X$  se actualiza al final de la iFR como:

$$X = X + f_{TO} + N_{FR} - \delta_{FRTO} \quad (4.100)$$

con  $N_{FR}$  definida en (4.98). Para la cantidad de segmentos enviados en total se toman en cuenta, adicionalmente, la retransmisión fallida y las transmisiones nuevas perdidas, de forma que  $Y$  se actualiza al paso por una iFR como:

$$Y = Y + N_{FR} + f_{TO} + 1 \quad (4.101)$$

con  $N_{FR}$  definida en (4.98). La variable  $\delta$  se actualiza al salir de la iFR con el número de transmisiones nuevas perdidas en esta fase menos las pérdidas del LE anterior que se lograron recuperar mediante retransmisiones exitosas, es decir:

$$\delta = \delta - f_{TO} + \delta_{FRTO} \quad (4.102)$$

con  $f_{FR}$  definida en (4.95). El TO se debe a la pérdida de la retransmisión de alguno de los segmentos perdidos en el LE previo, cuyo SN no está disponible en el algoritmo. Por lo tanto, el valor de  $a$  no cobra importancia en este caso. Tampoco importa el valor de las variables  $A_{CA\_aux}$  y  $d$  al finalizar una iFR, puesto que éstos no se usan posteriormente en el algoritmo.

El SN del último segmento transmitido en la FR, que también corresponde al máximo SN, se puede calcular con (4.89) y (4.94), usando  $N_{FR}$  definida como en (4.98).

#### 4.2.4. Fase TO

El temporizador de retransmisión se actualiza con cada RA recibido [11], incluyendo los PAs (variante “lento pero firme” de TCP NewReno [14]). En el emisor TCP, un evento de expiración del temporizador, TO, se interpreta como la indicación de que el primer segmento aún no reconocido se ha perdido, por lo que se inicia con la recuperación del LE a través del siguiente algoritmo:

TO.1. Se incrementa el factor de *back-off*:

$$n_{TO} = n_{TO} + 1 \quad (4.103)$$

de forma que un siguiente TO asociado con una retransmisión pendiente o incluso con el segmento inmediato posterior a la LW, tarará el doble en generarse. Se reinicia el temporizador [11].

TO.2. Si es el primer TO para un cierto SN, entonces  $\sigma$  se actualiza según (4.75) [15]<sup>20</sup>. Se actualiza la variable  $r = m$ , como en (4.74), y la  $cwnd$  se reduce a la  $lwnd$ , que equivale a 1 SMSS [15], es decir:<sup>21</sup>

$$w = 1 \quad (4.104)$$

TO.3. Se retransmite el segmento perdido.

---

<sup>20</sup>Ver la nota al pie 18.

<sup>21</sup>En el estándar, la actualización de las variables  $r$  y  $w$  se indica en todos los TOs; sin embargo, en el caso de ocurrir un TO sucesivo, es decir, un TO subsecuente para el mismo SN, la actualización de las variables  $r$  y  $w$  no alteraría los valores que éstas adoptaron en el primer TO para este SN. Por lo tanto, con fines de eficiencia, sólo se realiza una vez en el algoritmo dicha actualización.

*Transición TO - SS.*

TO.4. Si la retransmisión es exitosa, se restablece la variable<sup>22</sup>

$$\delta = 0 \quad (4.105)$$

y se continúa la transmisión a partir de ella usando un SS siguiente.

*Aproximación TO-SS:* Como se indica en la figura 4.1, esta transición existe si la retransmisión del segmento que activó el TO es exitosa. En este caso, según el paso [TO.4.], su ACK provocará el aumento de  $w = 2$  como se establece en el SS. Así, esta retransmisión exitosa es, en realidad, el primer segmento transmitido en el SS, es decir, la transmisión en su primera ronda, cuando  $w = 1$ . Por lo tanto, esta retransmisión no se refleja en las variables globales sino hasta salir del SS.

Aunque al tener un TO la transmisión se recomienza a partir del segmento perdido que lo activó, no necesariamente se van a retransmitir todos los segmentos a partir de él. Ésto se debe a que el receptor TCP almacena los segmentos fuera de orden que recibe en su cola de recepción. Cuando llega la retransmisión del segmento perdido que activó el TO sucede una de dos cosas:

- a) Si no hay más pérdidas pendientes por recuperar, entonces la retransmisión exitosa llena totalmente la secuencia de segmentos en la cola de recepción y el receptor TCP responde con un ACK acumulado que reconoce hasta el SN máximo transmitido,  $m$ . En el transmisor, la *cwnd* se desplaza para comenzar con el segmento con  $SN = m + 1$  en la segunda ronda del SS. En este caso, únicamente se envió una retransmisión.
- b) Si todavía hay retransmisiones pendientes, es decir, si aún no se han recuperado por completo todas las pérdidas, entonces la secuencia de segmentos en la cola de recepción se llena parcialmente con la llegada de la retransmisión exitosa. El receptor envía un ACK acumulado con el que se reconoce hasta el último segmento en orden y la *cwnd* se desplaza hasta comenzar con la siguiente pérdida.

---

<sup>22</sup>En el algoritmo,  $\delta$  no se actualiza al finalizar el TO, porque este valor no se usa más adelante.

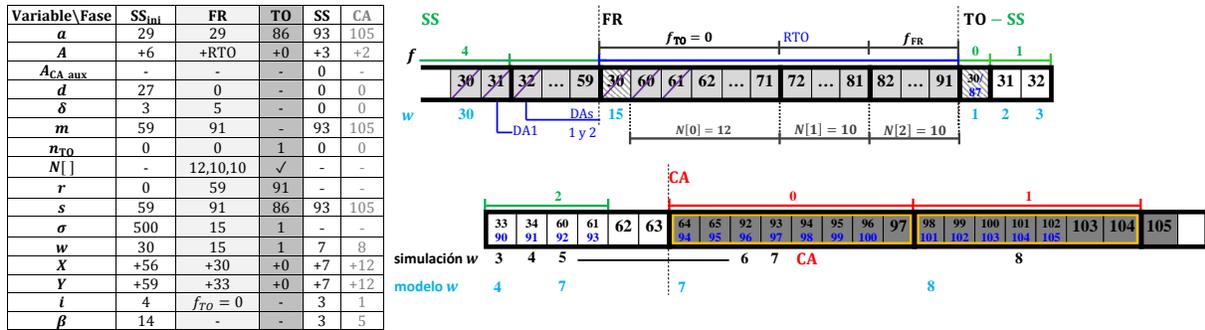


Figura 4.18: Fragmento de una conexión TCP NewReno. Caso G.

De forma que se salta la transmisión de todos los segmentos no perdidos entre la pérdida que activó el LE y la siguiente pérdida. En la segunda ronda del SS, se envían los  $w = 2$  segmentos de la *cwnd*.

En caso de ser más de dos la pérdidas que se deben recuperar después del TO, la dinámica en la transmisión de TCP resulta con saltos en la secuencia de SNs, pues la *cwnd* se va desplazando conforme llegan los ACKs acumulados para comenzar con la siguiente pérdida. Sin embargo, ésto tampoco quiere decir que sólo se retransmiten los segmentos perdidos: En cada ronda se transmite en ráfaga la *cwnd* completa; si el segmento perdido al inicio de la *cwnd* está seguido de uno o más segmentos que no fueron perdidos, entonces éstos se retransmitirán innecesariamente. Así la cantidad de segmentos que se retransmiten dependerá, en cada momento, de la relación entre la distancia entre las pérdidas contra el tamaño de la *cwnd*.

Para ejemplificar esta situación, considérese el ejemplo de la figura 4.18. En el SS inicial un LE ocurre. La LW abarca los segmentos [30 - 59], con los segmentos 30, 31 y 32 perdidos. La primera retransmisión de la FR falla, con lo que sucede un nuevo LE. El tamaño de LW se incrementa por el paso [FR.2.], para abarcar los segmentos [30 - 91], con los segmentos 30, 31, 32, 60 y 61 perdidos. Se activa el TO. En la primera ronda del SS siguiente se envía la retransmisión del segmento 30. Su RA permite que  $w = 2$ , de forma que en la segunda ronda se envían las retransmisiones de los segmentos 31 y 32. Hacia el final de esta ronda, el RA 31 genera el aumento  $w = 3$  y el desplazamiento de la *cwnd* para abarcar los segmentos [32 - 34], lo que resulta en la retransmisión innecesaria de los segmentos no perdidos 33 y 34. Inmediatamente después del RA 31 llega el ACK generado por la retransmisión exitosa del segmento 32, el cual es un ACK acumulado que reconoce hasta el segmento 59. La *cwnd* se aumenta a  $w = 4$ , y se recorre

para abarcar los segmentos [60, 61, 62, 63]. De éstos, la tercera y cuarta retransmisión son innecesarias. El RA 60 permite que  $w = 5$  y genera retransmisión innecesaria de los segmentos 64 y 65. Finalmente, la retransmisión exitosa del segmento 61 hace que la  $cwnd$  crezca a  $w = 6$  y la desplaza para seguir la transmisión a partir del segmento  $m + 1$ , es decir, la  $cwnd$  se recorre para abarcar los segmentos [92 - 97]. Estrictamente, en ese instante se termina con la recuperación de la ventana de pérdida ( $a = r$ ).

Como se nota del ejemplo, es muy probable que después del TO no se haga una retransmisión completa de la LW sino que se salta la transmisión de algunos segmentos no perdidos en ella. La secuencia exacta de los segmentos enviados dependerá de la ubicación de las pérdidas en la LW.

A partir de un examen minucioso de los resultados de simulación de una serie de casos de conexiones TCP NewReno con TOs, se concluyó que la transmisión posterior a un TO incluye exactamente tantas retransmisiones como segmentos perdidos haya en la LW,  $\delta$ , si  $\delta = 1$  ó  $\delta = w$ . En cualquier otro caso, el número de retransmisiones es mayor (es decir, involucra retransmisiones innecesarias). Como la información de la LW disponible para el algoritmo consiste en su tamaño y en la cantidad de pérdidas (mas no su ubicación) en ella, no se modela el seguimiento de los SNs, sino sólo la cantidad de segmentos transmitidos. Para ello se introduce la:

*Aproximación SS/CA/FR - TO.* Al expirarse por primera vez un TO correspondiente a un cierto SN (TO no sucesivo), se actualizan las variables:

$$s = \begin{cases} s - \delta & ; (a \geq r) \text{ OR (FR-TO) OR } ((a < r) \text{ AND } a_s = 0) \\ a & ; ((a < r) \text{ AND } a_s = 1) \end{cases} \quad (4.106)$$

$$a_s = \begin{cases} 1 & ; s > a \\ 0 & ; s = a \end{cases} \quad (4.107)$$

$$a = s \quad (4.108)$$

Esto es, si un TO ocurre en alguna de las siguientes tres circunstancias:

- Cuando el último LE se ha recuperado estrictamente por completo, es decir, cuando  $a \geq r$ .

- Por una iFR, es decir, por una transición FR - TO.
- Durante la recuperación de un LE detectado mediante un TO, es decir, siendo  $a < r$ , pero en dicho TO el resultado de esta aproximación fue igual que como si no se hubiera aplicado (i.e.  $a_s = 0$ ), es decir, el valor calculado para  $a$  el igual al valor de la variable en el momento en que se presenta el TO.

la cantidad de retransmisiones posteriores a un TO se aproxima como  $\delta$ . Es decir, ya que no se tiene la información para saber exactamente cómo se realiza la dinámica de transmisión después de un TO, ésta se aproxima considerando que de la LW, que comprende los segmentos  $[a + 1 - s]$  con  $\delta$  pérdidas, se comienza la retransmisión a partir del segmento  $s - a + 1$ . Con esta aproximación se pierde la posibilidad de dar seguimiento a los SNs de los segmentos conforme se van transmitiendo a lo largo de la conexión, pero se obtiene una buena aproximación a la cantidad de segmentos que se transmite en ella. Por el contrario, si durante la recuperación de un LE detectado mediante un TO se produce una nueva pérdida, es decir, si  $a < r$  y en dicho TO se aplicó esta aproximación, entonces ya no se realiza de nuevo; es decir, ya no se produce un salto para comenzar la retransmisión, pues éste ya se realizó en el TO previo.

Por ejemplo, en la figura 4.7, un TO se activa por la pérdida del segmento 82, en la ronda  $f = 3$  de una FR. El LE se extiende hasta la CA siguiente, de forma que La LW abarca los segmentos  $[82 - 85]$ . Hay sólo  $\delta = 1$  pérdida en la LW. La condición  $a > r$  se cumple en la activación de este TO, lo que se refleja en que  $m = s = 85$  al término de la CA y permite la aplicación de la *Aproximación SS/CA/FR - TO*. La recuperación real consistiría en lo siguiente: Una vez ocurrido el TO, se retransmite el segmento 82, el cual es reconocido a través de un ACK acumulado que reconoce hasta el segmento 85; lo que permite que continúe la transmisión en secuencia a partir de los segmentos 86 y 87 en la segunda ronda del SS. Por otro lado, debido a la *Aproximación SS/CA/FR - TO*, en el TO se actualizan  $a = s = s - \delta = 84$  y  $a_s = 1$ . Es decir, en lugar de hacer el análisis de la dinámica real de transmisión de la conexión, en el algoritmo se considera que después del TO se tiene una transmisión en secuencia a partir del segmento  $a + 1 = 85$ . En este caso, la cantidad de segmentos transmitida por causa de este TO es adecuadamente aproximada por el algoritmo. En este caso, la único error en la aproximación es respecto del SN de la retransmisión. Esto sucede, de hecho, en todos los LEs con  $\delta = 1$  recuperados a partir de un TO: el SN de la retransmisión es distinto del real, pero una vez recibido el ACK acumulado, el seguimiento en los SNs a través del algoritmo corresponde con el de la conexión.

En la figura 4.3 se presentan otros casos de ejemplo: Un TO se presenta con la pérdida del segmento 84, transmitido en la ronda  $f = 1$  de una cFR. La LW abarca los segmentos [84 - 98], todos ellos perdidos, por lo que  $\delta = 15$ . En este caso la *Aproximación SS/CA/FR - TO* se puede aplicar porque  $a > r$  al finalizar la CA, sin embargo, el resultado de la aproximación no altera el valor real de  $a$ , por lo que  $a_s = 0$ . Es decir, se calcula  $a = s = s - \delta = 98 - 15 = 83$ , por lo que en el algoritmo se considera que la transmisión continúa a partir del segmento 84, lo cual coincide con la recuperación real. En este caso, por lo tanto, la aproximación también resulta adecuada. Se puede demostrar que, de hecho, no hay error debido a esta aproximación en todos los casos en los que la LW tenga todos sus segmentos perdidos.

Continuando con los ejemplos de la figura 4.3, se presenta un nuevo LE por la pérdida de la retransmisión del segmento 91 en la primer ronda de la CA. La condición de que no se completó el intento de recuperación del LE, es decir, que no se retransmitieron todos los segmentos perdidos (independientemente de que se perdieran de nuevo o no), se refleja en las condiciones  $s < m$  y  $a < r$  al finalizar la fase. El nuevo LE tiene sólo  $\delta = 1$  pérdidas. Aunque  $a < r$ , la *Aproximación SS/CA/FR - TO* se puede aplicar porque en el TO previo el resultado de la aproximación fue tal que no hubo un salto para comenzar la retransmisión, es decir,  $a_s = 0$ . Al aplicar de nuevo a aproximación se considera que la transmisión continúa en secuencia a partir del segmento 97 (en lugar de la retransmisión del 91 y el salto en la secuencia de transmisión al segmento 98), por lo que  $a_s = 1$ . En este caso la aproximación también es adecuada. El tercer TO en esta figura se activa por la pérdida del segmento 99. De nuevo es un LE con  $\delta = 1$  y la aproximación resulta adecuada.

El error que esta aproximación introduce depende de la ubicación de los errores en la LW. En la figura 4.8 se muestra un ejemplo en el que se subestima por un segmento la cantidad de segmentos enviados después del TO. En la figura, se muestra la numeración de los segmentos para su comparación entre lo que sería realmente (el número superior) y lo que se considera por el algoritmo (el número inferior). La pérdida que activa el TO es la del segmento 94. La LW abarca los segmentos [94 - 100], de los cuales se pierden los segmentos [94 - 99]. La condición  $a > r$  se cumple en la activación de este TO, por lo que  $m = s = 100$  al término de la CA. Debido a la *Aproximación SS/CA/FR - TO*, se actualizan  $a = s = s - \delta = 100 - 6 = 94$  y  $a_s = 1$ . En la recuperación real, después del

TO se retransmite el segmento 94, mientras que para el algoritmo esta retransmisión es del segmento 95. En la segunda ronda del SS, en la recuperación real se retransmiten los segmentos 95 y 96, mientras que para el algoritmo se suponen los segmentos 96 y 97. El ACK de la retransmisión real del segmento 96 provoca la transición SS - CA y la transmisión de los segmentos [97 - 99]. Para el algoritmo, en la primera ronda de la CA se transmiten los segmentos [98 - 100]. En ambos casos, tanto el real como el supuesto por el algoritmo, al final de esta ronda se termina con la recuperación completa del LE. Sin embargo, el SN en el algoritmo es uno más que el real, lo que implica una subestimación de un segmento en la cantidad de segmentos enviados a partir de este TO.

Regresando al ejemplo de la figura 4.18. Después de que el TO se activa,  $a = s = s - \delta = 91 - 5$  por la *Aproximación SS/CA/FR - TO*. Esto implica que, después del TO, en lugar de suponer la retransmisión del segmento 30, se supone la del segmento 87, por lo que  $a_s = 1$ . En la segunda ronda del SS se supone la transmisión de dos los siguientes segmentos en secuencia a partir del 87, mientras que realmente se retransmiten los segmentos 31 y 32. En la ronda  $f = 2$  del SS, se considera la retransmisión de los siguientes segmentos en secuencia [90 - 93], mientras que en la realidad se retransmiten los segmentos [33, 34, 60 - 63]. Las retransmisiones innecesarias en el SS se responden con DAs, de manera que no contribuyen para el crecimiento de  $w$ . En el algoritmo no se considera que se realicen retransmisiones innecesarias, por lo que todos los ACKs contribuyen al crecimiento de  $w$ . Esta diferencia se nota en la comparación del tamaño de  $w$  en la simulación contra  $w$  en el algoritmo. Debajo de cada segmento está el valor de  $w$  cuando la *cwnd* comienza con él. Por ejemplo, en la simulación, el ACK del segmento 33 es un DA, por lo que no afecta a  $w$ . Sin embargo, para el algoritmo, todas las retransmisiones son necesarias así que la retransmisión 90 (que sustituye a la 33) provoca un crecimiento  $w = 4$ . La transición SS - CA ocurre con el primer RA recibido siendo  $w = \sigma$ . En el algoritmo, esta transición ocurre antes que en la simulación, pues  $w = 7$  desde la llegada del RA 92 (que sustituye al 60), en cambio, en la simulación  $w = 7$  hasta con el RA del segmento 93. En este caso, por lo tanto, se subestima la cantidad de segmentos enviada así como la duración de la conexión.

En la figura 4.19 se presenta otro caso en donde la *Aproximación SS/CA/FR - TO* también resulta en una subestimación tanto de la cantidad de segmentos enviados como de la duración de la conexión. A partir de la pérdida del segmento 30 en la ronda  $i = 4$

Variable\Fase	SS <sub>ini</sub>	FR	TO	SS	CA
$a$	29	29	83	90	105
$A$	+6	+RTO	+0	+3	+2
$A_{CA\_aux}$	-	-	-	0	-
$d$	27	0	0	0	0
$\delta$	3	6	0	0	0
$m$	59	89	✓	90	105
$n_{TO}$	0	0	1	0	0
$N[ ]$	-	12, 9, 9	✓	✓	✓
$r$	0	59	89	✓	✓
$s$	59	89	83	90	105
$\sigma$	500	15	7	✓	✓
$w$	30	15	1	7	9
$X$	+56	+27	+0	+7	+15
$Y$	+59	+31	+0	+7	+15
$i$	4	$i_{TO} = 0$	-	3	1
$\beta$	14	-	-	3	8

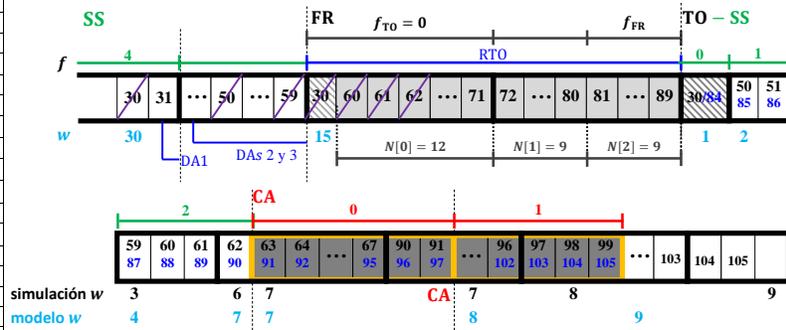


Figura 4.19: Fragmento de una conexión TCP NewReno. Caso H.

del SS, un LE se presenta con  $\delta = 3$  pérdidas en la LW conformada por los segmentos [30 - 59]. Su recuperación se intenta a través de una FR. Sin embargo, la primera retransmisión falla, por lo que el LE se extiende hasta el final de la iFR, de forma que la LW crece para abarcar los segmentos [30 - 89], con  $\delta = 6$  pérdidas. Al cabo de una RTO la iFR termina y se presenta un TO. Aunque  $a < r$ , la Aproximación SS/CA/FR - TO se puede aplicar ya que proviene de una iFR. En la recuperación real, la primera retransmisión después del TO es, de nuevo, la del segmento 30. Con su RA acumulado, la  $cwnd$  se desplaza hasta la siguiente pérdida para transmitir los segmentos 50 y 51 en la ronda  $f = 1$  del SS. De éstos, el único perdido fue el 50; así que se tiene la retransmisión innecesaria del segmento 51. El segmento 50 se reconoce con un RA acumulado, con el que crece  $w = 3$  y se desplaza para abarcar los segmentos [59 - 61]. La retransmisión innecesaria del 51 se reconoce con un DA, que no altera el valor de  $w$  ni desplaza la  $cwnd$ . Los tres segmentos del 59 al 61 son retransmisiones necesarias por lo que sus RAs provocan que  $w$  crezca hasta  $w = 6$  y se desplace para abarcar los segmentos [62 - 67]. La retransmisión del segmento 62 también es necesaria. La recuperación del LE termina estrictamente con la llegada del RA 62, el cual provoca que crezca  $w = 7$  y se desplace hasta comenzar con el segmento  $m + 1$ , abarcando los segmentos [90 - 96]. Los segmentos [63 - 67] son retransmisiones innecesarias cuyos DAs no afectan a la  $cwnd$ . Por otro lado, la Aproximación SS/CA/FR - TO provoca un salto en la transmisión para continuar con los segmentos en secuencia a partir del 84. De nuevo, en el algoritmo se considera que todas las retransmisiones son necesarias, así que sus ACKs afectan el crecimiento de la  $cwnd$  y su desplazamiento. De forma que, para el algoritmo, la retransmisión exitosa de los segmentos 85 y 86 en la segunda ronda del SS hace que crezca  $w = 4$  y se desplace hasta [87 - 90]. La recepción del RA

Variable\Fase	SS <sub>ini</sub>	FR	TO	SS	CA
$a$	29	29	83	90	105
$A$	+6	+RTO	+0	+3	+2
$A_{CA\_aux}$	-	-	-	0	-
$d$	27	0	-	0	0
$\delta$	3	6	-	0	0
$m$	59	89	-	90	105
$n_{TO}$	0	0	1	0	0
$N $	-	12, 9, 9	-	-	-
$r$	0	59	89	-	-
$s$	59	89	83	90	105
$\sigma$	500	15	7	-	-
$w$	30	15	1	7	9
$X$	+56	+27	+0	+7	+15
$Y$	+59	+31	+0	+7	+15
$i$	4	$f_{ro}=0$	-	3	1
$\beta$	14	-	-	3	8

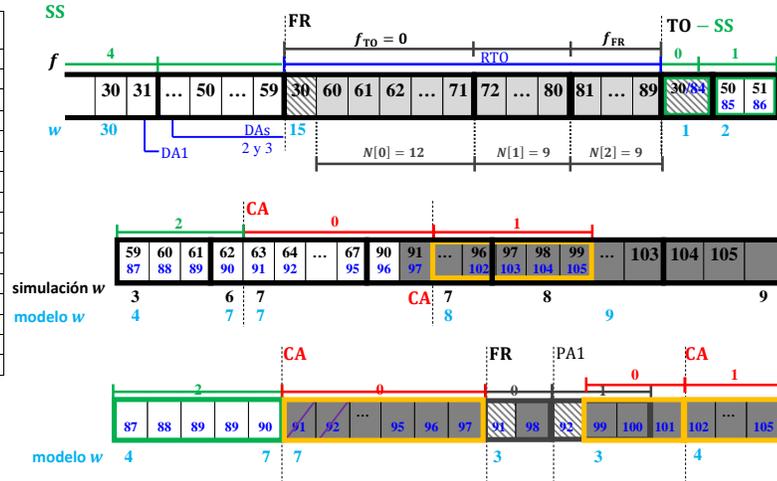


Figura 4.20: Fragmento de una conexión TCP NewReno. Caso I.

87 provoca la transición SS - CA en el algoritmo y el desplazamiento de la  $cwnd$  hasta [91 - 97]. Por lo tanto, en este ejemplo, hay un adelanto de 6 segmentos y de un RTT en la recuperación supuesta por el algoritmo que en la real.

En la figura 4.20 se presenta una conexión cuya secuencia de fases diverge de la de la figura 4.19 a partir de la CA, en la que se pierden las retransmisiones innecesarias 98 y 99. Como se ha mencionado anteriormente, las retransmisiones innecesarias no tienen ningún efecto en el transmisor, por lo que su pérdida no afecta la dinámica real de la transmisión. Sin embargo, en el algoritmo todas las retransmisiones son necesarias, así que la pérdida de las retransmisiones 91 y 92, correspondientes a las retransmisiones reales 98 y 99, implican un nuevo LE y su recuperación mediante una FR. Esto acarrea un error tanto en la estimación de la cantidad de segmentos enviados como en la duración de la conexión a partir de este evento. En este caso A se sobrestima, lo que se puede notar como sigue: en la transmisión real, el segmento 105 se transmite en el séptimo RTT después del TO; mientras que para el algoritmo este segmento se transmite en la octava ronda después del TO. Un caso de una sobrestimación mayor de A se observa en la figura 4.21. En este ejemplo se pierde la retransmisión innecesaria del segmento 51 que, de nuevo, en la simulación no tiene efecto; sin embargo, para el algoritmo implica un siguiente TO. Puesto que es una retransmisión del LE anterior la que se pierde, se cumple la condición  $a < r$ . En consecuencia, el mecanismo de *back-off* sigue activo y el TO se presenta después de 2 RTO rondas. Como resultado, la duración de la conexión se sobrestima: por ejemplo, el segmento 105 se supone que se transmite

Variable\Fase	SS <sub>ml</sub>	FR	TO	SS	CA
$\alpha$	29	29	83	90	105
$A$	+6	+RTO	+0	+3	+2
$A_{CA\_aux}$	-	-	-	0	-
$d$	27	0	0	0	0
$\delta$	3	6	0	0	0
$m$	59	89	-	90	105
$n_{TO}$	0	0	1	0	0
$N[ ]$	-	12, 9, 9	-	-	-
$r$	0	59	89	-	-
$s$	59	89	83	90	105
$\sigma$	500	15	7	-	-
$w$	30	15	1	7	9
$X$	+56	+27	+0	+7	+15
$Y$	+59	+31	+0	+7	+15
$i$	4	$f_{TO} = 0$	-	3	1
$\beta$	14	-	-	3	8

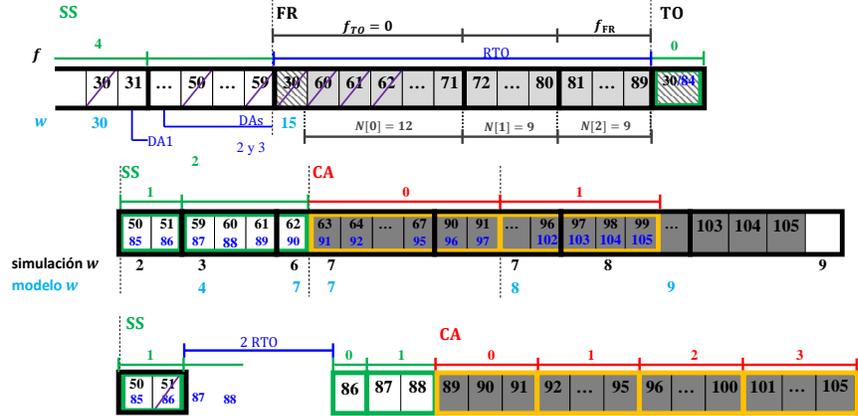


Figura 4.21: Fragmento de una conexión TCP NewReno. Caso J.

1 + 2RTO después de su transmisión real.

De los ejemplo anteriores, es claro que la cantidad de situaciones que pueden presentarse en la recuperación de una LW a partir de TO, y por lo tanto, las posibilidades en la dinámica de la transmisión de una conexión TCP NewReno son innumerables, por lo que la complejidad en su análisis crece sobremanera. Para evitar tal grado de complejidad, en este trabajo se usa la *Aproximación SS/CA/FR - TO*.

#### Transición TO - TO.

TO.5. Si la retransmisión falla, un nuevo TO ocurre (TOs sucesivos) después de un tiempo igual a la RTO ponderada por el mecanismo de *back-off*, y se recomienza con el paso TO.1. En este caso, sólo un segmento es enviado en la fase (la retransmisión fallida), de forma que:

$$Y = Y + 1 \quad (4.109)$$

Y la duración de la conexión se actualiza como:

$$A = A + \min(RTO_{max}, 2^{n_{TO}}RTO) \quad (4.110)$$

Las demás variables globales no se alteran. A partir del ejemplo en la figura 4.12 se puede notar que la actualización de las demás variables globales  $a$ ,  $r$ ,  $s$ ,  $\sigma$  y  $w$  sólo es necesaria en un TO no sucesivo (primer TO en la figura). La forma en que se actualizan las variables  $A$  y  $Y$  depende de si el TO termina con una transición TO - TO o con una TO - SS (que es el caso del segundo TO en la figura). Finalmente, el factor  $n_{TO}$  se incrementa en uno siempre que sucede un TO.

### 4.3. Algoritmo

<b>PARÁMETROS DE ENTRADA:</b>	<p>Tamaño, en segmentos, del flujo TCP NewReno, <math>C_{-}</math>.</p> <p>RTO promedio, <math>RTO</math>, en rondas.</p> <p>RTO inicial, en rondas, <math>RTO_{ini}</math>.</p> <p>RTT promedio, <math>R_{-}</math>.</p> <p>SMSS</p> <p><i>ssthresh</i> inicial, <math>\sigma_{ini}</math>.</p> <p>Parámetros de entrada del módulo de pérdidas. Si es un módulo Gilbert:  <math>p_{01}</math>, <math>p_{10}</math>, <math>p_{e0}</math> y <math>p_{e1}</math>.</p>
-------------------------------	--

<b>VARIABLES GLOBALES:</b>	<p>Duración de la conexión medida en rondas, <math>A</math>.</p> <p>Variable auxiliar en el cálculo de <math>a</math> en un TO, <math>A_{CA\_aux}</math>.</p> <p>SN recibido exitosamente máximo, <math>a</math>.</p> <p>Variable auxiliar en el cálculo de <math>a</math> en un TO, <math>a_s</math>.</p> <p>Posición del segmento que activa la transición a la siguiente fase dentro de la <math>cwnd</math> en la ronda <math>i</math> de cada fase, <math>\beta</math>.</p> <p>Tamaño del flujo de datos, medido en segmentos, <math>C_-</math>.</p> <p>Cantidad de DAs al salir de una fase, <math>d</math>.</p> <p>Primer segmento perdido en la fase, <math>\Delta</math>.</p> <p>Cantidad de segmentos perdidos en la fase, <math>\delta</math>.</p> <p>Índice de las rondas dentro de cada fase, <math>f</math>.</p> <p>Equivale al número de rondas que duró la última fase FR, <math>f_{FR}</math>.</p> <p>Fase siguiente, FASE.</p> <p>Índice de la ronda en la que se presenta el segmento que provoca la transición a otra fase, <math>i</math>.</p> <p>SN transmitido máximo, <math>m</math>.</p> <p>Arreglo que contiene el número de segmentos transmitidos en cada ronda de la última FR, <math>N[f_{FR} + 1]</math>.</p> <p>Exponente de la base 2 con que se pondera a la RTO, <math>n_{TO}</math>.</p> <p>Variable <i>recover</i> de TCP NewReno, <math>r</math>.</p> <p>SN del último segmento transmitido., <math>s</math>.</p> <p><math>ssthresh</math>, <math>\sigma</math></p> <p>Tamaño de la <math>cwnd</math>, <math>w</math>.</p> <p>Cantidad de segmentos enviados exitosamente en el intervalo <math>A</math>, <math>X</math>.</p> <p>Cantidad de segmentos enviados en total en el intervalo <math>A</math>, <math>Y</math>.</p>
<b>INICIALIZACIONES:</b>	<p><math>a = m = r = X = Y = 0</math></p> <p><math>A = 0</math></p> <p><math>FASE = SS_{ini}</math></p> <p><math>n_{TO} = 0</math>.</p> <p><math>\sigma = \sigma_{ini}</math>.</p> <p><math>w = 1</math>.</p>
<b>PARÁMETROS DE SALIDA:</b>	<p>Tasa de transmisión efectiva, en Mbps, de la conexión TCP NewReno, <math>THR_E</math>.</p> <p>Tasa de transmisión, en Mbps, de la conexión TCP NewReno, <math>THR</math>.</p>
<b>CONDICIÓN DE PARO:</b>	<p><math>FASE = EC</math></p>

---

**ALGORITMO PARA LA EVALUACIÓN DEL THR DE UN FLUJO TCP NEWRENO  
DE TAMAÑO DETERMINADO**

---

```

1. while FASE ≠ EC do
2.     switch FASE
3.         case CA: execute ALGORITMO CA
4.         case SS: execute ALGORITMO SS
5.         case FR: execute ALGORITMO FR
6.         case TO: execute ALGORITMO TO
7.     end switch
8. end while
9. return   $THR_E \leftarrow \frac{X(SMSS)}{AR.(1E6)}$ ,  $THR \leftarrow \frac{Y(SMSS)}{AR.(1E6)}$ 

```

---



---

**ALGORITMO DE PÉRDIDAS**

---

<b>PARÁMETROS DE ENTRADA:</b>	Segmento inicial, $S_i$ . Segmento final, $S_f$ .
<b>PARÁMETROS DE SALIDA:</b>	Número de segmento perdido, $\Delta$ .

```

1. for  $S = S_i$  to  $S_f$  do
2.     if se pierde el segmento  $S$  then return  $\Delta \leftarrow S$  end if
3. end for
4. return  $\Delta \leftarrow 0$ .

```

---



---

**ALGORITMO LW**

---

<b>PARÁMETROS DE ENTRADA:</b>	Segmento inicial, $S_i$ . Ventana de transmisión, $W$ .
<b>PARÁMETROS DE SALIDA:</b>	Número de pérdidas en $W$ , $\delta_W$ Posición de la primera pérdida en $W$ , $\beta_W$ .

```

1. for  $S = S_i, \delta_W = 0$ , to  $S = W$  do
2.     if se pierde el segmento  $S$  then if es la primera pérdida then  $\beta_W \leftarrow S - 1$  end if,  $\delta_W \leftarrow \delta_W + 1$  end if
3. end for
4. return  $\delta_W$ 

```

---

---

**ALGORITMO SS**

---

1. **if**  $a = 0$  **then**
2.      $\Delta \leftarrow$  ALGORITMO DE PÉRDIDAS  $\{1, \text{mín}(C_- - a, \sigma)\}$ .
3. **else**
4.      $\Delta \leftarrow$  ALGORITMO DE PÉRDIDAS  $\{2, \text{mín}(C_- - a, \sigma)\}$ .
5. **end if**
6. **if**  $\Delta > 0$  **then**
7.      $i = \lfloor \log_2 \Delta \rfloor, \beta = \Delta - 2^i, a_{SS} = \Delta - 1, m_{SS} = \text{mín}(C_- - a, 2\Delta - 1), w = 2^i + \beta$
8.      $\delta \leftarrow 1 +$  ALGORITMO LW  $\{2, m_{SS} - a_{SS}\}$
9. **else**
10.    **if**  $C_- - a \leq \sigma$  **then**
11.        $a_{SS} = m_{SS} = C_- - a$
12.    **else**
13.        $i_\sigma = \lfloor \log_2 \sigma \rfloor, \beta_\sigma = \sigma - 2^{i_\sigma}, a_{SS} = m_{SS} = \sigma, w = 2^{i_\sigma} + \beta$
14.    **end if**
15. **end if**
16.  $s = a + m_{SS}, d = m_{SS} - a_{SS} - \delta, X = X + a_{SS} + d, Y = Y + m_{SS}, a = a + a_{SS}, m = \text{máx}(m, s)$
17. **if**  $a > r$  **then**  $n_{TO} = 0$  **end if**
18. **if**  $\delta = 0$  **then**
19.      $A = A + i + 1$
20.     **if**  $a = C_-$  **then** FASE  $\leftarrow$  EC **else** FASE  $\leftarrow$  CA **end if**
21. **else**
22.     **if**  $a > r$  AND  $d \geq 3$  **then**
23.       FASE  $\leftarrow$  FR
24.       **if**  $\beta = 0$  **then**  $A = A + i + 1$  **else**  $A = A + i + 2$  **end if**
25.     **else**
26.       FASE  $\leftarrow$  TO
27.       **if**  $\beta = 0$  **then**  $A = A + i + \text{mín}(\text{RTO}_{\text{max}}, 2^{n_{TO}} \text{RTO})$  **else**  $A = A + i + 1 + \text{mín}(\text{RTO}_{\text{max}}, 2^{n_{TO}} \text{RTO})$  **end if**
28.     **end if**
29. **end if**
30. **return**

---

---

## ALGORITMO CA

---

1.  $\delta_{CA} = 0$
2. **if**  $\delta = 0$  **then**
3.      $\Delta \leftarrow$  ALGORITMO DE PÉRDIDAS  $\{1, C_- - a\}$ .
4.     **if**  $\Delta > 0$  **then**
5.          $\delta_{CA} = 1, i = \lceil -\sigma - \frac{1}{2} + \sqrt{\sigma^2 - \sigma + \frac{9}{4} + 2\Delta} \rceil, \beta = \Delta - i(\sigma + \frac{i-1}{2}), a_{CA} = \Delta - 1, m_{CA} = \min(C_- - a, \Delta - 1 + \sigma + i)$
6.     **else**
7.          $i = \lceil -\sigma - \frac{1}{2} + \sqrt{(\sigma + \frac{1}{2})^2 - 2(\sigma - C_- + a)} \rceil, \beta = C_- - a - i(\sigma + \frac{i-1}{2}), a_{CA} = m_{CA} = C_- - a$
8.     **end if**
9.      $s = a + m_{CA}$
10. **else**
11.      $i = 0, a_{CA} = 0, m_{CA} = N[f_{FR} - 1] + N[f_{FR}], s = m + m_{CA}$
12.     **if**  $N[f_{FR}] = 0$  **then**  $A = A + i$  **else**  $A = A + i + 1$  **end if**
13. **end if**
14. **if**  $(\delta_{CA} > 0)$  OR  $(\delta > 0)$  **then**
15.      $\delta_{CA} \leftarrow \delta_{CA} +$  ALGORITMO LW  $\{\delta_{CA} + 1, m_{CA} - a_{CA}\}$
16. **end if**
17.  $w = \sigma + i, d = m_{CA} - a_{CA} - \delta_{CA}, X = X + a_{CA} + d, Y = Y + m_{CA}, a = a + a_{CA}, m = \max(m, s)$
18. **if**  $a > r$  **then**  $n_{TO} = 0$  **end if**
19. **if**  $\delta = 0$  AND  $\delta_{CA} = 0$  **then**
20.     **if**  $\beta \leq A_{CA\_aux}$  **then**  $A = A + i$  **else**  $A = A + i + 1$  **end if**
21.     FASE  $\leftarrow$  EC
22. **else**
23.     **if**  $a > r$  AND  $d \geq 3$  **then**
24.         FASE  $\leftarrow$  FR
25.         **if**  $\delta = 0$  **then**  $A = A + i + 1$  **else if**  $N[f_{FR}] = 0$  **then**  $A = A + i$  **else**  $A = A + i + 1$  **end if end if**
26.     **else**
27.         FASE  $\leftarrow$  TO
28.         **if**  $\delta = 0$  AND  $\beta \leq A_{CA\_aux}$  **then**  $A = A + i + \min(RTO_{max}, 2^{n_{TO}} RTO)$  **end if**
29.         **if**  $\delta = 0$  AND  $\beta > A_{CA\_aux}$  **then**  $A = A + i + 1 + \min(RTO_{max}, 2^{n_{TO}} RTO)$  **end if**
30.         **if**  $\delta > 0$  **then**  $A = A + i + \min(RTO_{max}, 2^{n_{TO}} RTO)$  **end if**
31.     **end if**
32.      $\delta = \delta + \delta_{CA}$
33. **end if**
34. **return**

---

**ALGORITMO FR**

---

1.  $\sigma \leftarrow \max(2, \lfloor \frac{w}{2} \rfloor)$ ,  $r \leftarrow m$ ,  $w \leftarrow \sigma$
  2. **for**  $f = 0$  **to**  $\delta - 2$  **do**
  3.  $N[f] \leftarrow \min(C_- - m - \sum_{g=0}^{f-1} N[g], \max(0, \sigma + d - m + a + f - \sum_{g=0}^{f-1} \delta_{FR}[g]))$
  4.  $\Delta \leftarrow$  ALGORITMO DE PÉRDIDAS {para la retransmisión  $f + 1$ }.
  5. **if**  $\Delta > 0$  **then go to** iFR **end if**
  6.  $[\delta_{FR}[f], \beta_{aux}] \leftarrow$  ALGORITMO LW { 1,  $N[f]$  }
  7. **if** es la primera pérdida en la FR **then**  $f_\delta \leftarrow f$ ,  $\beta_\delta \leftarrow \beta_{aux}$  **end if**
  8. **end for**
  9.  $\Delta \leftarrow$  ALGORITMO DE PÉRDIDAS {para la retransmisión  $\delta$ }.
  10. **if**  $\Delta > 0$  **then go to** iFR **end if**
  11.  $f_{FR} = \delta$ ,  $A = A + \delta$ ,  $A_{CA.aux} = N[\delta - 1]$ ,  $N_{FR} = \sum_{g=0}^{\delta-2} N[g]$ ,  $\delta_{FRCA} = \sum_{g=0}^{\delta-2} \delta_{FR}[g]$ ,  
 $X = X + \delta + N_{FR} - \delta_{FRCA}$ ,  $Y = Y + N_{FR} + \delta$ ,  $\delta = \delta_{FRCA}$ ,  $d = 0$ ,  $m = s = m + N_{FR}$ , FASE  $\leftarrow$  CA
  12. **if**  $\delta_{FRCA} = 0$  **then**
  13.  $a = s$ ,  $N[\delta] = \min(C_- - m - \sum_{g=0}^{\delta-1} N[g], \sigma - N[\delta - 1])$
  14. **else**
  15.  $a = m + \sum_{g=0}^{f_\delta-1} N[g] + \beta_\delta$ ,  $N[\delta] = \min(C_- - m - \sum_{g=0}^{\delta-1} N[g], \max(0, \sigma - \sum_{g=f_\delta}^{\delta-1} N[g] - \beta_\delta))$ .
  16. **end if**
  17. **return**
- 

**iFR**

---

18.  $f_{TO} \leftarrow f$ ,  $f_{FR} \leftarrow f_{TO} + \lceil \text{RTO} \rceil$
  19. **for**  $f = f_{TO} + 1$  **to**  $f_{FR}$  **do**
  20.  $N[f] = \min(C_- - m - \sum_{g=0}^{f-1} N[g], \max(0, \sigma + d - m + a + f_{TO} - \sum_{g=0}^{f-1} \delta_{FR}[g]))$
  21.  $\delta_{FR}[f] \leftarrow$  ALGORITMO LW { 1,  $N[f]$  }
  22. **end for**
  23.  $A = A + f_{TO} + \text{RTO}$ ,  $N_{FR} = \sum_{g=0}^{f_{FR}} N[g]$ ,  $\delta_{FRTO} = \sum_{g=0}^{f_{FR}} \delta_{FR}[g]$ ,  
 $X = X + f_{TO} + N_{FR} - \delta_{FRTO}$ ,  $Y = Y + N_{FR} + f_{TO} + 1$ ,  $\delta = \delta - f_{TO} + \delta_{FRTO}$ ,  
 $m = s = m + N_{FR}$ , FASE  $\leftarrow$  TO
  24. **return**
-

---

**ALGORITMO TO**

---

1.  $n_{TO} \leftarrow n_{TO} + 1$
  2. **if**  $a = 0$  **then** usa  $RTO_{ini}$  **else** usa  $RTO$  **end if**
  3. **if** es el primer TO para el mismo SN **then**
  4.      $r \leftarrow m, \sigma \leftarrow \text{máx}(2, \lfloor \frac{w}{2} \rfloor), w \leftarrow 1$
  5.     **if**  $(a \geq r)$  OR (FR-TO) OR  $((a < r)$  AND  $a_s = 0)$  **then**  $s \leftarrow s - \delta$  **else**  $s \leftarrow a$  **end if**
  6.     **if**  $s > a$  **then**  $a_s \leftarrow 1$  **else**  $a_s \leftarrow 0$  **end if**
  7.      $a = s$
  8. **end if**
  9.  $\Delta \leftarrow$  ALGORITMO DE PÉRDIDAS {para la retransmisión}.
  10. **if**  $\Delta > 0$  **then**  $A = A + \text{mín}(RTO_{max}, 2^{n_{to}} RTO), Y \leftarrow Y + 1, \text{FASE} \leftarrow TO$
  11. **else**  $\text{FASE} \leftarrow SS$  **end if**
  12. **return**
- 

#### 4.4. Validación del algoritmo.

La validación del algoritmo se realiza a través de su comparación contra un conjunto de pruebas de simulación en ns-2, de igual forma a como se llevó a cabo en la sección 3.5. La topología empleada para la simulación (mostrada en la figura 3.5) consiste en una red simple con sólo un enlace “cuello de botella”<sup>23</sup>, configurado como sigue:

- Tasa de transmisión del enlace de 100 Mbps.
- Cola tipo *Drop-Tail*, con capacidad suficiente como para no provocar pérdidas de paquetes por congestión.
- Se introduce un módulo de pérdidas, que fuerza el estado de pérdida de los paquetes que lo atraviesan según un modelo Gilbert simplificado.
- Retardo fijo.

El retardo en los enlaces de acceso de la fuente y del destino TCP se calcula para que, sumados con el retardo en el “cuello de botella”, den un RTT de 75 ms.

Se evalúa la tasa de transmisión del flujo TCP NewReno de tamaño determinado que atraviesa este enlace. La tasa se calcula mediante el cociente de la cantidad total de segmentos enviados

---

<sup>23</sup>Es necesario precisar que en ns-2, la congestión se presenta en los enlaces y no en los nodos. Es decir, cada enlace tiene asociada una cola, cuya saturación es la que provoca la pérdida de los paquetes.

por el transmisor TCP sobre el intervalo de tiempo en que duró la conexión. La cantidad de segmentos enviados por el transmisor TCP durante la conexión se obtiene de la lectura de una variable preestablecida en el simulador ns-2, la cual se actualiza en el transmisor TCP: *ndatapack\_* [55]. Como se mencionó en la discusión sobre la lectura de los parámetros a partir de la simulación, presentada en la sección 2.4.5, en un caso como éste, en donde no hay más flujos atravesando el enlace “cuello de botella”, esta cantidad equivale a las que arrojan los monitores de flujo y de la cola en el enlace. Se ejecutaron las pruebas para distintos tamaños del flujo TCP, medido en bits. Para cada punto sobre las curvas de ambas aproximaciones se calcula el error como está definido en [37], es decir, como  $|\text{simulación} - \text{modelo}| / \text{simulación}$ .

Respecto de los parámetros de entrada del algoritmo, el RTT promedio,  $R_$ , es un parámetro de la simulación, según el cual se calculan los retardos en los enlaces de la red; sin embargo también se corrobora calculando su promedio a lo largo de la conexión. De igual forma, durante la simulación, se calcula el promedio de la RTO, y su división entre  $R_$  se toma como el parámetro RTO.  $RTO_{ini} = 3s$ . Como *ssthresh* inicial, se usa un valor arbitrariamente alto:  $\sigma_{ini} = 500$ . El tamaño del flujo se varía en un rango de  $[5e4, 5e9]$ . El tamaño máximo de segmento se considera como  $SMSS = 960$  bytes.

Para la validación del algoritmo se emplea el modelo simplificado de Gilbert, con  $p_{e0} = 0$ ,  $p_{e1} = 1$  y los parámetros  $p_{01}$  y  $p_{10}$  variando en un rango dentro de los valores calculados de mediciones en [74] y [66], y que han sido usados en trabajos como [67] y [68].  $p_{01}$  se varía en un rango desde 0.003 hasta 0.08 y  $p_{10}$  toma valores entre 0.1 y 0.9; por lo que la longitud promedio de las ráfagas de paquetes perdidos, según (3.7), corresponde con valores desde 1.1 hasta 10. La tasa promedio de pérdidas, de acuerdo con (3.6), varía, en consecuencia, en un rango de  $[0.3\%, 44\%]$ . En las figuras 4.22 a 4.30 se muestran las curvas de la THR, en Mbps, para estos rangos de  $p_{01}$  y  $p_{10}$ . El rango de  $p$  correspondiente a cada caso se especifica en el cuadro 4.1.

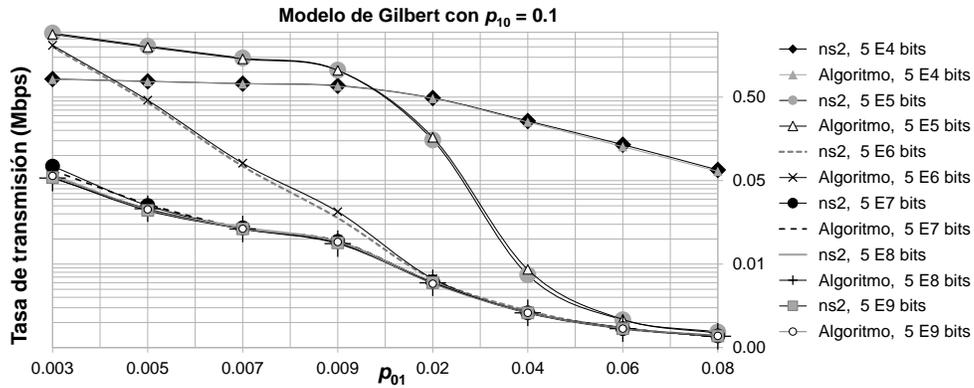


Figura 4.22: Tasa de transmisión, en Mbps, de un flujo TCP NewReno de tamaño determinado sobre una red con un modelo de pérdidas tipo Gilbert Simplificado, con  $p_{10} = 0,1$ .

$p_{10}$	$B_L$	Rango de $p$	Error promedio %
0,1	10.0	[0.029, 0.44]	3.4
0,2	5.0	[0.015, 0.28]	3.3
0,3	3.3	[0.010, 0.21]	2.3
0,4	2.5	[0.007, 0.16]	1.6
0,5	2.0	[0.006, 0.14]	1.3
0,6	1.7	[0.005, 0.12]	1.5
0,7	1.4	[0.004, 0.10]	1.9
0,8	1.3	[0.004, 0.09]	1.9
0,9	1.1	[0.003, 0.08]	2.0

Cuadro 4.1: Datos de gráficas en las figuras de resultados.

El error promedio del algoritmo resulta en 2.15 %. Es claro, por tanto, que una representación más fiel de los mecanismos del protocolo en la herramienta de evaluación se traduce en una mayor precisión de sus resultados.

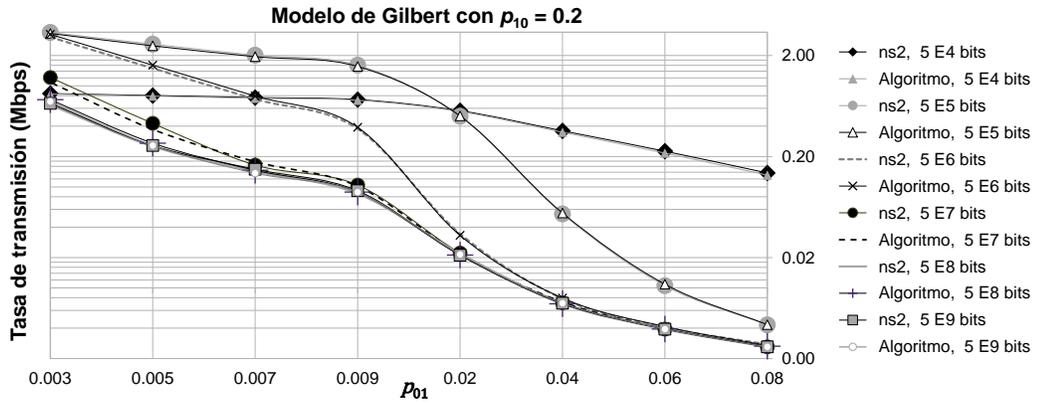


Figura 4.23: Tasa de transmisión, en Mbps, de un flujo TCP NewReno de tamaño determinado sobre una red con un modelo de pérdidas tipo Gilbert Simplificado, con  $p_{10} = 0,2$ .

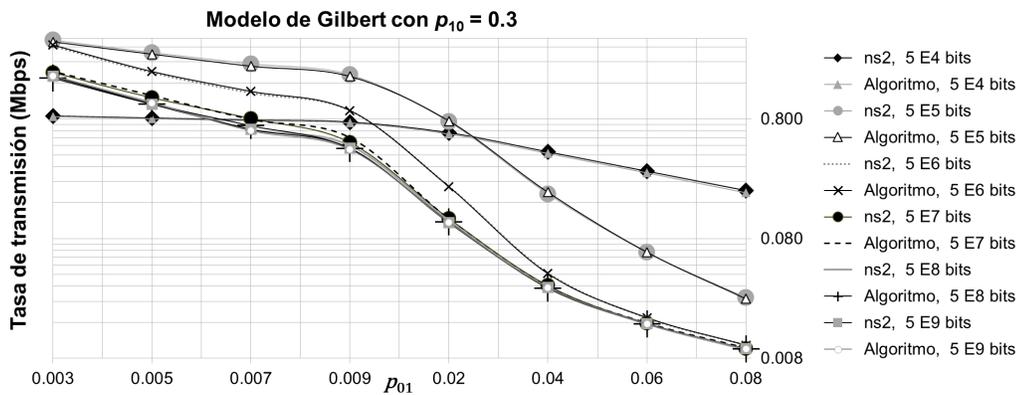


Figura 4.24: Tasa de transmisión, en Mbps, de un flujo TCP NewReno de tamaño determinado sobre una red con un modelo de pérdidas tipo Gilbert Simplificado, con  $p_{10} = 0,3$ .

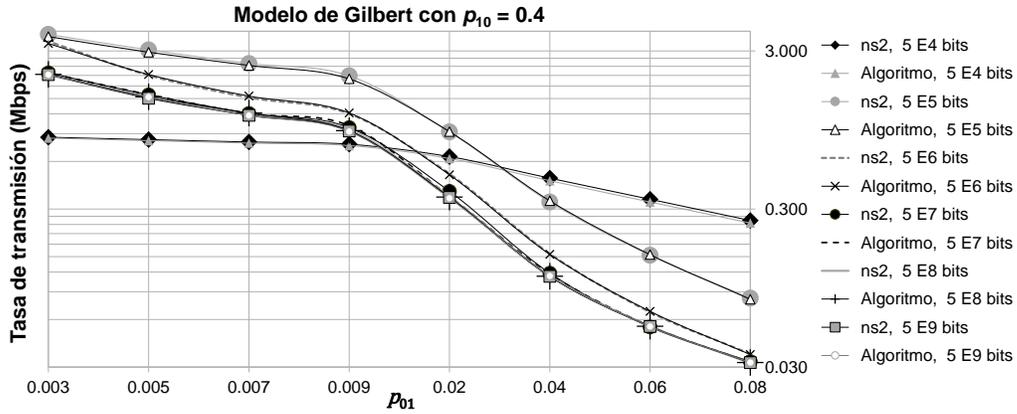


Figura 4.25: Tasa de transmisión, en Mbps, de un flujo TCP NewReno de tamaño determinado sobre una red con un modelo de pérdidas tipo Gilbert Simplificado, con  $p_{10} = 0,4$ .

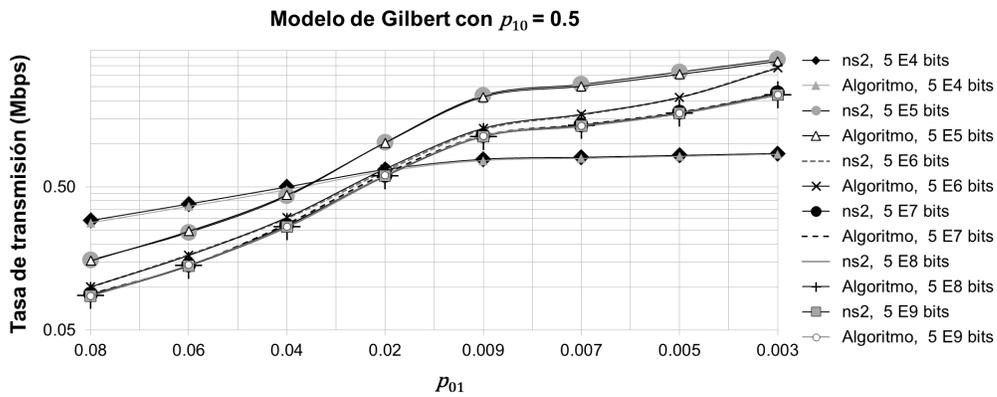


Figura 4.26: Tasa de transmisión, en Mbps, de un flujo TCP NewReno de tamaño determinado sobre una red con un modelo de pérdidas tipo Gilbert Simplificado, con  $p_{10} = 0,5$ .

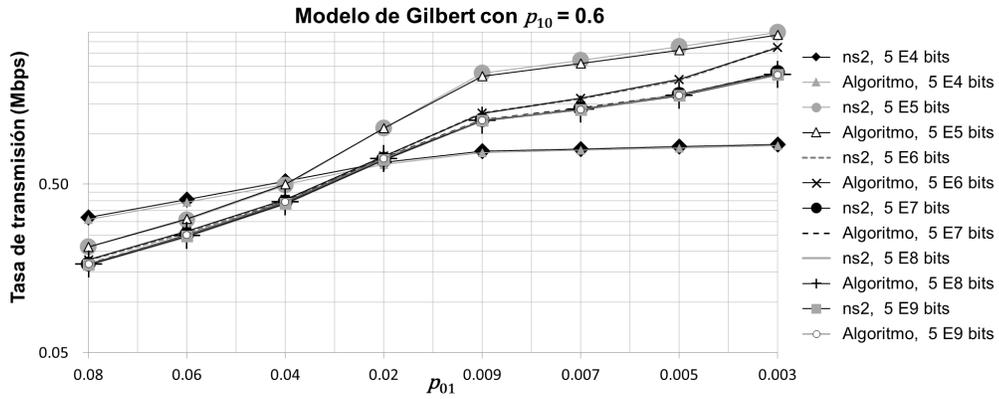


Figura 4.27: Tasa de transmisión, en Mbps, de un flujo TCP NewReno de tamaño determinado sobre una red con un modelo de pérdidas tipo Gilbert Simplificado, con  $p_{10} = 0,6$ .

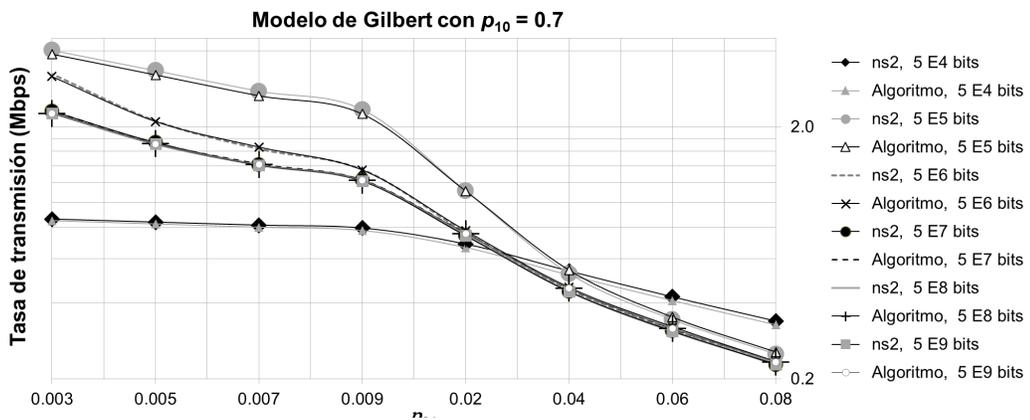


Figura 4.28: Tasa de transmisión, en Mbps, de un flujo TCP NewReno de tamaño determinado sobre una red con un modelo de pérdidas tipo Gilbert Simplificado, con  $p_{10} = 0,7$ .

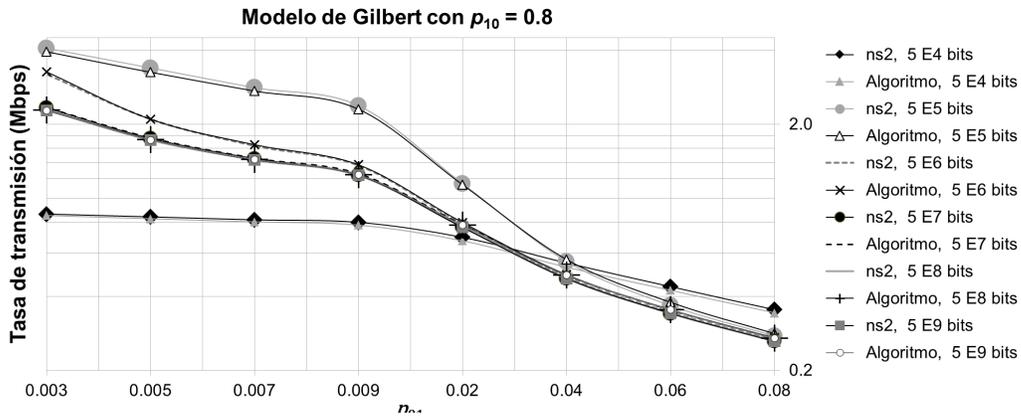


Figura 4.29: Tasa de transmisión, en Mbps, de un flujo TCP NewReno de tamaño determinado sobre una red con un modelo de pérdidas tipo Gilbert Simplificado, con  $p_{10} = 0,8$ .

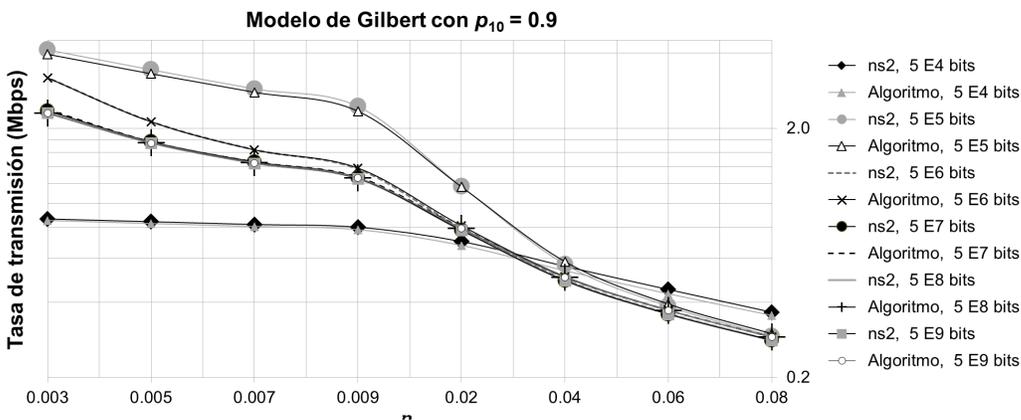


Figura 4.30: Tasa de transmisión, en Mbps, de un flujo TCP NewReno de tamaño determinado sobre una red con un modelo de pérdidas tipo Gilbert Simplificado, con  $p_{10} = 0,9$ .

## 4.5. Conclusiones del algoritmo.

En este capítulo se presenta un algoritmo como herramienta de evaluación del valor esperado del THR (y del  $THR_E$ ) de una conexión TCP NewReno en una red cableada. El algoritmo usa como parámetros de entrada el RTT promedio, la RTO promedio, el valor inicial del *ssthresh*, el tamaño del flujo TCP, el umbral máximo de la RTO, el tamaño máximo de segmento y los parámetros del modelo de pérdidas empleado para caracterizar el comportamiento de pérdidas de la red.

Entre los aspectos innovadores de esta herramienta de evaluación está su aplicación para flujos TCP NewReno de cualquier tamaño, en contraste con los modelos analíticos recientes que evalúan el THR en estado estable de una conexión de larga duración. Su importancia se debe a que los flujos cortos constituyen el mayor porcentaje de la cantidad de flujos transitando en Internet [38] [39] [47].

Este algoritmo se desarrolla como una extensión al presentado en el capítulo 3. Sin embargo, una de sus principales premisas cambia: el presente algoritmo usa el tamaño del flujo como parámetro de entrada mientras que el del capítulo 3 se enfoca únicamente a flujos de larga duración. La validación de este algoritmo se realiza mediante su comparación contra un conjunto de resultados de simulación. A diferencia del algoritmo desarrollado en el capítulo 3, el presente no es comparable con los modelos analíticos existentes por dicho cambio de premisa.

A partir de un examen minucioso de los resultados de la simulación de una serie de conexiones TCP NewReno, se obtuvo una muestra representativa de las posibilidades de operación de los mecanismos del protocolo. De dicho examen se pudo desprender el descifrado de la ejecución de los mecanismos de recuperación de pérdidas, los cuales, tanto en el algoritmo del capítulo 3 como en los modelos analíticos existentes, se representan sólo a través de su simplificación robusta. Al igual que dicho algoritmo, el presente realiza un seguimiento de la dinámica de transmisión de una conexión TCP NewReno, la cual está determinada por el comportamiento de pérdidas de la red. La diferencia con el algoritmo del capítulo 3 radica en los siguientes puntos:

- Este algoritmo incluye una representación exacta de la operación del mecanismo FR de TCP NewReno. Particularmente, se sigue con detalle la dinámica de transmisión cuando: a) alguna retransmisión en la FR falla, lo que obliga a su terminación con un TO; y b) existen pérdidas de transmisiones nuevas dentro de la FR, lo que implica la posibilidad de tener FRs sucesivas.

- El presente algoritmo usa una aproximación simple, pero adecuada, de la duración y de la cantidad de segmentos enviados durante el mecanismo de recuperación de pérdidas activado por un TO.

El algoritmo se ejecuta para calcular las tasas  $THR$  y  $THR_E$  de una conexión TCP NewReno particular de cierto tamaño. Comenzando por la fase  $SS_{ini}$ , el algoritmo va pasando a través de una secuencia de fases (cada una correspondiente a un mecanismo del protocolo) hasta que transita a la fase final, EC. La transición entre fases es guiada según la secuencia de pérdidas generada por el modelo que representa el comportamiento de pérdidas de la red. Usa un conjunto de variables globales, las cuales se van actualizando con el paso por cada fase. Entre éstas se encuentra el contador de segmentos enviados, el contador de segmentos enviados exitosamente, el SN reconocido máximo y la duración de la conexión. Una vez que el SN reconocido máximo se iguala con la cantidad de segmentos que constituyen al flujo, el algoritmo se detiene, y las tasas de transmisión,  $THR$  y  $THR_E$ , se calculan. Este procedimiento se repite una cierta cantidad de veces para obtener los promedios de ambas medidas de desempeño.

Parte de la conclusión del algoritmo del capítulo 3 se mantiene, puesto que la precisión de los resultados del presente algoritmo se debe principalmente a dos factores:

- La incorporación de una representación más cercana y detallada de los mecanismos del protocolo, en particular de los mecanismos de recuperación de pérdidas.
- La posibilidad de usar un modelo de pérdidas adecuado para caracterizar distintos escenarios de operación en la red.

Además, en comparación con la simulación como herramienta de evaluación del  $THR$ , con este algoritmo se evitan la complejidad y el consumo de tiempo y recursos asociados con ella.

El algoritmo se valida a través de su comparación con un conjunto de pruebas de simulación usando el simulador ns-2. Se consideran varios escenarios de pérdidas, todos representados a través del modelo simplificado de Gilbert, por ser un modelo ampliamente usado para representar el comportamiento de las pérdidas en redes de conmutación de paquetes [63] [70]. El error promedio correspondiente al total de los casos presentados es del 2.15 %.

Entre las principales ventajas que presenta este algoritmo como forma de evaluación del desempeño de TCP NewReno en comparación con los modelos analíticos existentes y la simulación por computadora se cuentan las siguientes:

- Constituye una forma de evaluación alternativa.
- Una representación de los mecanismos de TCP NewReno más cercana a la especificada en el estándar resulta en un error adecuadamente pequeño y significativamente menor que el arrojado por los modelos analíticos, incluyendo el más reciente y completo para este protocolo, reportado en [37].
- Permite la evaluación en cualquier escenario o ambiente de red, mediante su apropiada representación a través de un modelo de pérdidas.
- Se presenta como un marco unificado de la evaluación del desempeño de flujos TCP NewReno de cualquier tamaño.
- Con esta forma de evaluación se puede prescindir de la necesidad de conocer y manejar alguna herramienta sofisticada de simulación de redes, como el simulador ns-2.
- Adicionalmente, el consumo de tiempo y recursos que este algoritmo implica es sensiblemente menor que el requerido por un simulador de redes.

Entre las limitantes más severas que se pueden señalar de este algoritmo se tienen las siguientes:

- Su complejidad es mayor que la de un modelo analítico, pues aunque se trata de un algoritmo simple, requiere más programación que las ecuaciones de un modelo analítico. Sin embargo, dada la gran dificultad implícita en el análisis para la obtención de un modelo analítico, se justifica este pequeño incremento de la complejidad en el uso de esta forma alternativa de evaluación.
- Al ser una evaluación numérica, se requiere de la ejecución de un número adecuado de repeticiones para obtener el promedio de la medida de desempeño.



# Conclusiones generales y trabajo a futuro.

El presente trabajo doctoral se centró en el estudio del protocolo TCP NewReno y en particular en la evaluación de su desempeño a través de su tasa de transmisión (o tasa de transmisión efectiva). La importancia de este protocolo radica en que es en una de las versiones de TCP más ampliamente usadas en Internet, y se prevé que seguirá vigente durante varios años más debido al predominio actual y a la expansión tanto de los servicios como de los usuarios de aplicaciones basadas en la pila de protocolos TCP/IP.

El trabajo comienza con una revisión del protocolo. TCP NewReno se desarrolla como una modificación al protocolo TCP Reno, el cual a su vez es una mejora de TCP Tahoe. La descripción de la operación de todos ellos se centra en la especificación de los distintos mecanismos que los constituyen. Su enunciación, que puede encontrarse directamente en los RFCs correspondientes, es simple y relativamente pequeña, sin embargo, lleva implícita una operación cuya complejidad aumenta en gran medida. En parte por ello, existe una gran cantidad de fuentes adicionales que presentan una explicación mucho más didáctica de los mismos. Sin embargo, hasta donde sabemos, en todas ellas siempre se omiten algunos detalles del protocolo así como la exposición de las distintas posibilidades de operación de los mecanismos, de nuevo, dado el nivel de complejidad que ésta puede alcanzar.

Posteriormente se realizó una revisión bibliográfica sobre el tema de la evaluación del desempeño del protocolo TCP en redes cableadas. Las propuestas existentes consisten en modelos analíticos. El desarrollo histórico de los modelos comienza con aquellos que realizan grandes simplificaciones de los mecanismos de TCP, y continúa con las propuestas de modelos que poco a poco van incorporando en su análisis la representación menos simplificada, y por lo tanto, más cercana del protocolo. El contexto histórico termina con el modelo más reciente, y también más completo, para TCP NewReno, en cuyo tratamiento aún se encuentran varias suposiciones sobre los mecanismos de recuperación de pérdidas. Éstas le otorgan un nivel de complejidad manejable, pero repercuten en un error de la estimación del desempeño inacepta-

ble para algunos escenarios de red, es decir, para ciertas características del comportamiento de las pérdidas en la red. La mayoría de estos trabajos de modelado consideran que las pérdidas de los segmentos en la red son independientes, lo cual también constituye una simplificación de la realidad, pues existe otra rama de estudio enfocada al modelado del comportamiento de pérdidas de las redes de conmutación de paquetes en la que se tiene demostrado que los modelos que representan más cercanamente los escenarios reales de red presentan memoria [63] [70].

La validación de los modelos analíticos suele realizarse a través de su comparación ya sea con los resultados obtenidos de la simulación de la operación en escenarios de red, o de mediciones realizadas en Internet o en redes particulares. La plataforma de simulación que más se emplea para este fin es ns2, por ser una plataforma abierta para su desarrollo y constante ampliación.

Como resultado del trabajo de investigación se encontró que, a pesar de la gran cantidad de trabajo en el desarrollo de herramientas analíticas para la evaluación del desempeño de este protocolo, éste sigue siendo un problema abierto, principalmente debido a dos factores:

- El enfoque prevalente sobre flujos de larga duración, que si bien constituyen gran parte del volumen del tráfico en Internet, no son el mayor porcentaje de la cantidad de flujos que transitan en ella [38] [39] [47].
- El error que las aproximaciones actuales producen en ciertos escenarios de operación, a causa de las simplificaciones que se realizan en los análisis tanto de los mecanismos del protocolo como de las características del comportamiento de pérdidas de la red.

En este trabajo doctoral se desarrollan dos herramientas de evaluación del desempeño de TCP NewReno en redes cableadas con la finalidad de cubrir los aspectos anteriores. Buscando eliminar las simplificaciones concernientes a los mecanismos del protocolo se recurrió al estudio detallado de las características del protocolo así como al examen minucioso de los resultados de una extensa gama de pruebas de simulación. Lo anterior permitió analizar las distintas condiciones de operación de los mecanismos de este protocolo, y por lo tanto, detectar aquellas que faltaban en los modelos existentes. Como una primera solución se plantea en el capítulo 3 una herramienta de evaluación que cubre parte de las condiciones ausentes en los modelos actuales: en particular incorpora al análisis todos los posibles casos de operación de los mecanismos SS y CA, aunque mantiene la mayoría de las suposiciones sobre los mecanismos de recuperación de pérdidas que se plantean en dichos modelos, incluyendo al

más completo presentado en [37]. Sin embargo, dado que la incorporación de estos aspectos del protocolo rompe con la suposición fundamental del método de análisis usado en los últimos modelos propuestos, para el desarrollo de esta nueva propuesta de evaluación es forzoso utilizar un método alternativo que permita la integración de todas las posibilidades de operación de estos mecanismos sin el costo de tener un nivel alto de complejidad. Por ello, en el capítulo 3 se propone un algoritmo para la evaluación del desempeño de una conexión TCP NewReno de larga duración, a través de su tasa de transmisión en estado estable. Adicionalmente, y considerando el segundo de los factores arriba mencionados, el algoritmo incorpora el uso de un modelo del comportamiento de pérdidas de la red, el cual puede escogerse para caracterizar cualquier ambiente de red.

Como una extensión del algoritmo del capítulo 3, en el capítulo 4 se desarrolla un nuevo algoritmo con el que se ofrece una solución total al problema abierto anteriormente descrito, a través de la cobertura de los siguientes aspectos:

- Es un algoritmo que sirve para evaluar el desempeño de una conexión TCP NewReno de cierto tamaño, a través de su tasa de transmisión promedio. Específicamente, el tamaño del flujo TCP es uno de los parámetros de entrada del algoritmo.
- Incorpora dentro del algoritmo el uso de un modelo del comportamiento de pérdidas de la red, el cual puede escogerse para caracterizar cualquier ambiente de red.
- Introduce la representación completa de los mecanismos SS y CA de TCP NewReno.
- Incluye una representación exacta de la dinámica de transmisión del protocolo cuando ocurre una recuperación de pérdidas a través del mecanismo FR.
- Introduce una aproximación simple, pero adecuada, de la cantidad de segmentos transmitida y el tiempo usado para ello en un evento de recuperación de pérdidas activada por un TO.

La herramienta de evaluación resultante se prueba contra los resultados de un conjunto de pruebas de la simulación de diversos escenarios de red, caracterizados todos ellos mediante un modelo simplificado de Gilbert. Se escoge este modelo por ser uno de los más ampliamente usados para modelar el proceso de pérdidas en una red de conmutación de paquetes. Sin embargo, el algoritmo permite estudiar el desempeño de TCP NewReno en cualquier ambiente de operación si éste se representa a través de un modelo particular que se incluya como parte del algoritmo. Considerando todos los casos de prueba, el error promedio arrojado fue

del 2.15 %.

Por lo tanto, este algoritmo se presenta como una herramienta de evaluación alternativa a la simulación por computadora, de forma que usándolo no es necesario manejar ninguna plataforma de simulación sofisticada ni se requiere del consumo de tiempo de ejecución ni de los recursos que ésta implica. Por otro lado, es cierto que la complejidad de su utilización supera a la de un modelo analítico, sin embargo, la generalidad de su aplicación así como su error significativamente menor la justifica.

Como trabajo a futuro queda pendiente ampliar este trabajo de investigación que se enfoca al desempeño de TCP en redes cableadas para abarcar nuevos y cada vez más frecuentes ambientes de operación en los que las condiciones sobre las que trabaja el protocolo cambian radicalmente, por lo que afectan directamente su desempeño. Un ejemplo de otro tipo de escenario de comunicaciones cuyas características respecto de una red cableada cambian drásticamente es el ambiente sobre redes *ad-hoc*. En éstas, el cambio dinámico de la topología propicia condiciones de operación distintas a las que prevalecen en una red cableada, en la cual la topología se puede suponer fija a lo largo de una conexión. Entre los factores afectados por este cambio de ambiente está el RTT, el cual deja de mantenerse relativamente estable. Dado que TCP basa su operación en este parámetro, en el análisis de TCP sobre redes cableadas, ésta es una premisa. De manera que el análisis de TCP para este otro ambiente puede cambiar fundamentalmente.

# Glosario.

ACK	Reconocimiento positivo en el control de errores de TCP.
AIMD	Técnica usada en el control de congestión de TCP conocida como de incremento aditivo/decremento multiplicativo ( <i>Additive Increase/Multiplicative Decrease</i> ).
CA	“Evasión de la congestión” ( <i>Congestion Avoidance</i> ) en TCP.
<i>cwnd</i>	Ventana de congestión en el transmisor TCP.
DA	ACK duplicado ( <i>duplicate ACK</i> ), es decir, que no reconoce nuevos datos.
<i>E</i>	Conteo de LEs en el flujo TCP NewReno de estudio en el intervalo de observación.
Evento de pérdida	Uno o más segmentos perdidos dentro de la misma ventana de transmisión.
FC	Fase “fin de conexión” TCP, que es la fase en donde termina toda conexión.
FR	“Recuperación rápida” ( <i>Fast Recovery</i> ) en TCP Reno y NewReno.
<i>iwnd</i>	Valor inicial de la <i>cwnd</i> .
LE	Evento de pérdida ( <i>Loss Event</i> ).
LW	Ventana de pérdida ( <i>Loss Window</i> ).
<i>lwnd</i>	Valor de la ventana de pérdida en TCP.

MSS	Tamaño máximo de segmento ( <i>Maximum Segment Size</i> ).
PA	RA parcial ( <i>Partial ACK</i> ), es decir, RA que reconoce nuevos datos estando en la FR, pero con SN menor que la variable <i>recover</i> .
RA	ACK que reconoce nuevos datos ( <i>Regular ACK</i> ).
RTO	Duración del temporizador de retransmisión ( <i>Retransmission TimeOut duration</i> ) en TCP.
RTT	Tiempo de “ida y vuelta” (RTT: <i>Round-Trip Time</i> ) de un segmento y su ACK en TCP.
<i>rwnd</i>	Ventana deslizante en el receptor TCP.
SACK	ACK selectivo ( <i>Selective ACK</i> ).
SN	Número de secuencia ( <i>Sequence Number</i> ) con el que se distingue un segmento en TCP.
SS	“Comienzo lento” ( <i>Slow Start</i> ) en TCP.
SSini	Fase SS inicial de cualquier conexión TCP.
<i>ssthresh</i>	Umbral del SS.
TA	Total de segmentos enviados exitosamente en un flujo TCP simulado durante un intervalo de observación ( <i>Total Acknowledged</i> ).
TCP	Protocolo de control de transporte ( <i>Transport Control Protocol</i> ).
TD	Tres DAs iguales y consecutivos ( <i>Triple Duplicate ACK</i> ).
THR	Tasa (o tasa efectiva) de transmisión ( <i>THRoughput</i> ).
TL	Total de segmentos perdidos en un flujo TCP simulado durante un intervalo de observación ( <i>Total Loss</i> ).
TS	Total de segmentos enviados en un flujo TCP simulado durante un intervalo de observación ( <i>Total Send</i> ).
TO	Evento de expiración del temporizador de retransmisión ( <i>TimeOut event</i> ) para el control de errores en TCP.
Ventana de transmisión	Segmentos enviados dentro del mismo RTT que, bajo las suposiciones comunes, corresponde con la emisión en ráfaga de los segmentos de la <i>cwnd</i>

# Referencias.

- [1] J. Postel, "Transmission control protocol", RFC 793. Septiembre, 1980.
- [2] J. Postel, "The TCP maximum segment size and related topics", RFC 879. Noviembre, 1983.
- [3] David D. Clark, "Window and acknowledgement strategy in TCP", RFC 813. Julio, 1982.
- [4] John Nagle, "Congestion control in IP/TCP internetworks", RFC 896. Enero, 1984.
- [5] Braden, R., Requirements for internet hosts communication layers", RFC 1122. Octubre, 1989.
- [6] J C Mogul, S E Deering, "Path MTU Discovery", RFC 1191. Noviembre, 1990.
- [7] Jacobson, V., Braden, R., D. Borman, "TCP Extensions for High Performance", RFC 1323. Mayo, 1992.
- [8] Mathis, M., Mahdavi, J., Floyd, S., A. Romanow, "TCP Selective Acknowledgment Options", RFC 2008. Octubre, 1996.
- [9] Floyd, S., Mahdavi, J., Mathis, M. y M. Podolsky, "An Extension to the Selective Acknowledgement (SACK) Option for TCP", RFC 2883. Julio, 2000.
- [10] K Ramakrishnan, S Floyd, D Black, "The addition of explicit congestion (ECN) to IP", RFC 3168. Septiembre, 2001.
- [11] V. Paxson, M. Allman, "Computing TCP's retransmission timer", RFC 2988, Noviembre, 2000.
- [12] Allman, M., Floyd, S. y C. Partridge, "Increasing TCP's Initial Window", RFC 3390. Octubre, 2002.
- [13] Blanton, E., Allman, M., Fall, K. y L. Wang, "A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP", RFC 3517. Abril, 2003.

- [14] S. Floyd, T. Henderson y A. Gurtov, “The NewReno modification to TCP’s fast recovery algorithm”, RFC 3782. Abril, 2004.
- [15] M. Allman, V. Paxson y E. Blanton, “TCP congestion control”, RFC 5681. Septiembre, 2009.
- [16] M. Duke, R. Braden, W. Eddy, E. Blanton, “A Roadmap for Transmission Control Protocol (TCP) Specification Documents”, RFC 4614. Septiembre, 2006.
- [17] R. Dunaytsev y D. Moltchanov, “TCP performance modeling in wired and wireless networks. Single source models”, *LAP Lambert Academic Publishing*, 2011.
- [18] Y. Tian, K. Xu y N. Ansari, “TCP in wireless environments: problems and solutions”, *IEEE Communications Magazine*, Vol. 43, No. 3, pp. S27-S32. Marzo, 2005.
- [19] A. Afanasyev, N. Tilley, P. Reiher y L. Kleinrock, “Host-to-host congestion control for TCP”, *IEEE Communications Surveys & Tutorials*, Vol. 12, No. 3, 2010, pp. 304-342.
- [20] C. Lochert, B. Scheuermann y M. Mauve, “A survey on congestion control for mobile ad hoc networks”, *Wireless Communications and Mobile Computing*, Vol. 7, No. 5, p. 655, 2007.
- [21] A. Medina, M. Allman y S. Floyd, “Measuring the evolution of transport protocols in the internet”, *Comput. Commun. Rev.*, Vol. 35, No. 2, pp. 37–51. Abril, 2005.
- [22] V. Jacobson, “Congestion avoidance and control”, *ACM SIGCOMM*, 1988, pp. 314–329.
- [23] W. Stevens, “TCP slow start, congestion avoidance, fast retransmit”, RFC 2001, 1997.
- [24] M. Allman, V. Paxson y W. Stevens, “TCP congestion control”, RFC 2581, 1999.
- [25] S. Floyd y T. Henderson, “The NewReno modification to TCP’s fast recovery algorithm”, RFC 2582, 1999.
- [26] K. Fall y S. Floyd, “Simulation-based comparisons of Tahoe, Reno, and SACK TCP”, *ACM Computer Communication Review*, Vol. 26, No. 3, pp. 5-21, 1996.
- [27] V. Paxson, “Empirically derived analytic models of wide-area TCP connections”, *IEEE/ACM Trans. Netw.*, Vol. 2, No. 4, pp. 316-336. Agosto, 1994.
- [28] M. Mathis, J. Semke, J. Mahdavi y T. Ott. “The macroscopic behavior of the TCP congestion avoidance algorithm”, *ACM Computer Communication Review*, Vol. 27, No. 3, pp. 67-82. Julio, 1997.
- [29] J. Heidemann, K. Obraczka y J. Touch, “Modeling the performance of HTTP over seve-

ral transport protocols”, *IEEE/ACM Trans. Netw.*, Vol. 5, No. 5, pp. 616-630. Octubre, 1997.

[30] T. Lakshman y U. Madhow, “The performance of TCP/IP for networks with high bandwidth-delay products and random loss”, *IEEE/ACM Trans. Netw.*, Vol. 5, No. 3, pp. 336-350. Junio, 1997.

[31] C. Joo y S. Bahk, “Start-up transition behavior of TCP NewReno”, *Electronics Letters*, Vol. 35, No. 21, pp. 1818-1820. Octubre, 1999.

[32] J. Padhye, V. Firoiu, D. Towsley y J. Kurose, “Modeling TCP Reno performance: a simple model and its empirical validation”, *IEEE/ACM Trans. Netw.*, Vol. 8, No. 2, pp. 133-145. Abril, 2000.

[33] N. Cardwell, S. Savage y T. Anderson, “Modeling TCP latency”, *Proc. of IEEE INFOCOM*, 2000, pp. 1742-1751.

[34] M. Goyal, R. Guerin y R. Rajan, “Predicting TCP throughput from non-invasive network sampling”, *Proc. of IEEE INFOCOM*, 2002, pp. 180-189.

[35] W. John y S. Tafvelin, “Analysis of internet backbone traffic and header anomalies observed”, *Proc. of ACM SIGCOMM conference on Internet measurement*, 2007, pp. 111-116.

[36] R. Dunaytsev, Y. Koucheryavy y J. Harju, “The PFTK-model revised”, *Computer Communications*, Vol. 29, No. 13-14, pp. 2671-2679. Agosto, 2006.

[37] N. Parvez, A. Mahanti y C. Williamson, “An analytic throughput model for TCP NewReno”, *IEEE/ACM Trans. Netw.*, Vol. 18, No. 2, pp. 448-461. Abril, 2010.

[38] M.-S. Kim, Y. Won y J. Hong, “Characteristic analysis of internet traffic from the perspective of flows”, *Computer Communications*, Vol. 29, No. 10, pp. 1639-1652. Junio, 2006.

[39] F. Qian, A. Gerber, Z. Mao, S. Sen, O. Spatscheck y W. Willinger, “TCP revisited: a fresh look at TCP in the wild”, *Proc. of ACM SIGCOMM conference on Internet measurement*, 2009, pp. 76-89.

[40] S. Floyd y V. Paxson, “Difficulties in simulating the Internet”, *IEEE/ACM Trans. Netw.*, Vol. 9, No. 4, pp. 392-403. Agosto, 2001.

[41] K. Thompson, G. Miller, y Rick Wilder, “Wide-area internet traffic patterns and characteristics”, *Network, IEEE*, Vol. 11, No. 6, pp. 10-23. Noviembre, 1997.

[42] N. Brownlee, “Internet stream size distributions”, *Proc. of ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, 2002, pp. 282-283.

- [43] D. Zheng, G. Lazarou y R. Hu, “A stochastic model for short-lived TCP flows”, *ICC*, Vol. 1, pp. 76-81. Mayo, 2003.
- [44] B. Sikdar, S. Kalyanaraman y K. Vastola, “Analytic models for the latency and steady-state throughput of TCP Tahoe, Reno, and SACK”, *IEEE/ACM Trans. Netw.*, Vol. 11, No. 6, pp. 959-971. Diciembre, 2003.
- [45] E. Altman, K. Avrachenkov y C. Barakat, “A stochastic model of TCP/IP with stationary random losses”, *IEEE/ACM Trans. Netw.*, Vol. 13, No. 2, pp. 356-369. Abril, 2005.
- [46] A. Wierman, T. Osogami, “A unified framework for modeling TCP-Vegas, TCP-SACK, and TCP-Reno”, *IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems*, 2003, pp. 269-278.
- [47] P. Hurtig, W. John y A. Brunstrom, “Recent trends in TCP packet-level characteristics”, *ICNS*, Venecia, Italia, 2011, pp. 49-56.
- [48] N. Parvez, A. Mahanti y C. Williamson, “TCP NewReno: slow-but-steady or impatient?”, *ICC*, 2006, pp. 716-722.
- [49] R. Dunaytsev, K. Avrachenkov, Y. Koucheryavy y J. Harju, “An analytical comparison of the slow-but-steady and impatient variants of TCP New Reno”, *Proc. WWIC*, Portugal, 2007, pp. 30-42.
- [50] R. Dunaytsev, Y. Koucheryavy y J. Harju, “TCP NewReno throughput in the presence of correlated losses: the slow-but-steady variant”, *ICCC*, 2006, pp. 1-6.
- [51] [51] X. Fu, W. Yu, S. Jiang, S. Graham y Y. Guan, “TCP performance in flow-based mix networks: modeling and analysis”, *IEEE Transactions on parallel and distributed systems*, Vol. 20, No. 5, pp. 695-709. Mayo, 2009.
- [52] J.-H. Hwang y C. Yoo, “Formula-based TCP throughput prediction with available bandwidth”, *IEEE Communications Letters*, Vol. 14, No. 4, pp. 363-365. Abril, 2010.
- [53] M. Mirza, J. Sommers, P. Barford y X. Zhu, “A machine learning approach to TCP throughput prediction”, *IEEE/ACM Trans. Netw.*, Vol. 18, No. 4, pp. 1026-1039. Agosto, 2010.
- [54] H. Tijms, “Stochastic models”, *Ed. Wiley*, 2003.
- [55] The NS Project. The Network Simulator: ns-2. <http://www.isi.edu/nsnam/ns>.
- [56] Código de ns-2 para el modelo en [37]. <http://www.cpsc.ucalgary.ca/carey/software.html>.
- [57] G. Carofiglio y Luca Muscariello, “On the impact of TCP and per-flow scheduling on

internet performance”, *IEEE/ACM Trans. Netw.*, Vol. 20, No. 2, pp. 620-633. Abril, 2012.

[58] L. Budzisz, R. Stanojevic, A. Schlote, F. Baker y Robert Shorten, “On the fair coexistence of loss- and delay-based TCP”, *IEEE/ACM Trans. Netw.*, Vol. 19, No. 6, pp. 1811 - 1824. Diciembre, 2011.

[59] N. Arianpoo, P. Jokar, y V. Leung, “Enhancing TCP performance in wireless mesh networks by cross layer design”, *International Conference on Computing, Networking and Communications*, 2012, pp. 177-181.

[60] F. Ren y C. Lin, “Modeling and improving TCP performance over cellular link with variable bandwidth”, *IEEE Trans. Mobile Computing*, Vol. 10, No. 8, pp. 1057-1079. Agosto, 2011.

[61] S. Floyd, T. Henderson y A. Gurtov, “The NewReno modification to TCP’s fast recovery algorithm”, RFC 6582. Abril, 2012.

[62] K.-C. Lan, J. Heidemann, “A measurement study of correlations of internet flow characteristics”, *Computer Networks*, Vol. 50, No. 1. Enero 2006.

[63] X. Yu, J. Modestino y X. Tian, “The accuracy of Markov chain models in predicting packet-loss statistics for a single multiplexer”, *IEEE Trans. Inf. Theory*, Vol. 54, No. 1, pp. 489-501. Enero 2008.

[64] E. N. Gilbert, “Capacity of a burst-noise channel”, *Bell System Technical J.*, Vol. 39, No. 5, pp. 1253-1265. Septiembre, 1960.

[65] E. O. Elliott, “A model of the switched telephone network for data communications”, *Bell System Technical J.*, Vol. 44, pp. 89-109. Enero, 1963.

[66] H. Sanneck, G. Carle, and R. Koodli, “A framework model for packet loss metrics based on loss run length”, *Proc. SPIE/ACM SIGMM Multimedia Computi. Netw. Conf.*, 2000, pp. 1-10.

[67] Z. Li, J. Chakareski, X. Niu, Y. Zhang y W. Gu, “Modeling and analysis of distortion caused by Markov-model burst packet losses in video transmission”, *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 19, No. 7. Julio 2009.

[68] P. Almstrom, M. Rabi y M. Johansson, “Networked state estimation over a Gilbert-Elliott type channel”, *IEEE Conference on Decision and Control / Chinese Control Conference Shanghai*, R.P. de China, 2009.

[69] I. Cidon, A. Khamisy y M. Sidi, “Analysis of packet loss processes in high-speed networks”, *IEEE Trans. Inf. Theory*, Vol. 39, No. 1, Enero 1993.

- [70] O. Hohlfeld, O., R. Geib y G. Hasslinger, “Packet loss in real-time services: markovian models generating QoE impairments”, *Workshop on Quality of Service*, 2008.
- [71] G. Hasslinger y O. Hohlfeld, “The Gilbert-Elliott model for packet loss in real time services on the internet”, *GI/ITG Conference Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB)*, 2008.
- [72] W. Leland, M. Taqqu, W. Willinger y Daniel V. Wilson, “On the self-similar nature of Ethernet traffic”, *IEEE/ACM Trans. Netw.*, Vol. 2. No. 1, Febrero, 1994.
- [73] K. Cardoso y J. Rezende, “Accurate hidden Markov modeling of packet losses in indoor 802.11 networks”, *IEEE Communications Letters*, Vol. 13, No. 6. Junio, 2009.
- [74] M. Yajnik, S. Moon, J. Kurose y D. Towsley, “Measurement and modeling of the temporal dependence in packet loss”, *Proc. IEEE INFOCOM*, USA, 1999, pp. 345-352.
- [75] Gallager, R. G., “Discrete Stochastic Processes”, *Kluwer*, Boston, 1996.