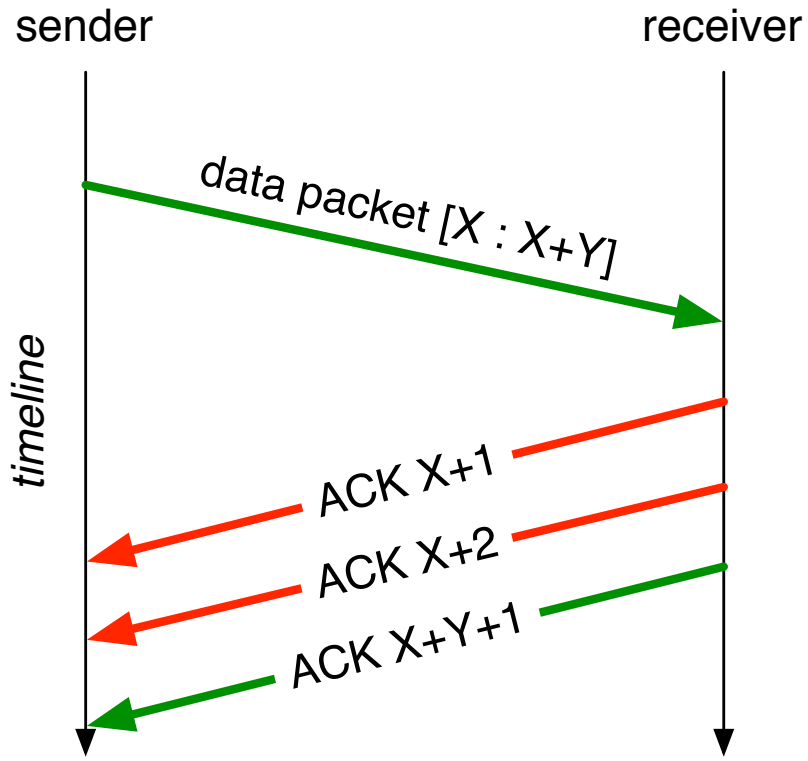
TCP ACK DIVISION REVISITED

A. Arcia-Moret, N. Montavont, J-M Bonnin, D. Ros

Introducción

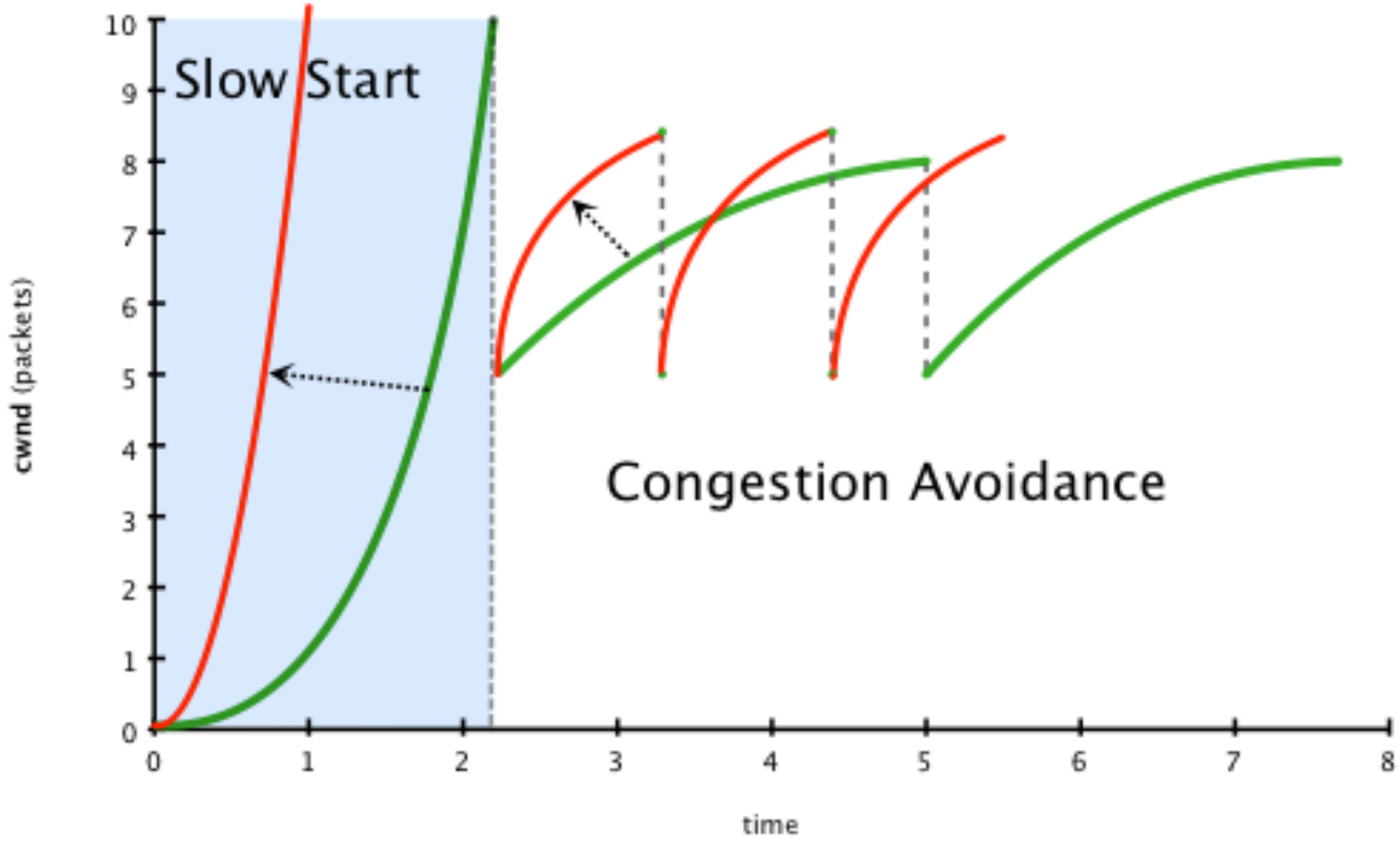
- El protocolo TCP (Transmission Control Protocol) transporta el **90%** de las comunicaciones en Internet.
  - ¡180.000 Terabytes por segundo! (Ref. British Telecom)
- Gran cantidad de tecnologías a enfrentar.
  - Redes de sensores, vehiculares, satelitales, etc.
- TCP descansa sobre el principio de ACK Clocking para efectuar una transmisión.

Principio de Ack Clocking en TCP

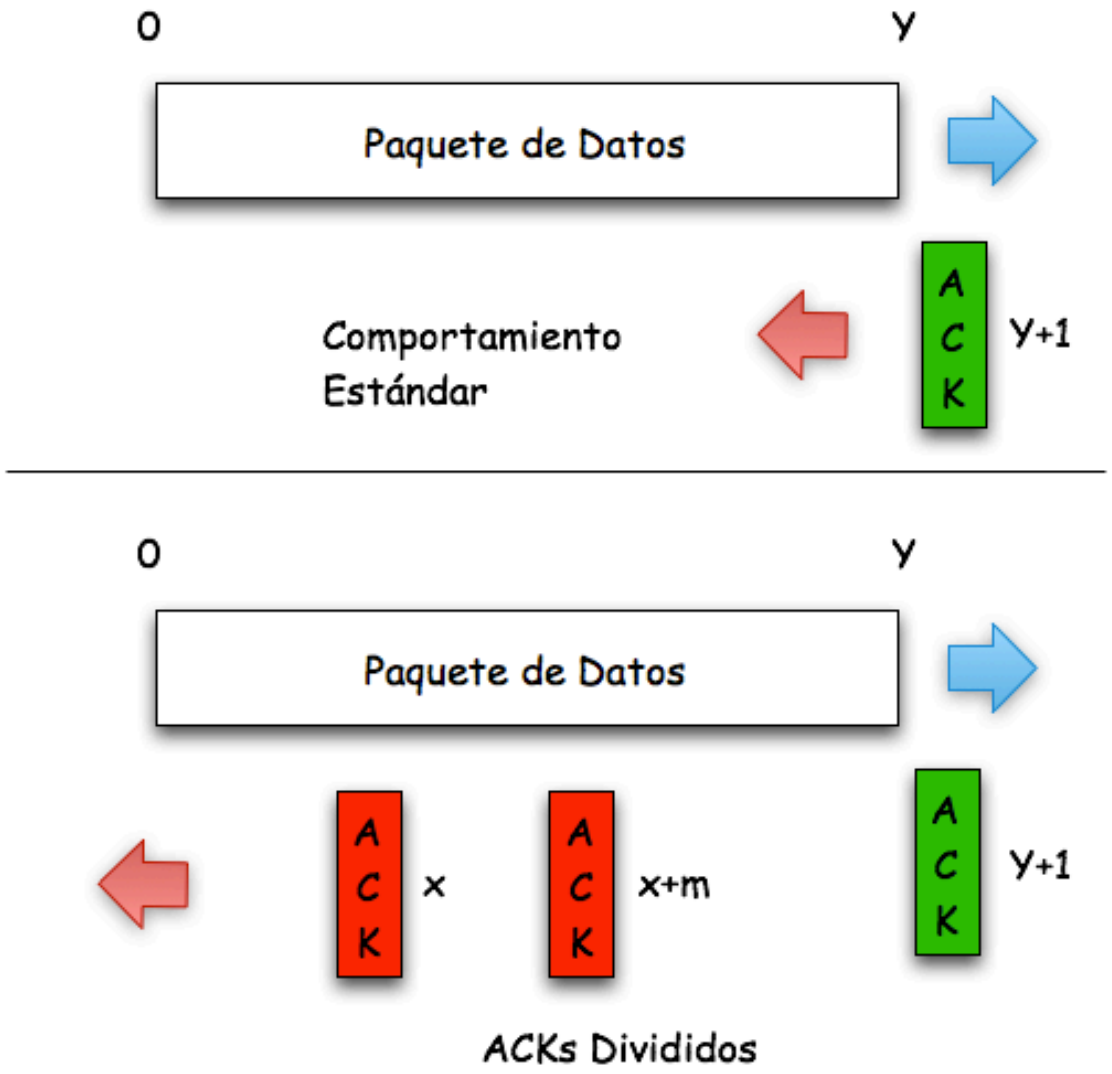


- Los ACKs informan al emisor de la llegada de paquetes de datos.
- Un ACK cubre una cantidad de datos recibidos e informa la variación futura del tráfico.
- Frecuencia de ACKs puede variar [RFC 5690], pero hay lineamientos estándares [RFC 791].

Efecto en la CWND al Acelerar el Clocking



Otra perspectiva de los divacks.



Estado del Arte

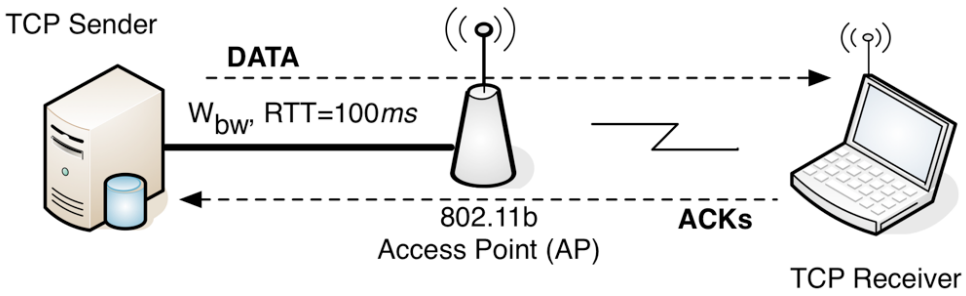
- Savage et al. anunciaron por primera vez los efectos del uso excesivo de ACKs para obtener más ancho de banda.
  - ACKs divididos —› *divacks*
  - ACKs duplicados en exceso
  - ACKs optimistas. (adelantados)
- Allman propone corrección estándar propuesta en el IETF [RFC 3465]
  - Conteo apropiado de bytes en vez de paquetes
  - Modificación mayor al algoritmo de control de congestión
- Uso benéfico en redes heterogéneas
  - Recuperación de ventana por pérdidas aleatorias
  - Ganancia rápida de ancho de banda en handovers

Propuesta

- Una reconsideración del impacto de la frecuencia aumentada de los ACKs
  - Mitigación de los divacks:
    - En medios de acceso compartidos.
    - Interacción con el algoritmo de Nagle.
  - Resultados Sorpresivos:
    - Una evaluación extendida inédita del impacto de los *divacks* en conexiones sensibles (alambradas) donde se muestra que no siempre se gana en el ataque.
    - La ganancia no es significativamente mayor que la versión más rápida de TCP.

Protección ofrecida por la MAC

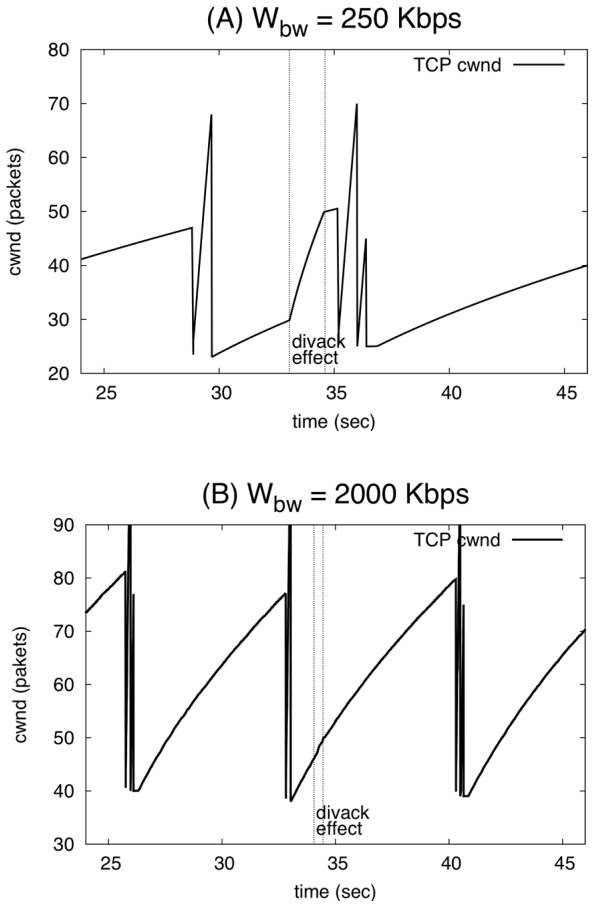
Topología



Experimentación

- Envío de un flujo largo
- Impacto de los divacks en congestion avoidance.
- Efecto de divacks sensible a la ubicación del cuello de botella.

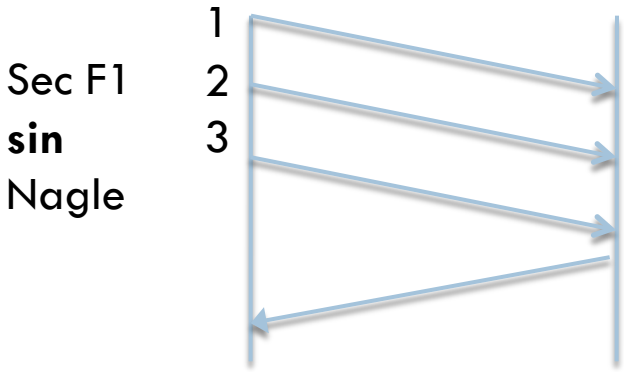
Efecto del cuello de botella

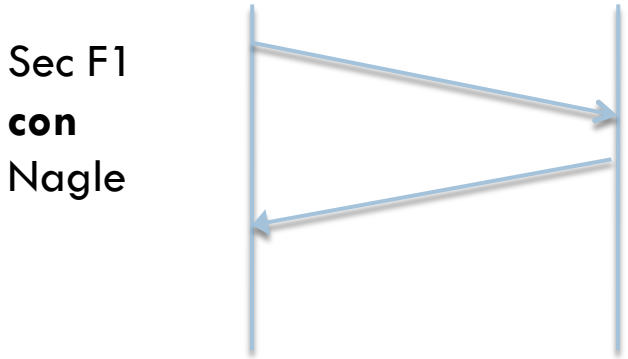


- Efecto del CSMA/CA con backoff aleatorio en la transmisión.
- 10 divacks por paquete ordenado (80 en total)
- En (A) los ACKs toman ventaja de la lentitud de los paquetes de datos.
- En (B) se observa un efecto ping-pong entre data packets y ACKs: La llamada **"auto-protección"**.

Interacción con el Algoritmo Nagle

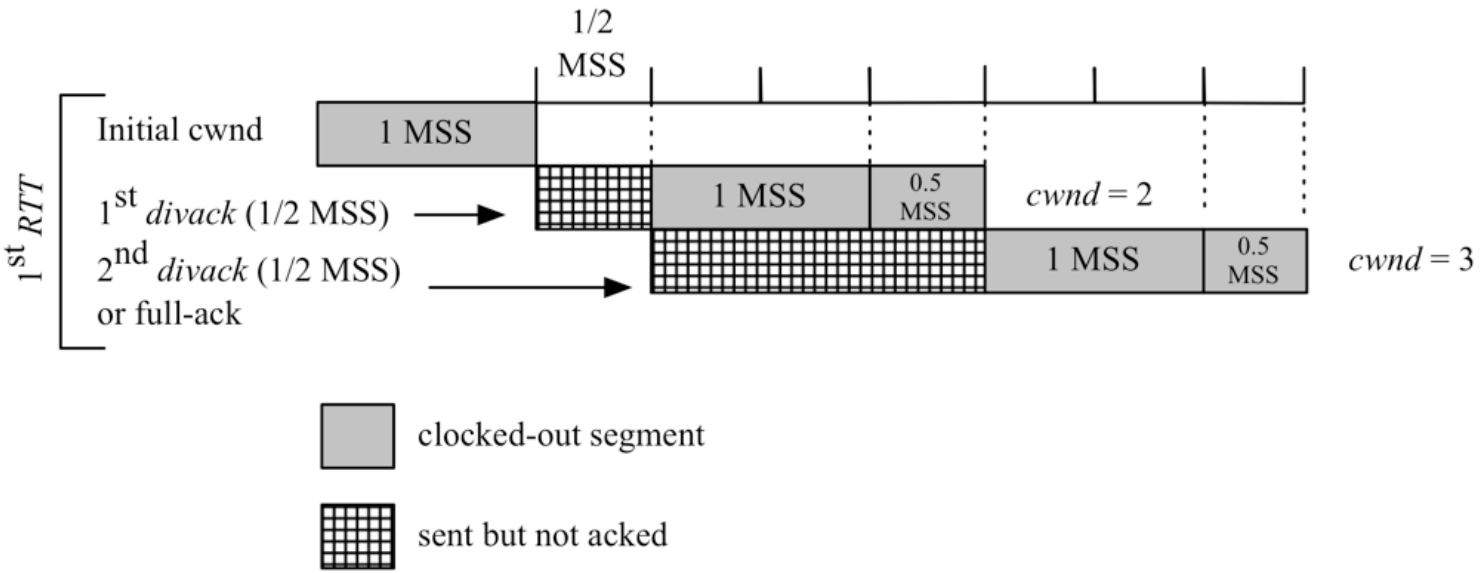
- **Nagle** propuesto para evitar envíos masivos de pequeños paquetes (Ej: conexiones interactivas a gran distancia)
  - Se retarda **un poco** la emisión de paquetes de datos para agrupar información.
  - Es una **pequeña protección** contra el uso de *divacks*.
    - Evitaría el envío masivo de "micro-paquetes".





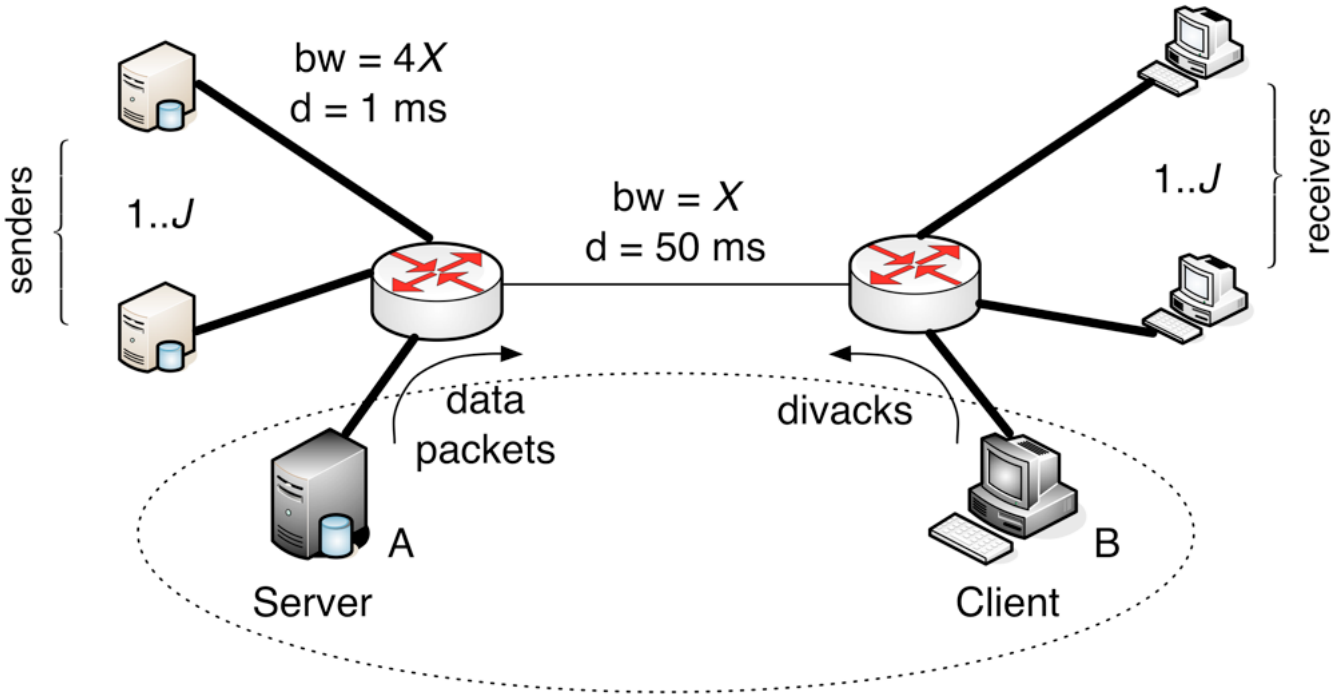
Interacción con el Algoritmo Nagle

- Nagle puede ser conveniente desactivarlo para escenarios donde la demora del último paquete sea importante (Ej: servidores de archivos pequeños).
- Divacks genera *4* segmentos (el doble de 1 apdp)



Evaluación en Presencia de Congestión

- ¿Representa Divacks realmente un problema?
- Evaluación en presencia de congestión:
  - X=10 Mbps, Buffers = 83 Pkts (Bandwidth-Delay Product)



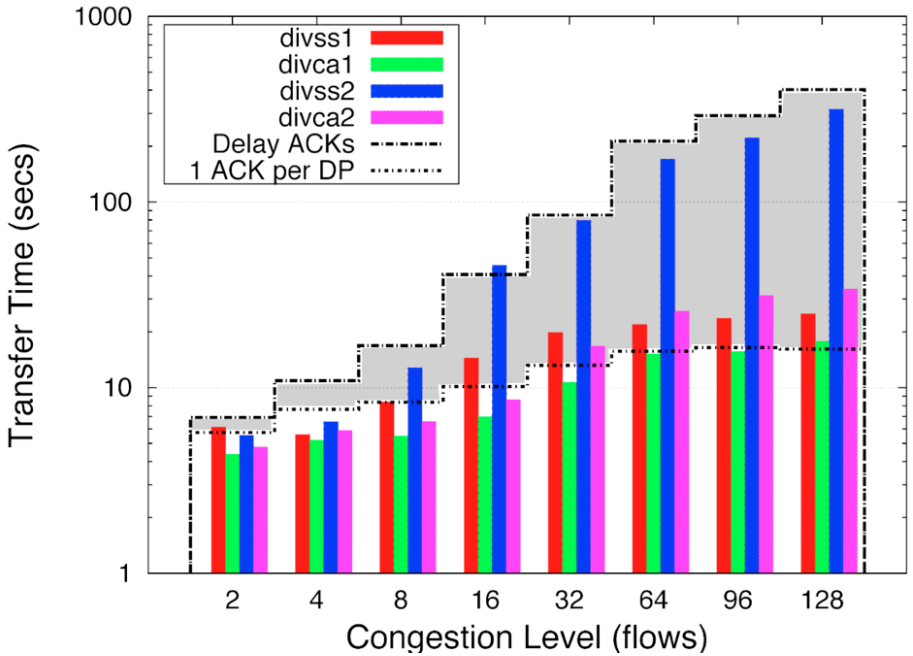
Métodos para evaluar la técnica de divacks.

Set	Policy	ACK division activated when the sender is in	Number of divacks sent when ACK division is activated	ACK sending mechanism when ACK division is deactivated
I	*divss1*	SS	m for every in-order data packet ($r = m$)	One ACK per in-order data packet ($r = 1$)
	divca1	CA		
	divssca1	SS+CA		
II	*divss2*	SS	m every other in-order data packet ($r = m/2$)	One ACK every other in-order data packet ($r = 0.5$)
	divca2	CA		
	divssca2	SS+CA		

Evaluación en Presencia de Congestión

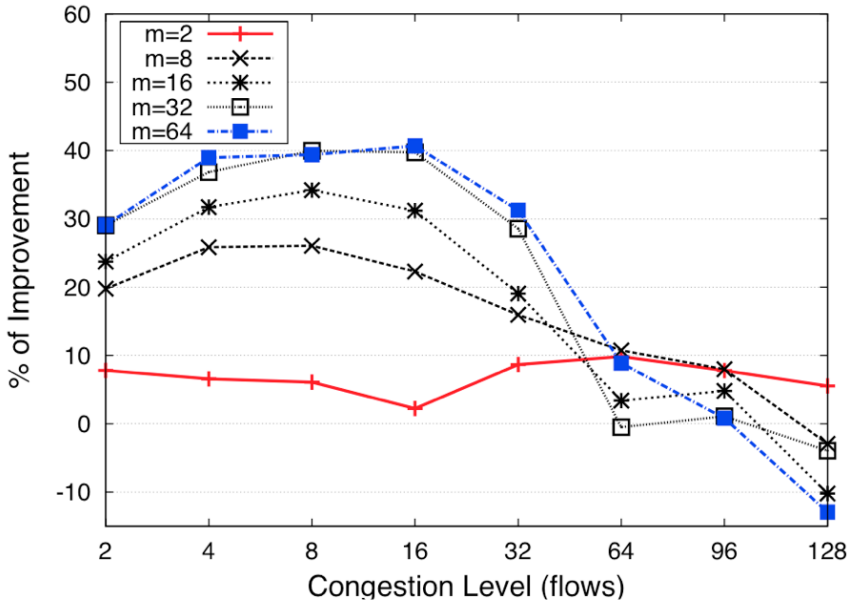
- Período de calentamiento de 400 segs de trafico de fondo.
- Varios métodos para medir el impacto de divacks
  - M-I: m divacks por cada paquete de datos
  - M-II: m divacks por cada 2 paquetes de datos
  - 2 políticas por cada método: slow start y cong. Avoidance
- Resultado tipo Sandwich:
  - Tapa de abajo: 1 ack por paquete
  - Tapa de arriba: 1 ack cada 2 paquetes
  - Relleno: divacks.

Resultado transferencias largas



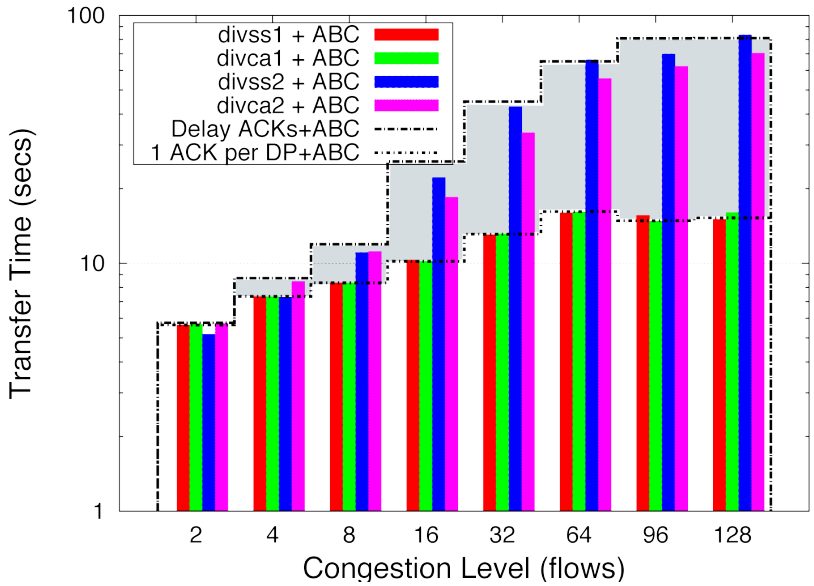
- Evaluación para m=16 varias cargas de congestión
- Ganancias máximas del 20% para poco y mediano trafico
- Divack no es atractivo para grandes cargas de congestión

¿Y si envío más divacks?



- Aumento sistemático de la ganancia para baja y mediana congestión.
- Siempre efectivo para divack mínimo: 2 divack por paquete.
- Un tiro en el pie cuando **m** crece y la congestión aumenta

¿Cómo sello el hueco de seguridad?



- Comportamiento agresivo se reduce a 1 apdp
- Comportamiento menos agresivo se reduce a una mejora menor de delayed ACKs.
- En general, el conteo de bytes es una solución para deterner los efectos de los Divacks.

Conclusiones

- Divacks no resulta todo el tiempo un peligro inminente para la Red.
  - Hay limitaciones propias ligadas al protocolo: Nagle
  - Hay limitaciones impuestas por el acceso al medio: CSMA
  - La congestión no se comporta predeciblemente —› no podemos ganar sistemáticamente
- Divacks se puede usar de forma beneficiosa si es bien dosificado:
  - Recuperación de la ventana por perdidas aleatorias
  - Arranque rápido en handovers
  - Recuperación rápida luego de interrupciones prolongadas

Trabajos Futuros (http://arcia.net.ve) (¡Se reclutan tesistas!)

- Uso de divacks en el periodo inicial de la conexión en presencia de handovers
  - Después de un time-out
  - En el arranque inicial de la conexión
- Evaluación (plataforma real) extendida en ambientes WiFi
  - Observación del impacto en otros usuarios
  - ¿Tenemos que protegernos del ataque?

Gracias. ¿Preguntas?