

IEEE 802.16 Mesh

Candidati:

Fabbri Francesca

Lo Duca Angelica

Machetti Lucio

Nurchis Maddalena

Sommario

- ✓ Introduzione
 - Specifiche di progetto
 - Tcp Reno e New Reno
 - Sack e Snack
- ✓ Analisi preliminare del sistema
 - Obiettivi
 - Tcp New Reno, Normal Ack, ftp
- ✓ Traffico ftp
 - Studio caso alto
 - Studio caso basso
 - ftp approfondimento
- ✓ Traffico cbr
 - Studio caso alto
 - Studio caso basso
- ✓ Conclusioni generali

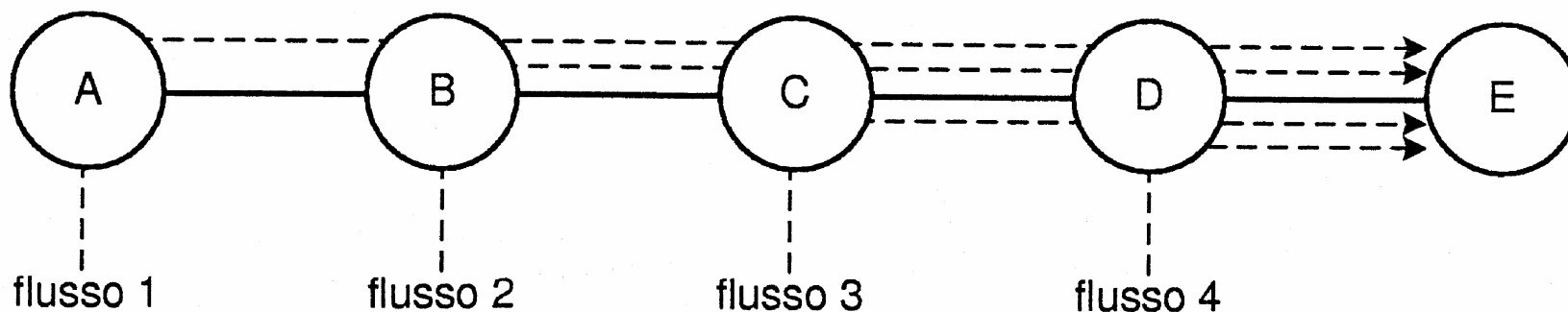
Introduzione

Introduzione

Specifiche del progetto

Specifiche di progetto

- ✓ **Scenario di simulazione:** 5 nodi sono collegati in una topologia disposta a catena, ciascuno possiede il traffico TCP diretto verso il nodo di uscita E



Specifiche di progetto

- ✓ **Condizioni di traffico:**
 - Modelli di traffico FTP e CBR
 - TCP Reno e New Reno
 - ✓ **Metriche:**
 - Ritardo end-to-end dei pacchetti
 - Throughput e packet loss dei flussi di traffico
 - ✓ **Analisi:** confronto tra le modalità di riscontro dei ricevitori tcp:
 - Normal ACK (ACK)
 - Selective ACK (SACK)
 - Selective negative ACK (SNACK)
- al variare dei parametri significativi

Introduzione

TCP Reno e New Reno

TCP Reno e New Reno

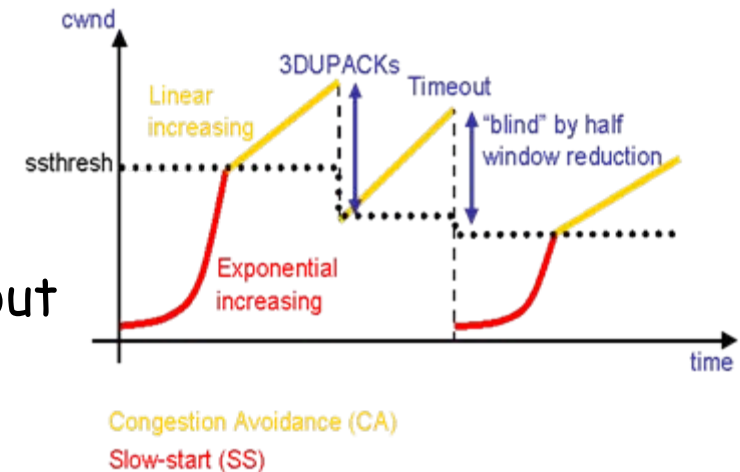
✓ TCP Reno utilizza i meccanismi di **fast retransmit** e **fast recovery**

– Fast retransmit:

- Ritrasmissione dopo 3 dupack o timeout

– Fast recovery:

- no slow start
- congestion window = congestion threshold = valore esistente della finestra prima della perdita / 2

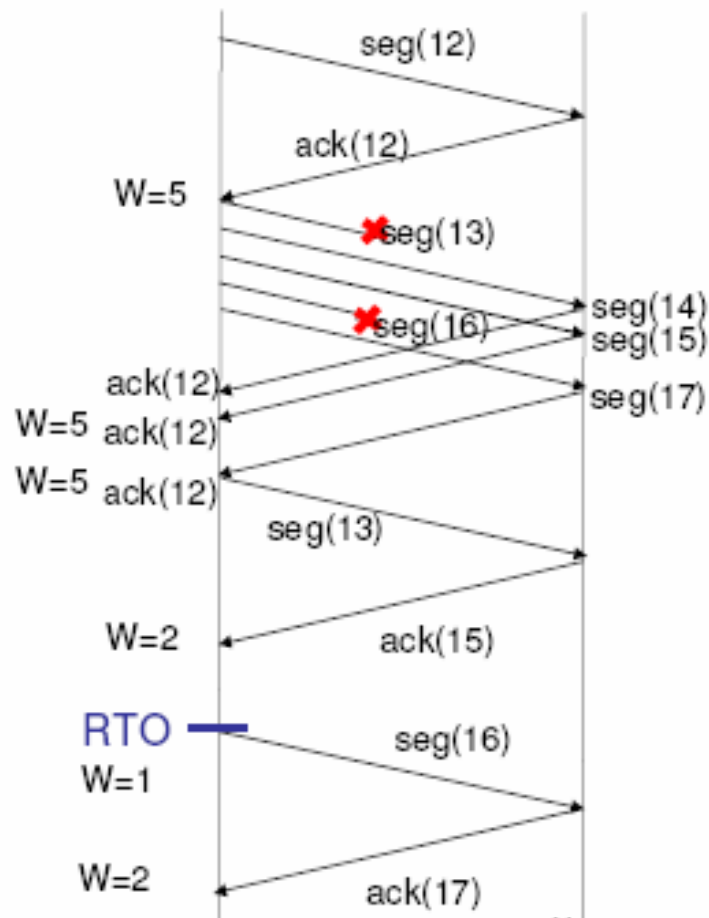


TCP Reno e New Reno

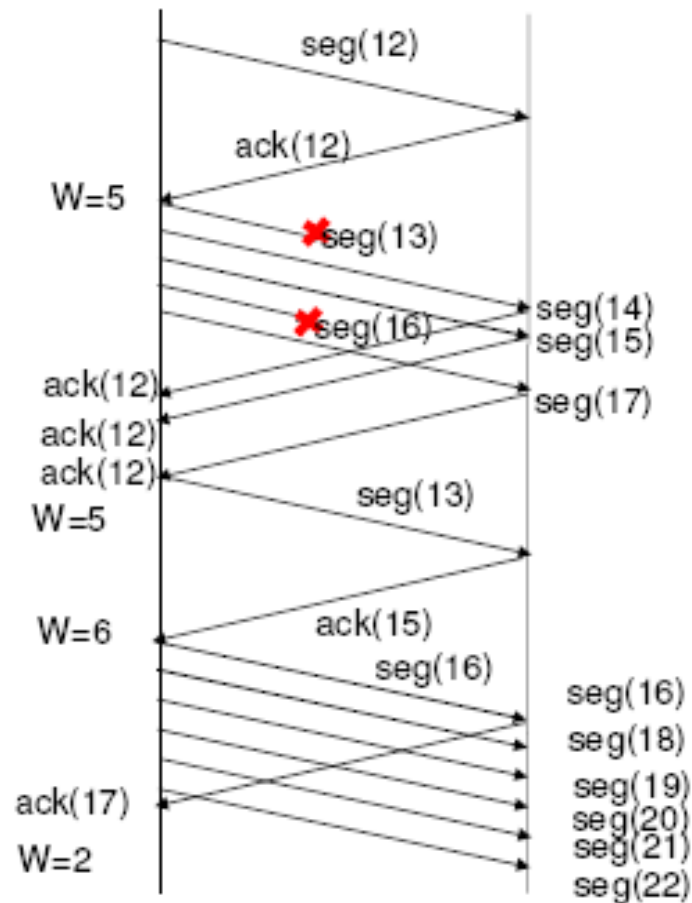
- ✓ TCP New Reno si differenzia dal Reno in:
 - In presenza di ack parziale, ritrasmissione del segmento successivo a quello riconosciuto
 - Ri-innesco del fast recovery ad ogni ack parziale
 - Uscita dal fast recovery quando tutti i segmenti trasmessi sono stati riconosciuti

TCP Reno e New Reno

✓ Confronto TCP Reno e New Reno:



Reno



New Reno

TCP Reno e New Reno

✓ Conclusioni:

- Miglioramento di prestazioni del New Reno rispetto al Reno quando ci sono più segmenti persi nella finestra corrente

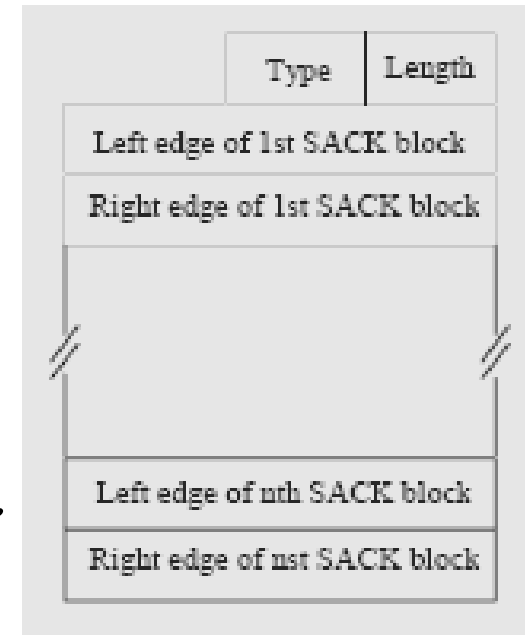
Introduzione

SACK e SNACK

SACK e SNACK

✓ TCP Sack prevede:

- Opzione Sack nell'header TCP dell'ACK
 - Lista dei segmenti ricevuti correttamente
- Ritrasmissione dei soli segmenti persi
- Pipe = numero di segmenti in trasmissione
- Invio di segmenti se $\text{pipe} < \text{cwnd}$
- Uscita dal fast recovery se tutti i segmenti in trasmissione sono riscontrati
- Lunghezza opzione Sack = $8xn + 2$ bytes (n numero di blocchi specificati)



SACK e SNACK

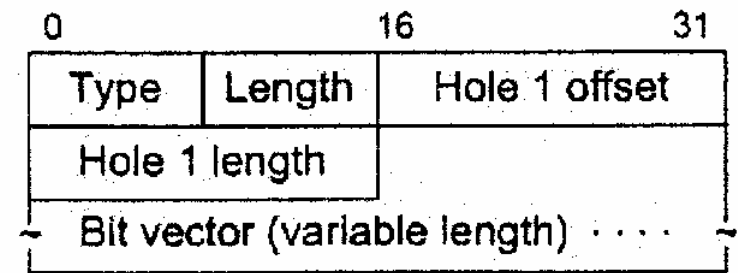
✓ Problemi del TCP Sack:

- Non più di 3 blocchi specificabili nell'opzione sack
- Necessari più ack per acquisire l'immagine del buffer del ricevitore

SACK e SNACK

✓ TCP Snack prevede:

- Snack = sack + nak
- Opzione snack nell'header TCP dell'ack
 - Lista dei segmenti mancanti nel buffer del ricevitore
 - *Hole 1 offset + hole 1 length + bit vector*
- Ritrasmissione dei soli segmenti mancanti indicati
- Uscita dal fast recovery se tutti i segmenti trasmessi sono stati riconosciuti



SACK e SNACK

✓ Conclusioni:

- Snack fornisce al sender una immagine completa del buffer del ricevitore con un singolo ack, non succede in Sack
- Prestazioni migliori del Sack e dello Snack rispetto al normal Ack nel caso di più segmenti persi nella stessa finestra

Analisi preliminare del sistema

Analisi preliminare del sistema



Obiettivi

Valori scelti per le simulazioni

Warm (tempo simulato)	100 sec
Duration (tempo simulato)	200 sec
Numero massimo di run	20
α	0.05
Confidence level threshold	0.10

Confronti

Tipi di riscontro confrontati

	SACK RENO	SNACK NEW RENO
ACK RENO		
ACK NEW RENO		

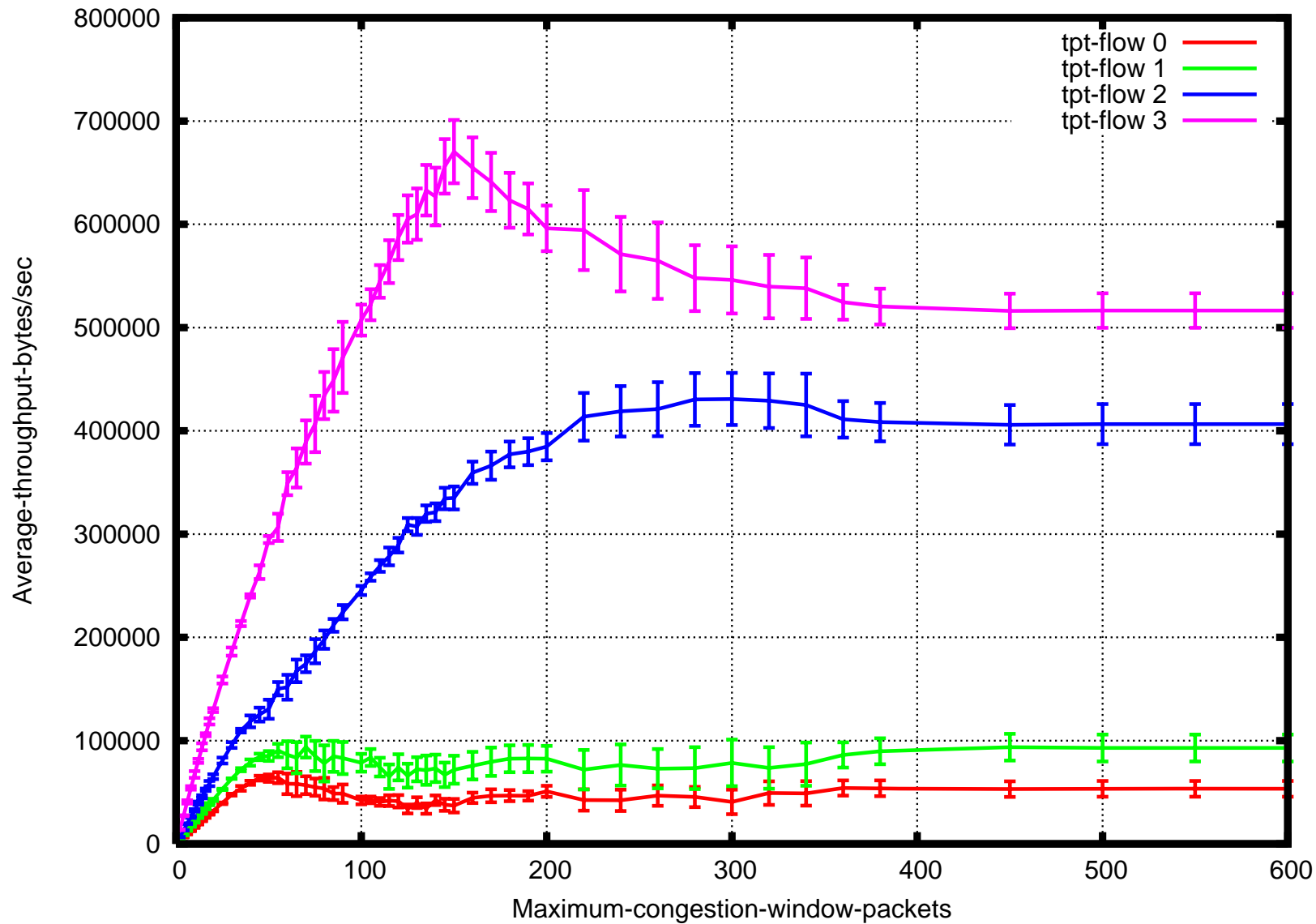
Simulazioni al variare dei parametri

<div style="display: flex; justify-content: space-between;"> Parametro fisso Parametro variabile </div>	Max congestion window (packets)	Max advertised window (packets)	Buffer (bytes)
Max congestion window	-	65536	1000000
Max advertised window	65536	-	1000000
Buffer	65536	65536	-

Analisi preliminare del sistema

Tcp New Reno, normal ack, ftp

Andamento del throughput al variare della cwnd



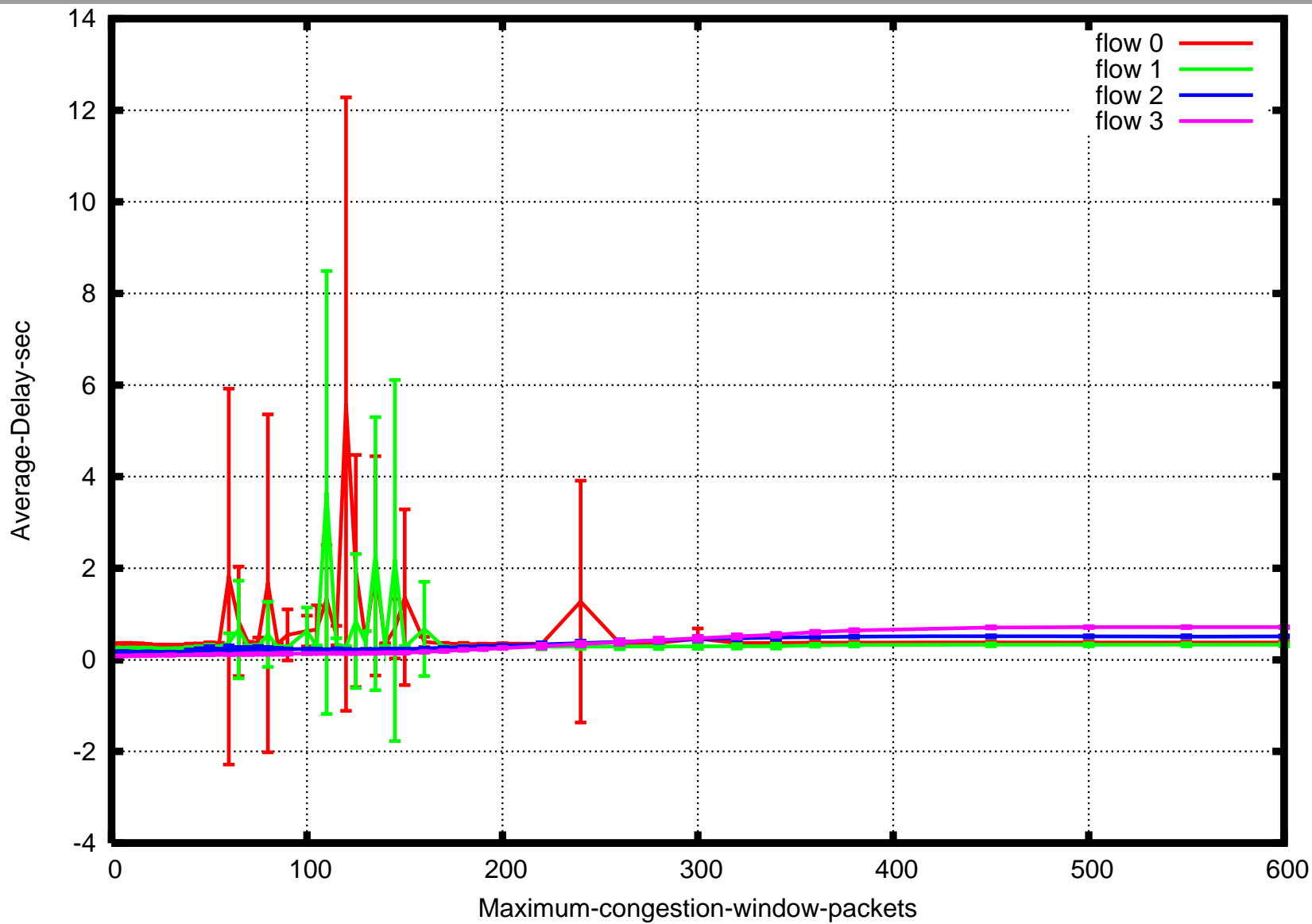
Andamento del throughput al variare della cwnd

- ✓ Per valori alti di congestion window l'unico fattore che determina il throughput è il link fisico.
- ✓ Il link fisico ha una **capacità massima di 1.742.000 bytes/sec**, questo implica che la somma dei throughputs di tutti e quattro i flussi sul link di uscita non può superare tale capacità. Questo infatti a regime si verifica.

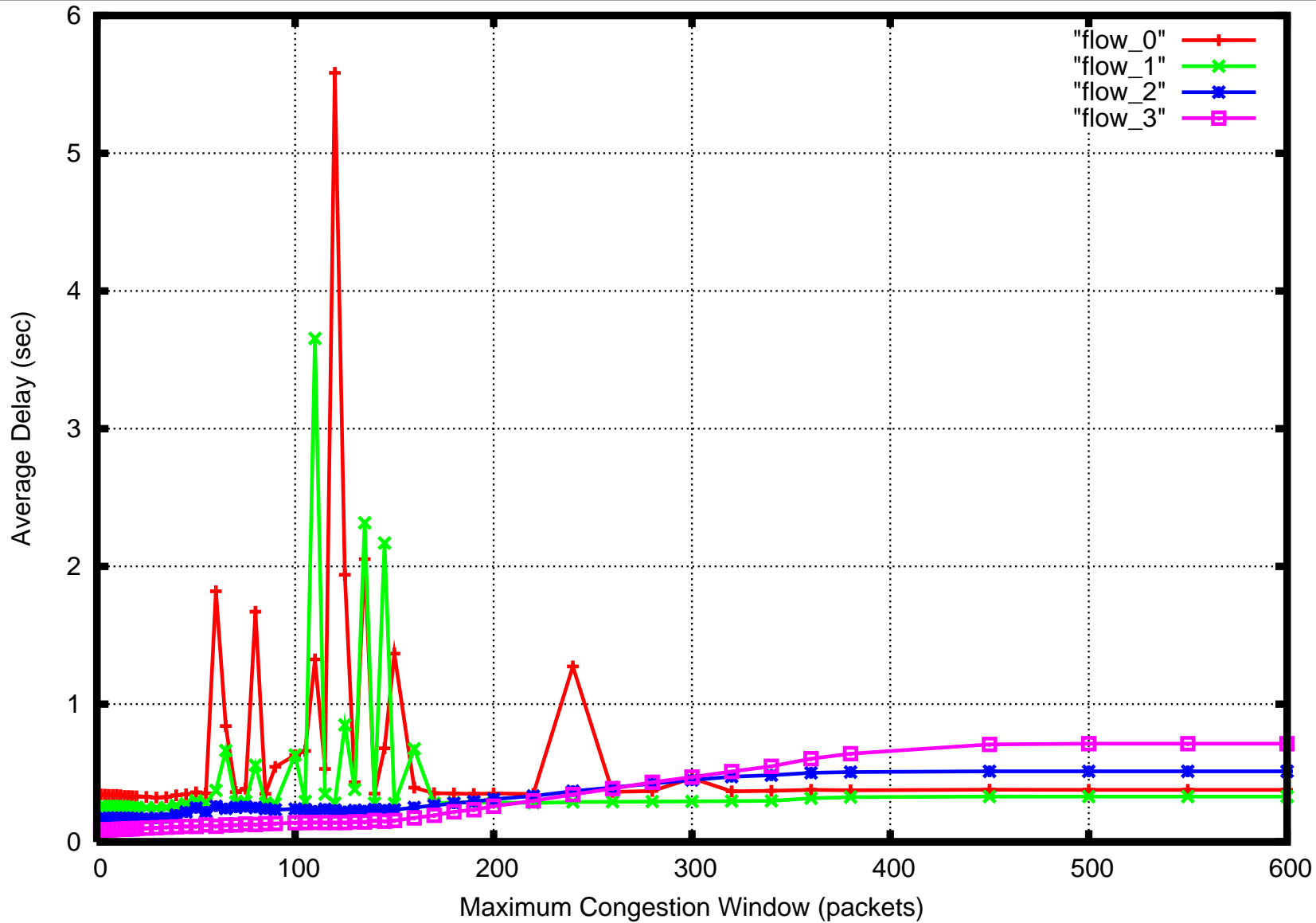
Andamento del throughput al variare della cwnd

- ✓ A regime il **flusso 3** (quello più vicino al nodo di uscita) raggiunge un throughput maggiore rispetto agli altri flussi, in quanto, trovandosi fisicamente più vicino alla destinazione, riceve prima degli altri l'ack di riscontro per cui può subito aumentare la sua congestion window.

Andamento del delay al variare della cwnd



Andamento del delay al variare della cwnd



Analisi preliminare del sistema

D'ora in poi non verranno più inseriti gli intervalli di confidenza per rendere più agevole la lettura e la valutazione dei grafici proposti.

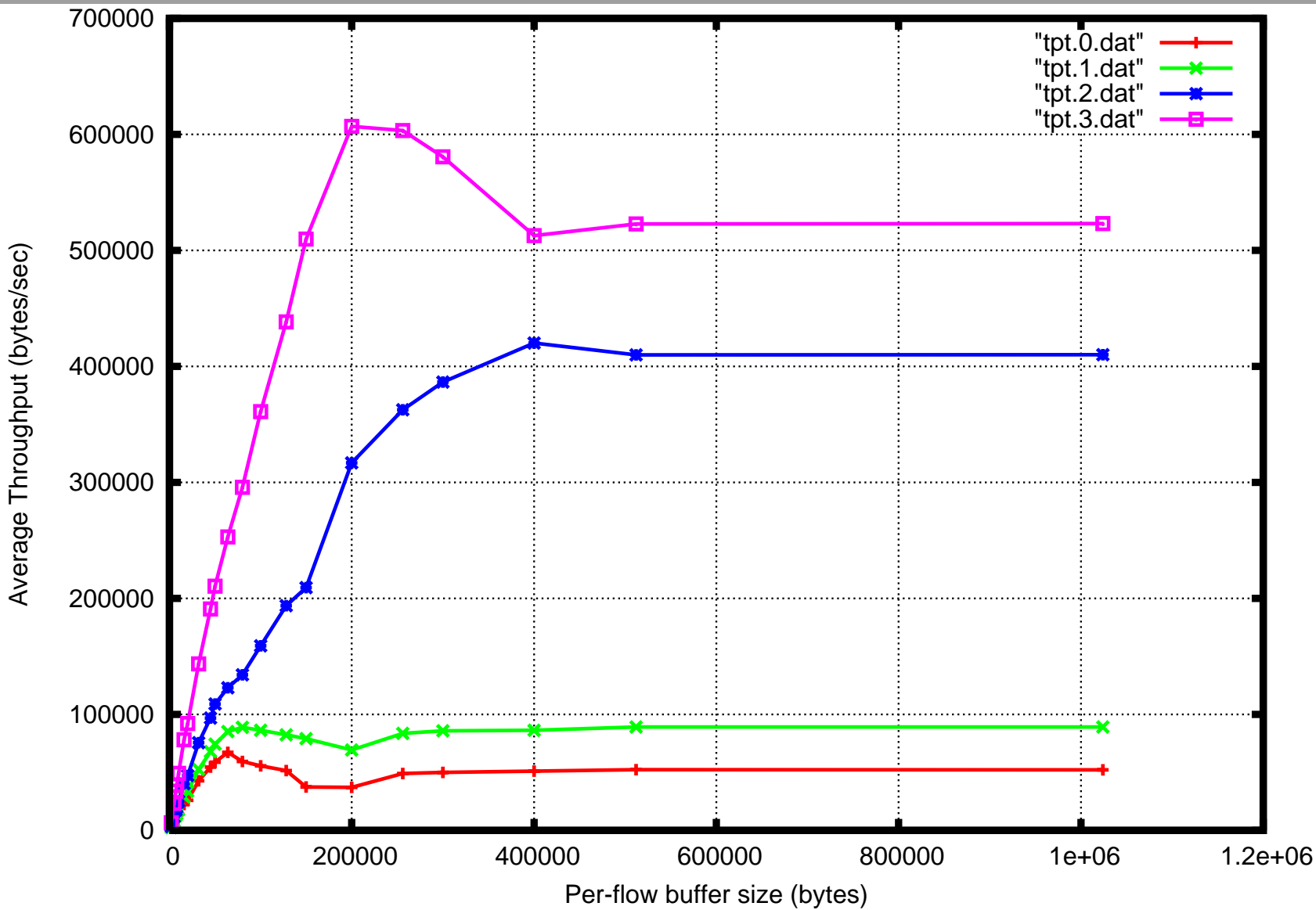
Andamento del delay al variare della cwnd

- ✓ Per **valori bassi** della congestion window, il ritardo del flusso 3 è minore rispetto a quello degli altri flussi, in quanto prevale la vicinanza fisica alla destinazione.
- ✓ Per **valori alti** della congestion window, il ritardo del flusso 3 è maggiore rispetto a quello degli altri flussi.

Andamento del delay al variare della cwnd (cont.)

- ✓ Questo è dovuto all'algoritmo di scheduling utilizzato (**Fair Round Robin**).
- ✓ Ogni flusso all'interno di un round può trasmettere al massimo 27072 bytes.
- ✓ A parità di **congestion window alta**, il flusso 3 ha un throughput maggiore, per cui inviando un maggior numero di dati nell'unità di tempo rispetto agli altri flussi, necessita di più rounds per trasmettere interamente i suoi dati.
- ✓ Pertanto il delay del flusso 3 è maggiore rispetto a quello degli altri.

Andamento del throughput al variare del buffer



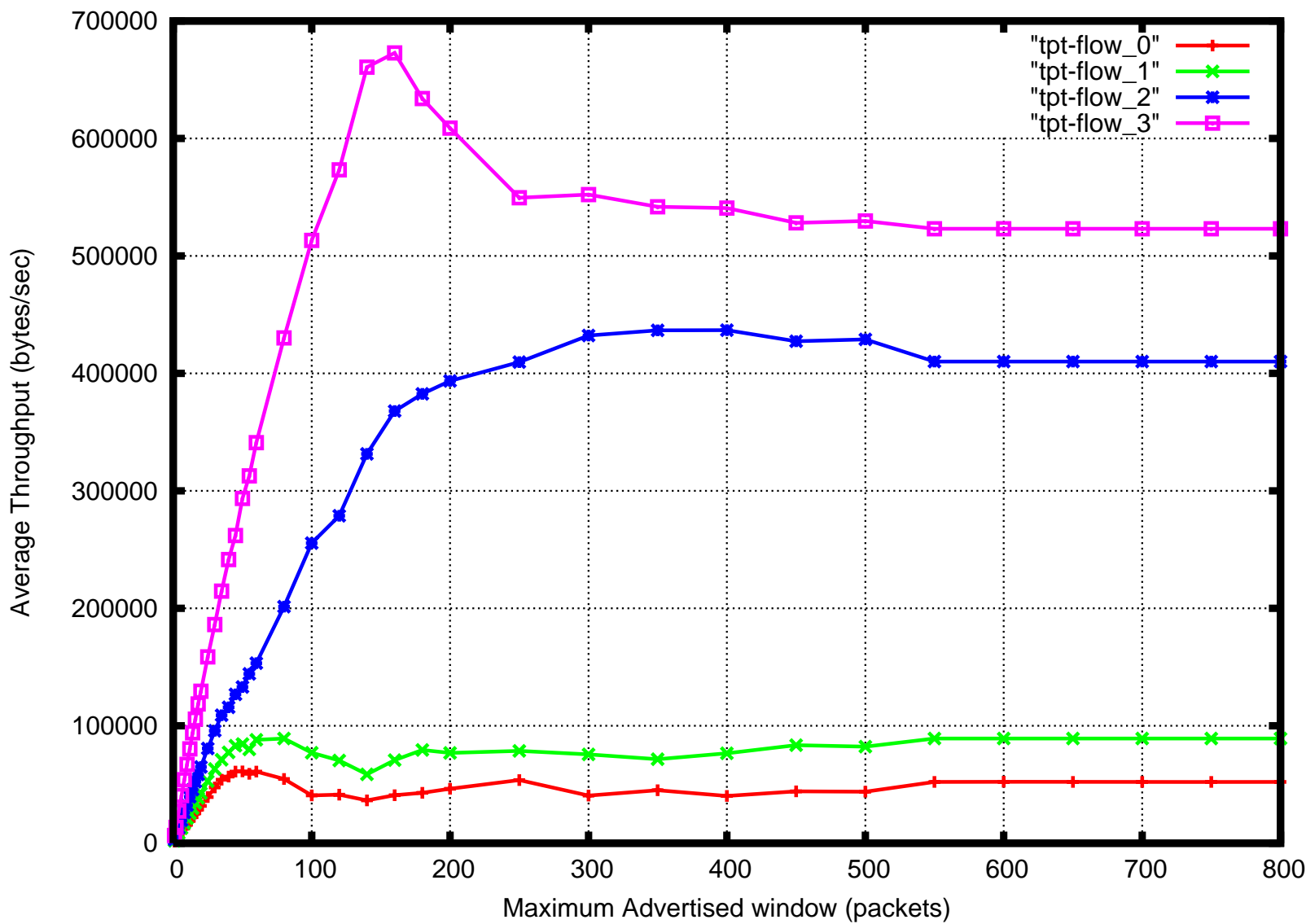
Andamento del throughput al variare del buffer

- ✓ Il comportamento del sistema è identico al caso della congestion window.
 - Per valori bassi il buffer è il fattore limitante
 - In corrispondenza di valori alti, il buffer non influenza il comportamento del sistema. Quindi l'unico fattore limitante è la capacità del link fisico.

Andamento del delay al variare del buffer

- ✓ Il comportamento del sistema è identico al caso della congestion window.

Andamento del throughput al variare della awnd



Andamento del throughput al variare della awnd

- ✓ Il comportamento del sistema è identico al caso della congestion window. Ciò è dovuto al fatto che la finestra effettiva è data dalla minore tra la congestion window e l'advertised window.

Casi di studio

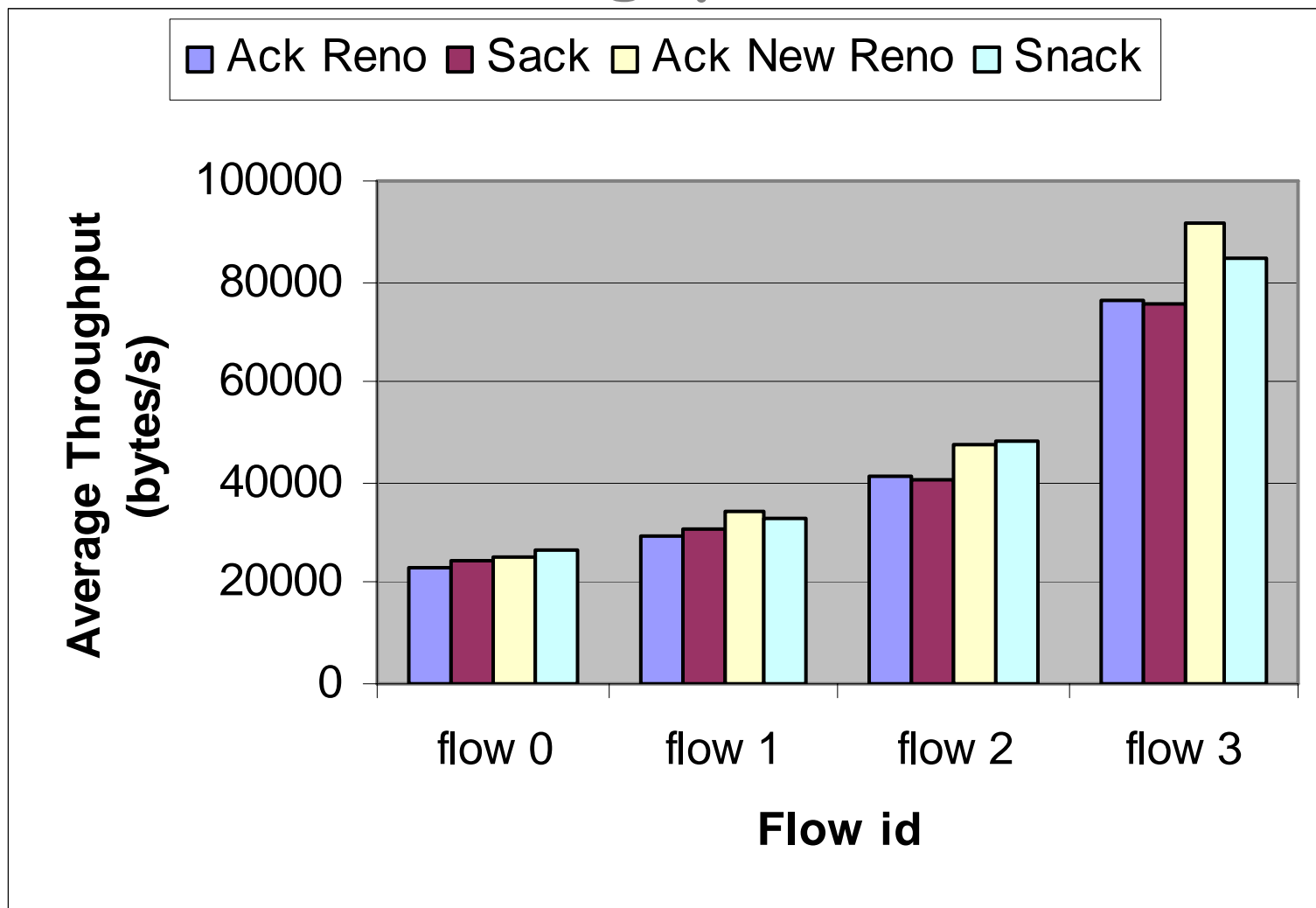
Parametro	Max congestion window (packets)	Max advertised window (packets)	Buffer (bytes)	Packet size (bytes)
Casi di studio				
Caso basso	40	45	20000	1000
Caso alto	450	600	512000	1000

Traffico FTP

Studio caso basso

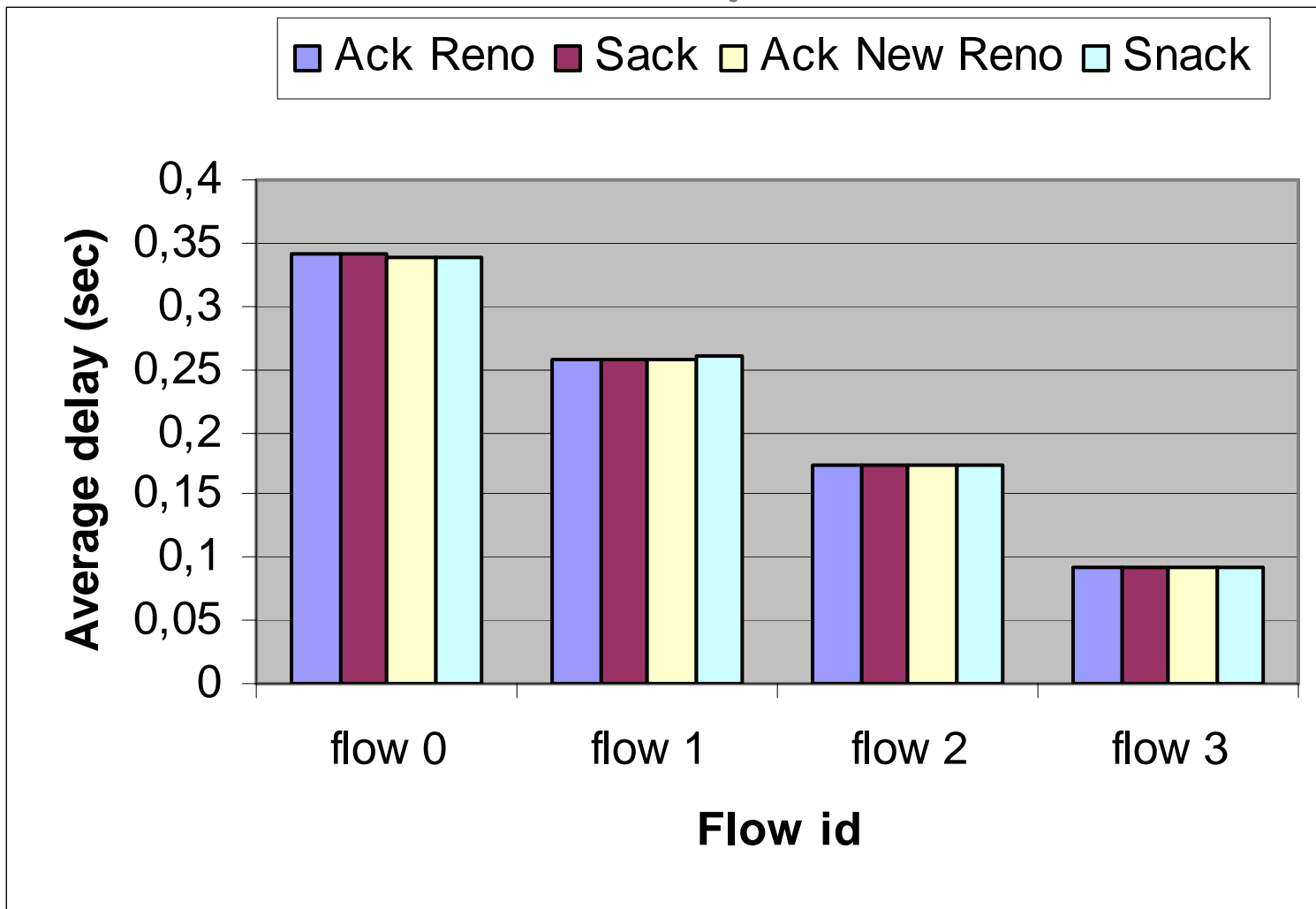
FTP Basso

Throughput



FTP Basso

Delay



FTP Basso

- ✓ Modifica scenario:
 - Caso basso con solo il flusso 3 acceso

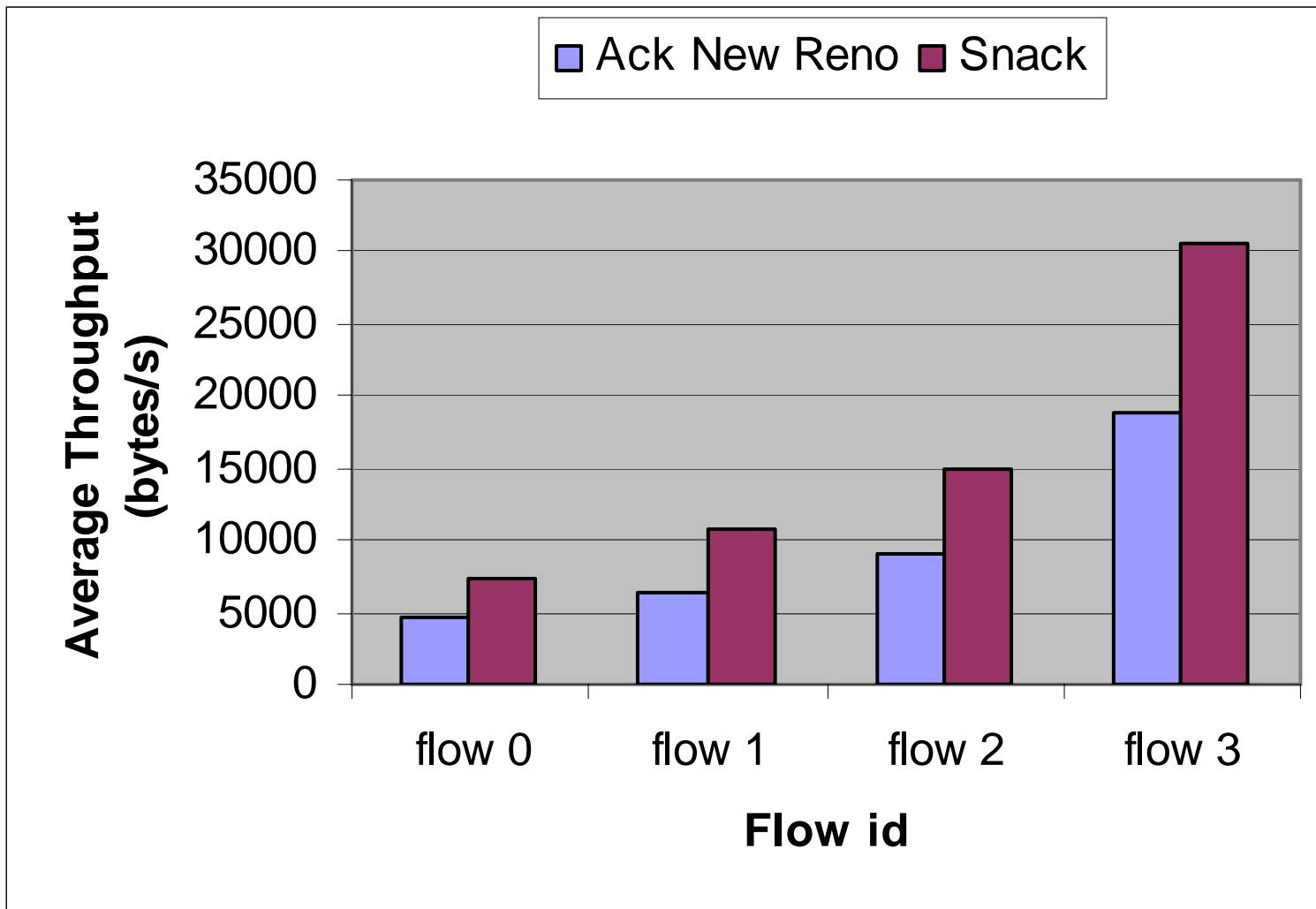
- ✓ Conclusioni:
 - Snack porta un througput maggiore rispetto a Ack

 - Flusso 3 in Snack risente della presenza di altri flussi

ACK New Reno	Snack
(average tpt bytes/s)	(average tpt bytes/s)
63363	65597

FTP Basso: perdita forzata

Throughput



FTP basso

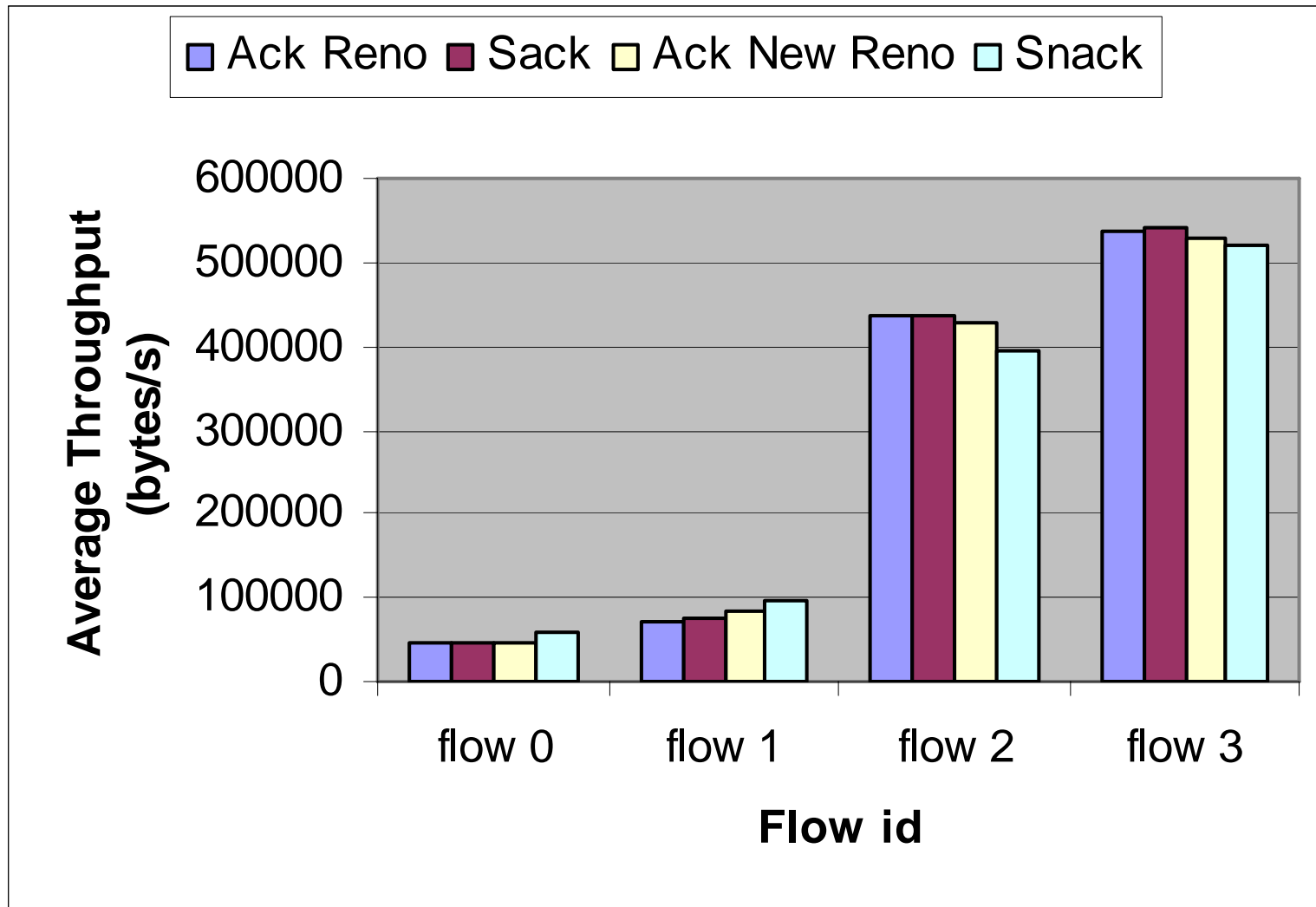
✓ Conclusioni:

- Perdite per buffer overflow
- Probabilità bassa di perdita di segmenti nella stessa finestra (finestra piccola)
 - Circa stesso throughput nel caso di Ack Reno e Sack
 - Circa stesso throughput per i flussi 0, 1 e 2 nel caso di Ack New Reno e Snack
- Per il flusso 3 con Ack New Reno si ottiene un **throughput maggiore**
 - Spengendo tutti i flussi tranne 3, Snack dà un throughput maggiore rispetto ad Ack New Reno
 - Con Snack il throughput del flusso 3 risente della presenza dei altri flussi
- Forzando la perdita di più segmenti nella stessa finestra, il comportamento dello Snack è migliore

FTP

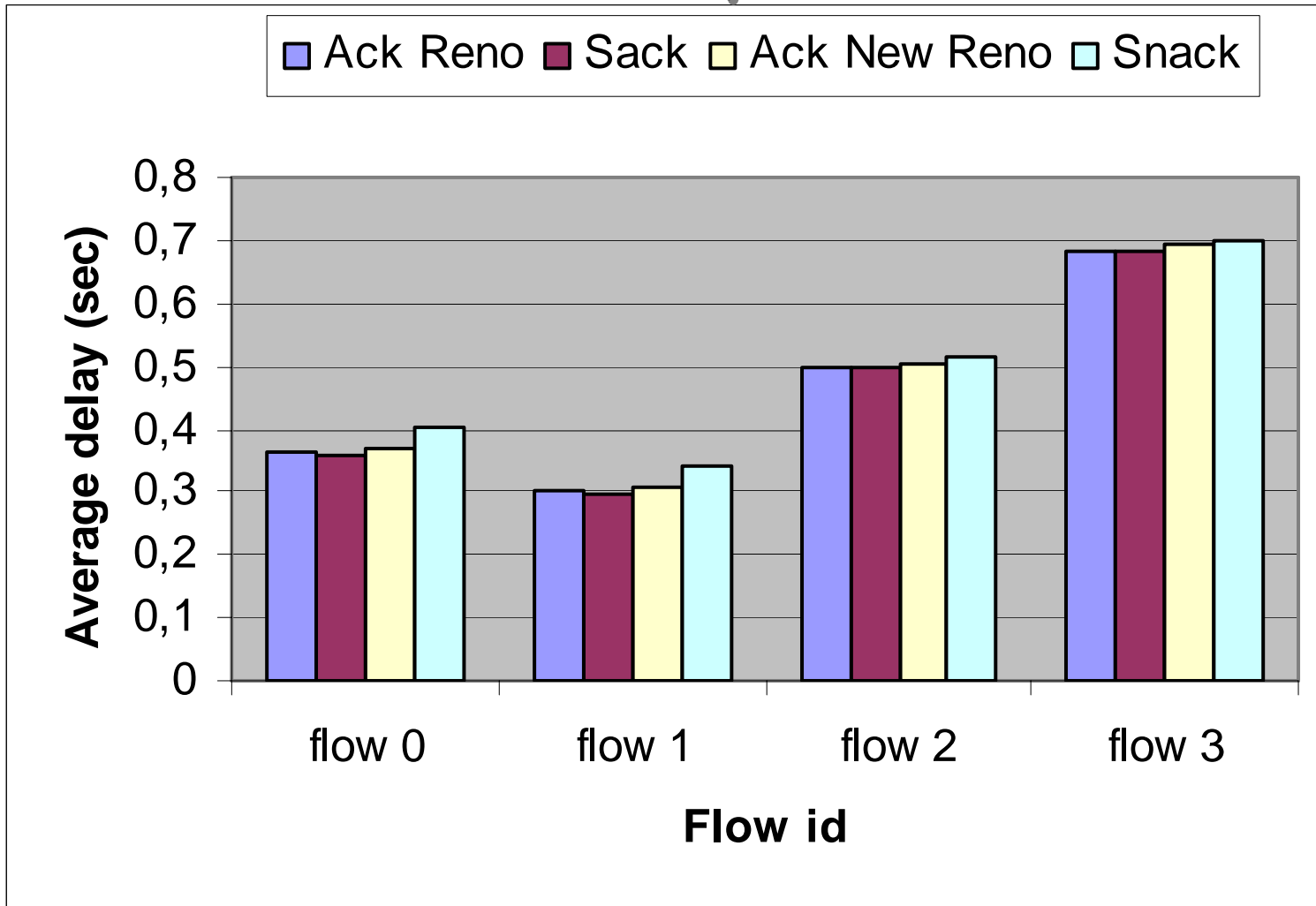
Studio caso alto

Throughput



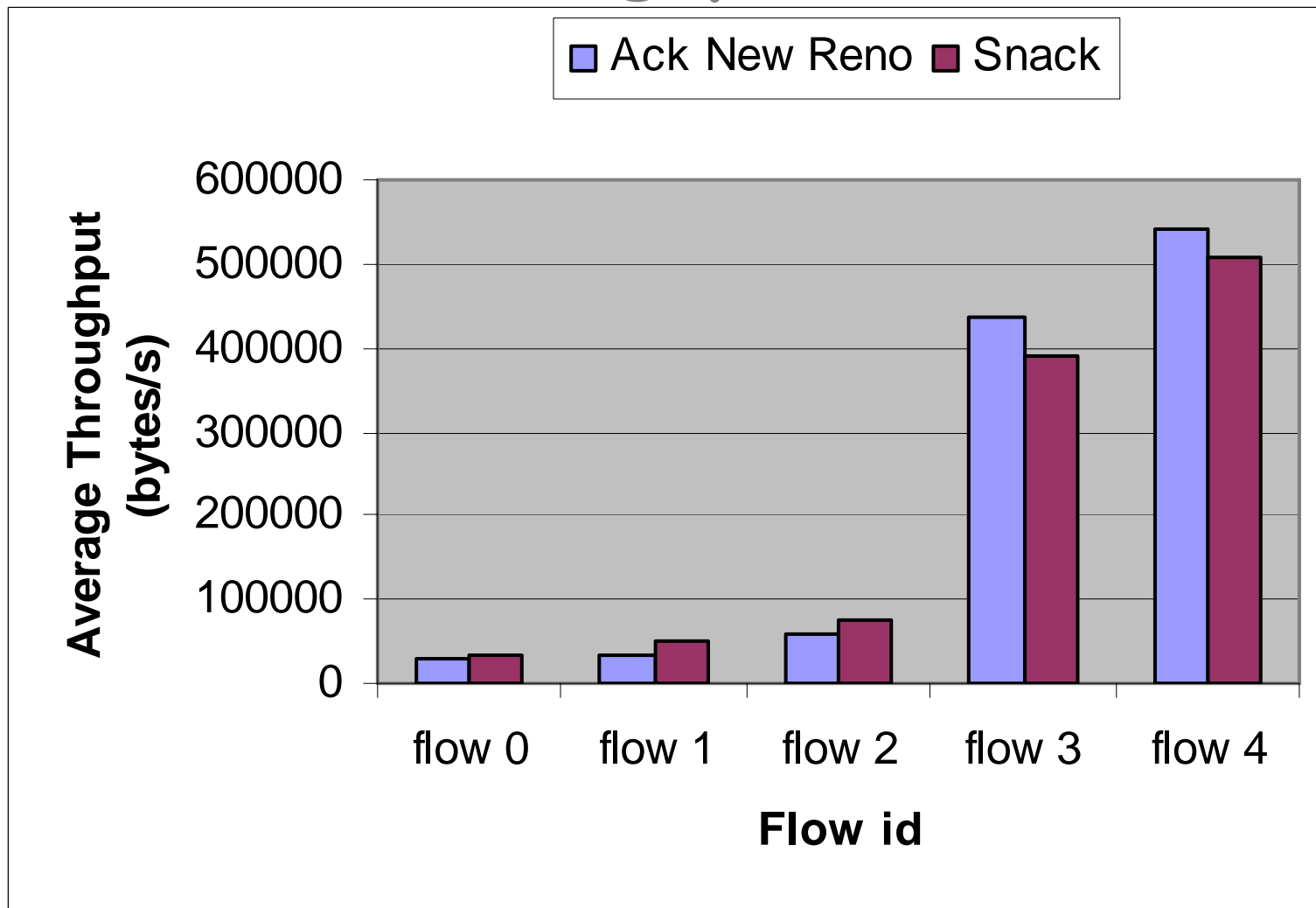
FTP alto

Delay



FTP alto: scenario a 6 nodi

Throughput



FTP alto

✓ Conclusioni:

- Azione di Sack e Snack per scatto del **timeout**
- Throughput dei flussi nel caso di Ack e Sack simili
 - Poche perdite
- Snack avvantaggia i flussi lontani rispetto a quelli più vicini
- Somma totale dei throughputs dei flussi nel caso dello Snack minore rispetto a quella sotto l'azione dell'Ack
 - Flussi lontani più costosi in termini di risorse utilizzate

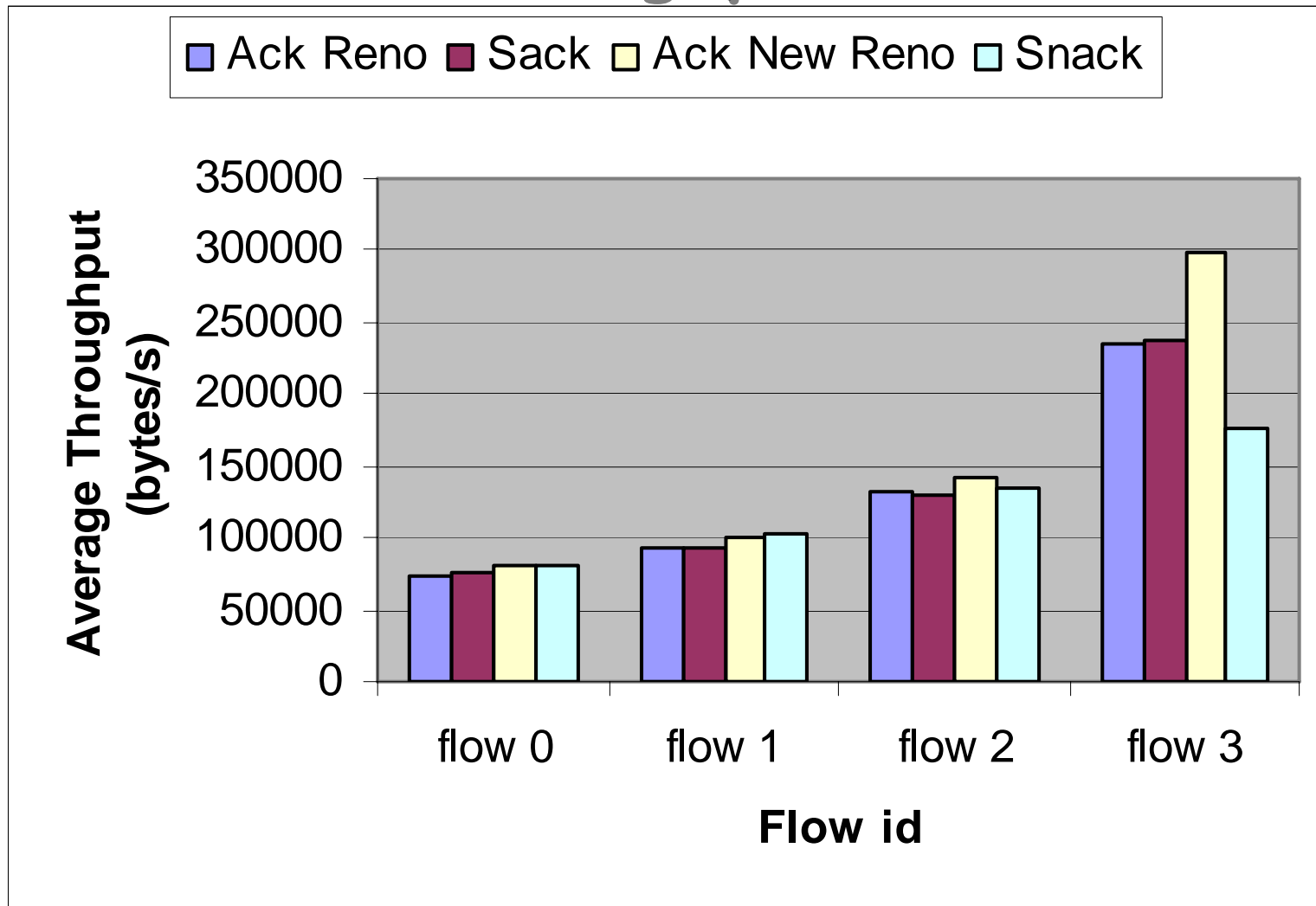
FTP: approfondimento

FTP: approfondimento

- ✓ Modifica allo scenario di simulazione (livello MAC):
 - Frame = parte controllo + parte dati
 - Per ogni burst da trasmettere va in esecuzione un **protocollo di handshake**, per avere l'autorizzazione a inviare pacchetti (parte controllo del frame)
 - Durata handshake = $(16/\text{control}) * 4\text{ms} * 3$
 - Parametro **CONTROL** impostato a 8 (durata = 24 ms)

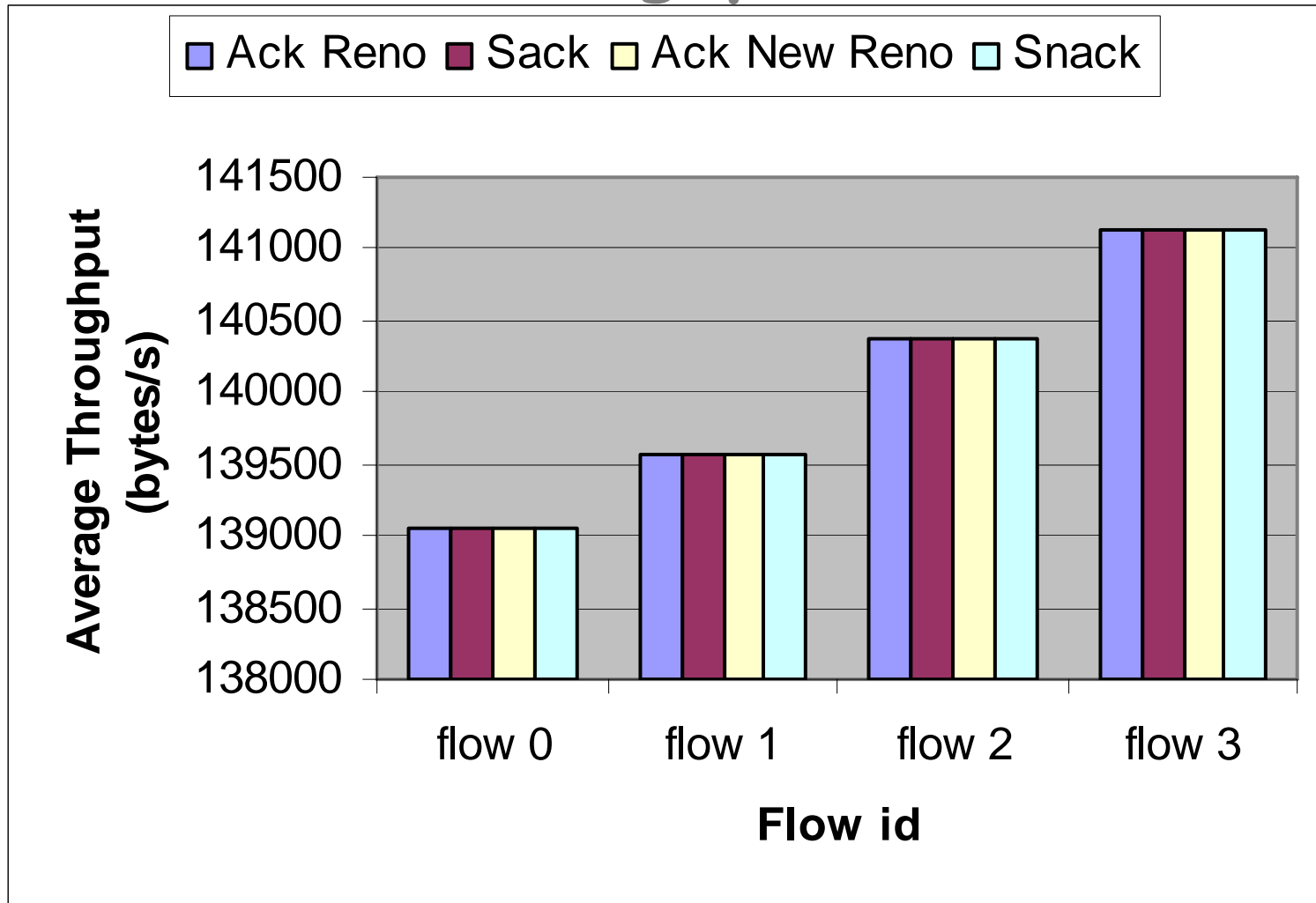
FTP Basso: control 8

Throughput



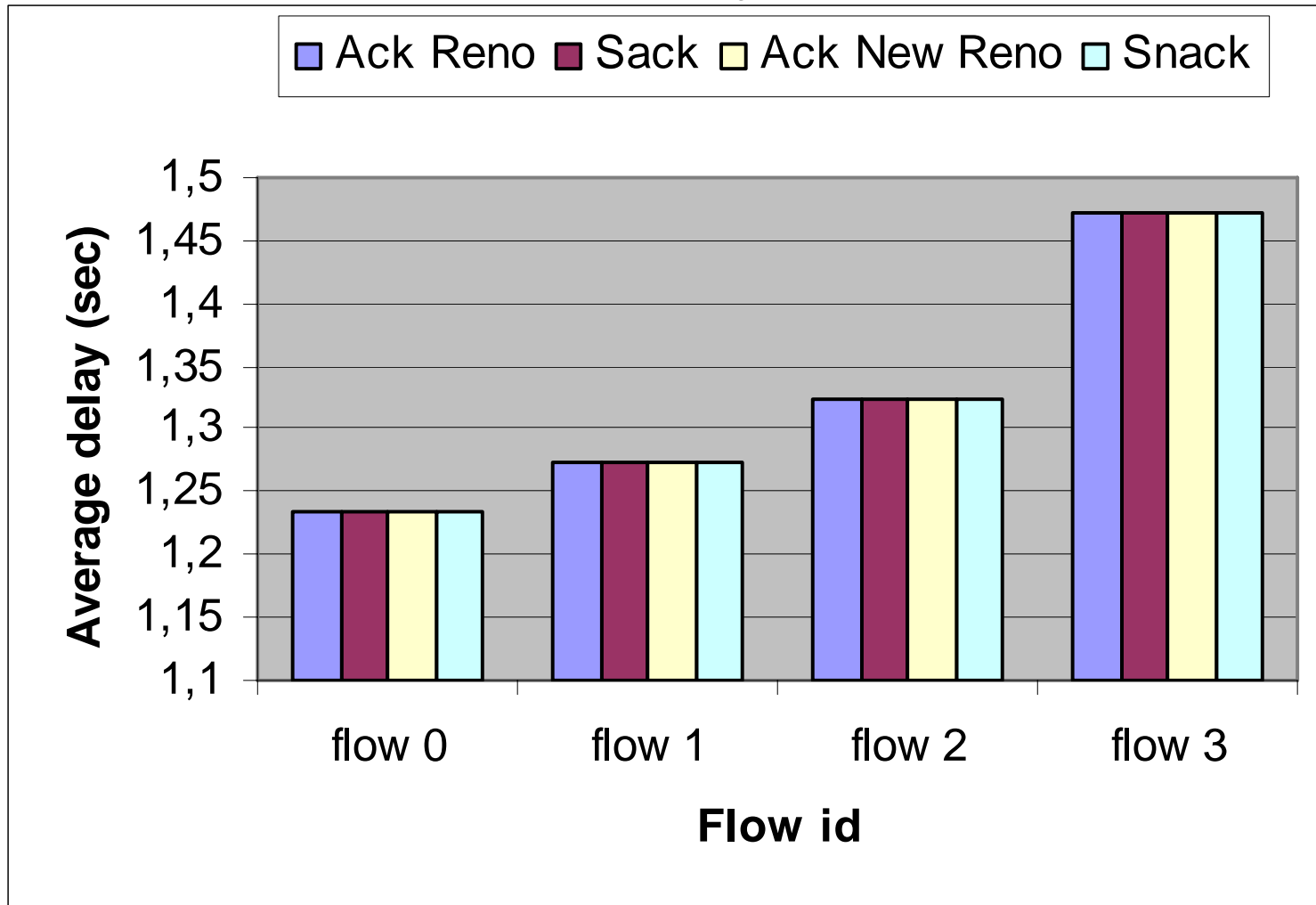
FTP alto: control 8

Throughput



FTP alto: control 8

Delay



FTP: approfondimento

✓ Conclusioni:

– Caso BASSO:

- Throughput simili tra Ack Reno e Sack
- Throughput simili tra Ack New Reno e Snack (ad eccezione del flusso 3)
- Aumento di throughput di tutti i flussi rispetto al caso in cui control è a 2

– Caso ALTO:

- Throughput uguali tra Ack, Sack e Snack perché non ci sono perdite
- Throughput dei flussi simili tra loro (Fair-RR)

Traffico CBR

Traffico CBR

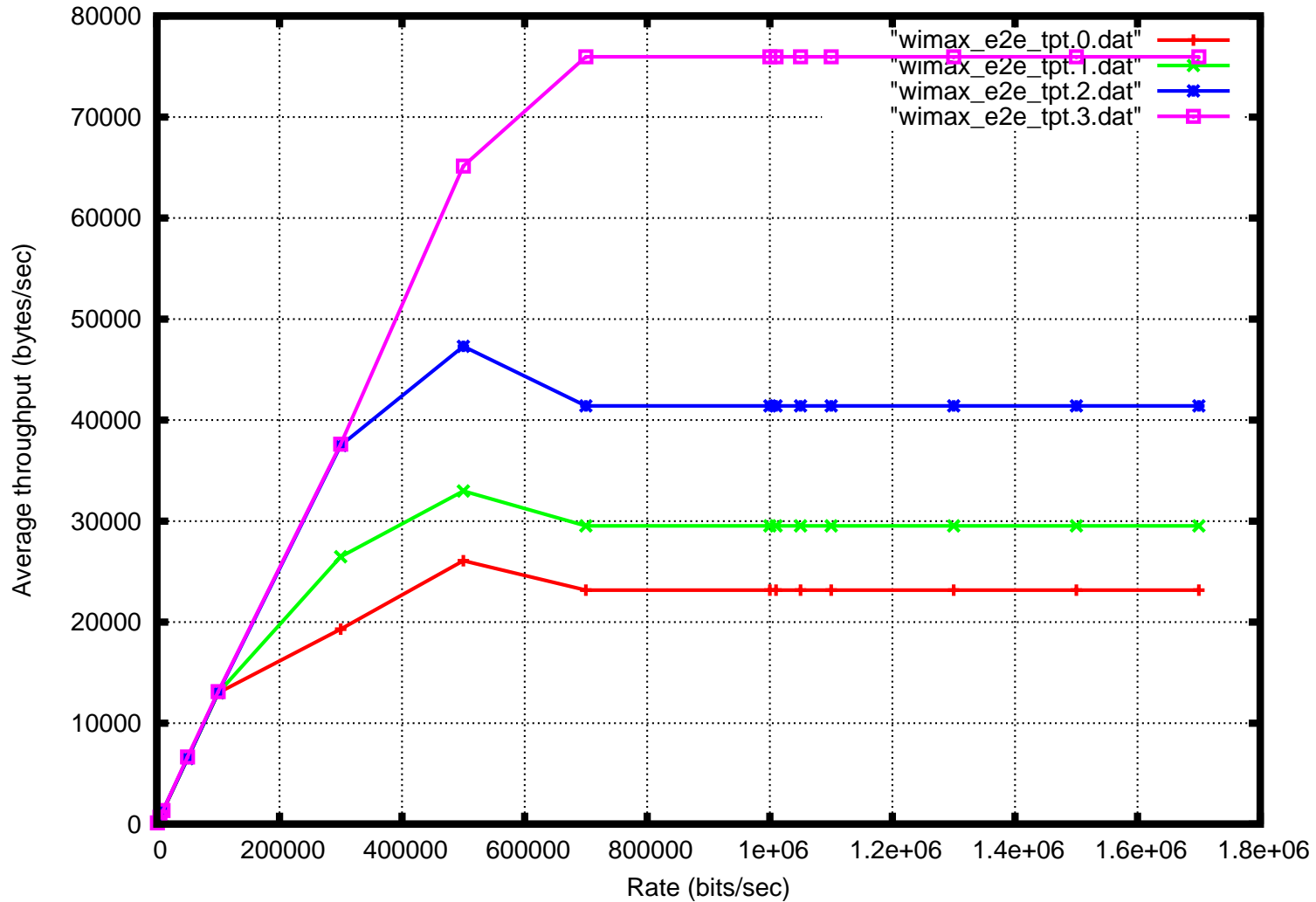
- ✓ Studio al variare del rate: 1000-100000000 bits/s
- ✓ Analisi risultati
- ✓ Individuazione dei valori significativi del rate:
 - **Caso basso**
 - Rate basso 100000 bits/s
 - Rate intermedio 300000 bits/s
 - Rate alto 700000 bits/s (tpt si stabilizza)
 - **Caso alto**
 - Rate basso 500000 bits/s
 - Rate intermedio 1700000 bits/s
 - Rate alto 6000000 bits/s (tpt si stabilizza), 7000000 bits/s per snack

Traffico CBR

Studio caso basso

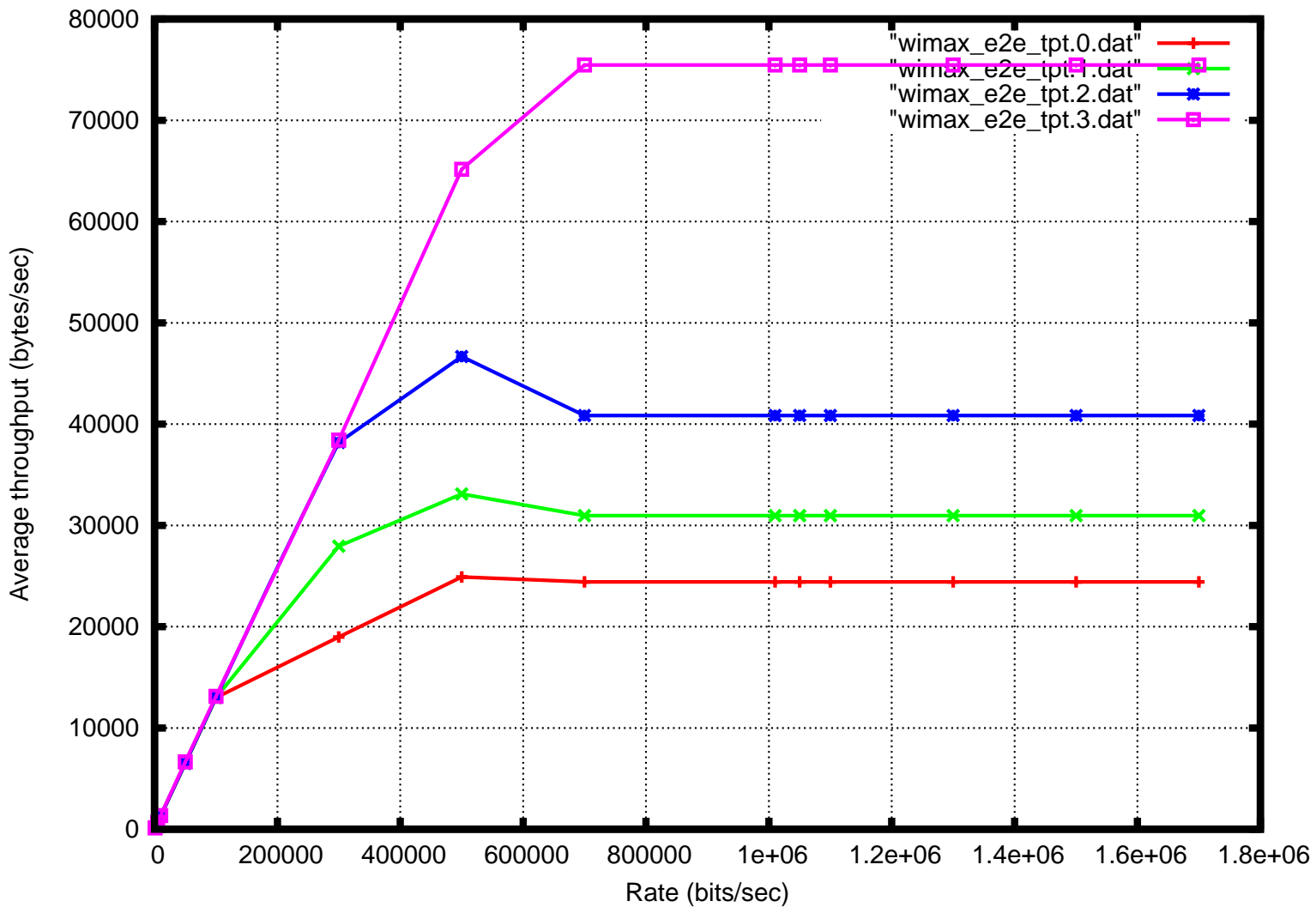
Studio caso basso

Tpt ACK Reno



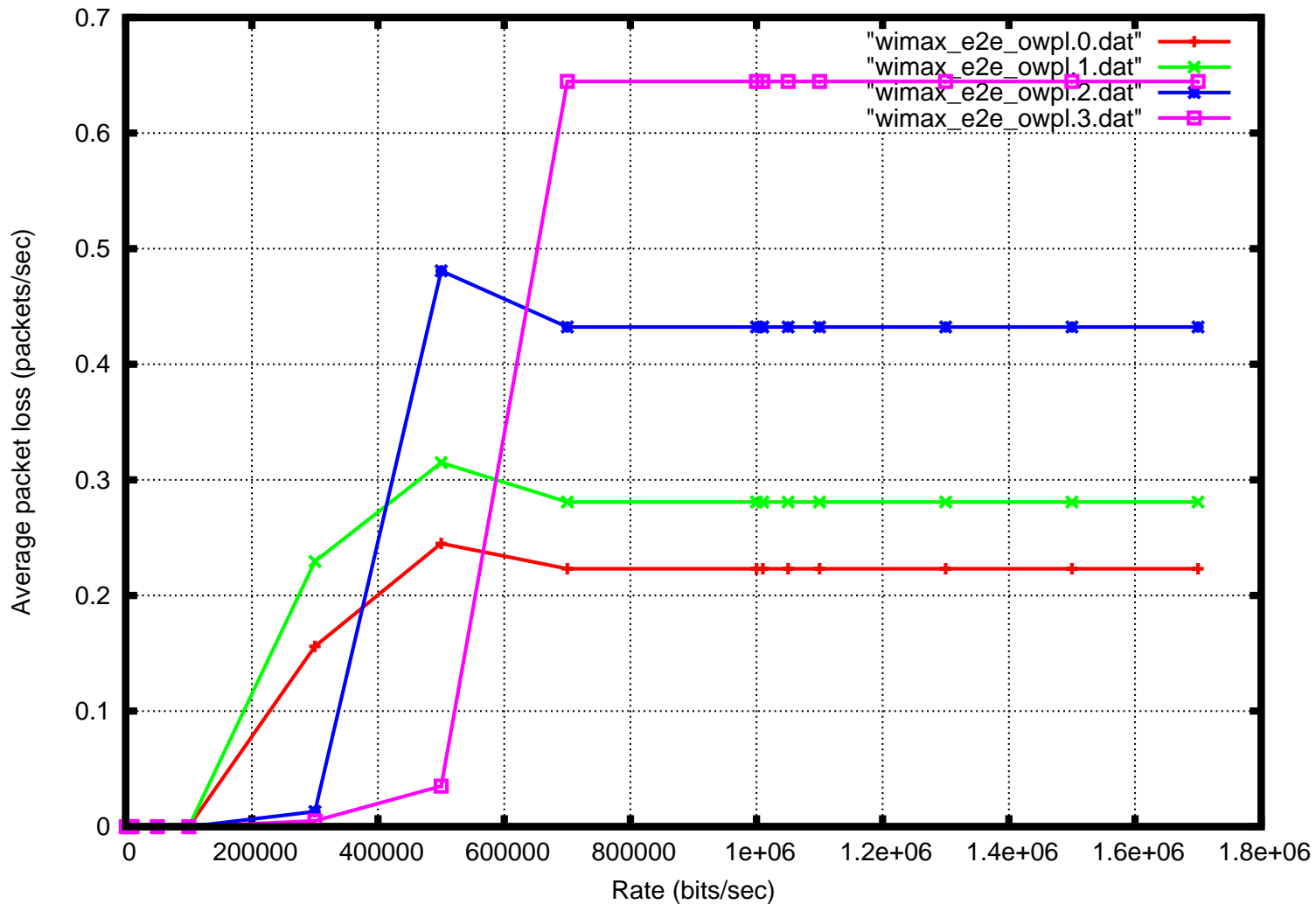
Studio caso basso

Tpt SACK



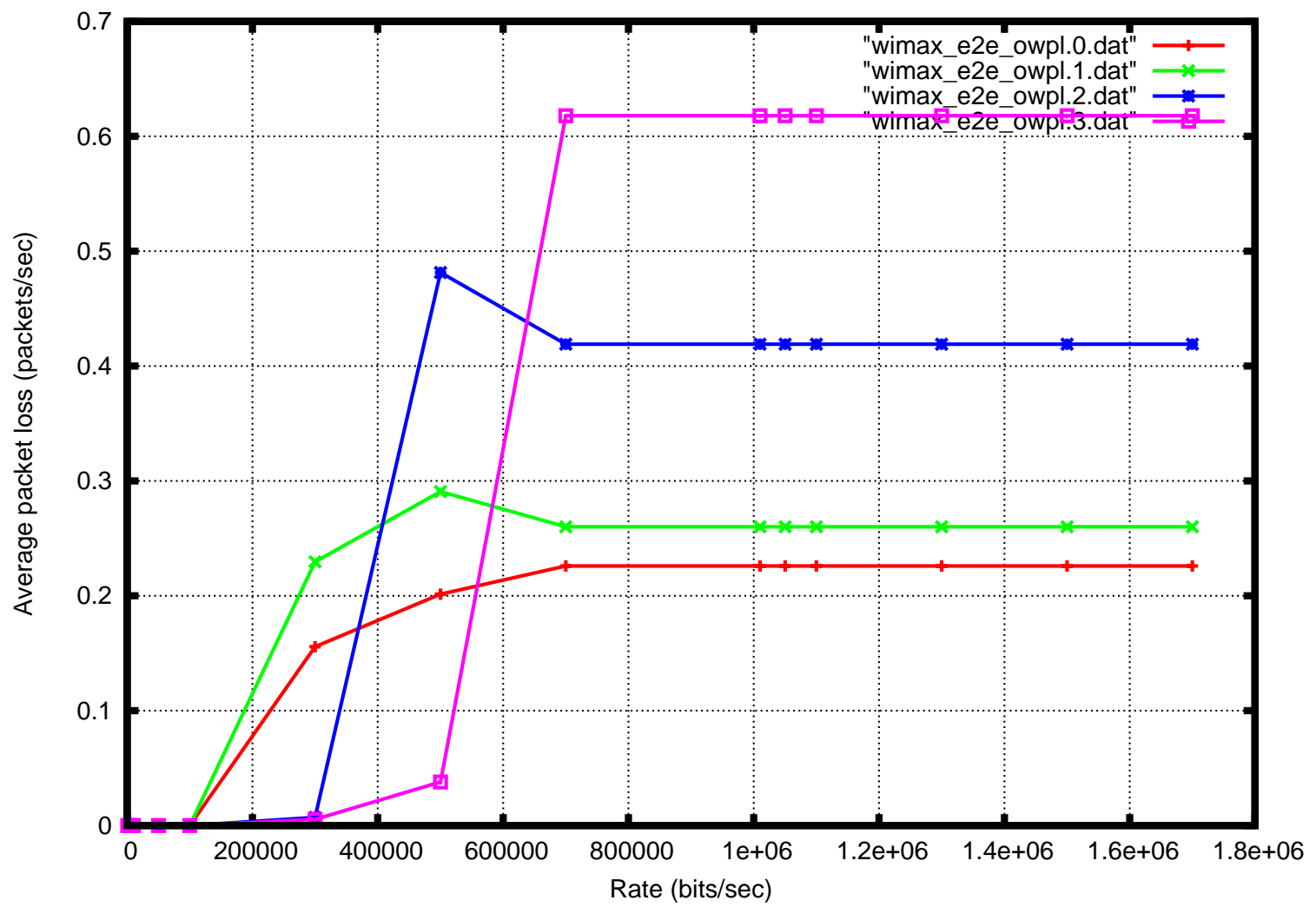
Studio caso basso

Packet loss ACK Reno



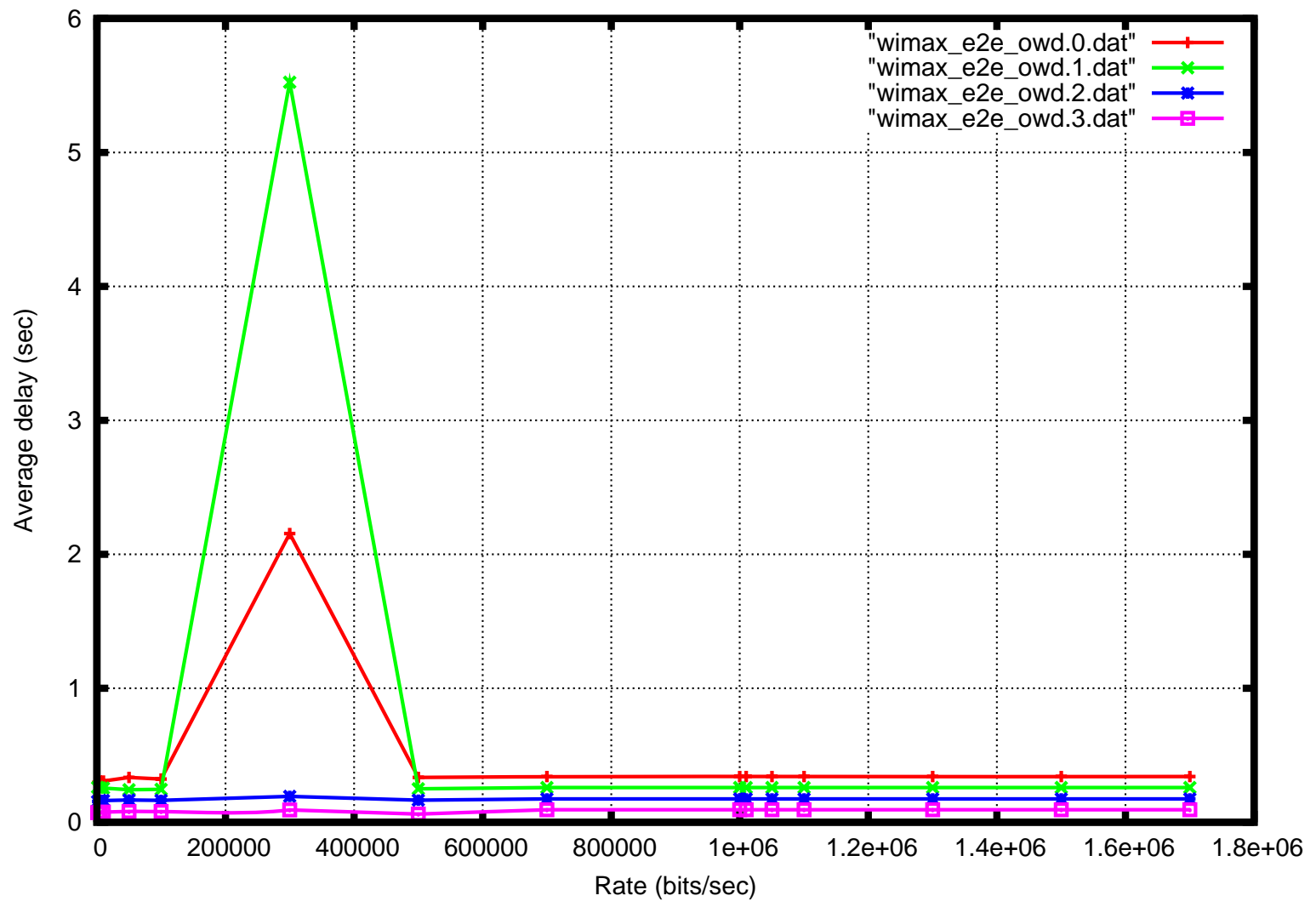
Studio caso basso

Packet loss SACK



Studio caso basso

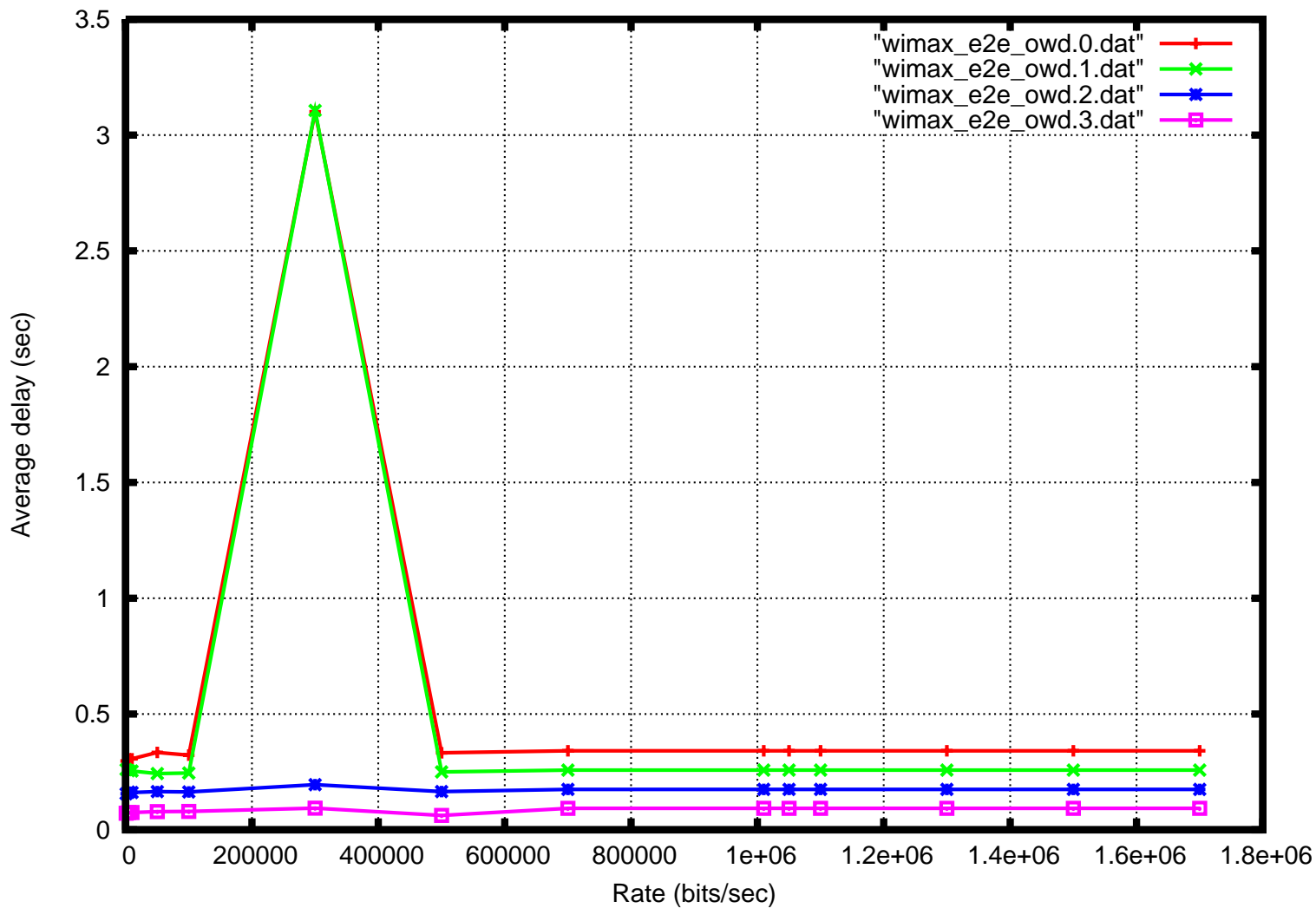
Delay ACK Reno



* Intervalli di confidenza elevati

Studio caso basso

Delay SACK



* Intervalli di confidenza elevati

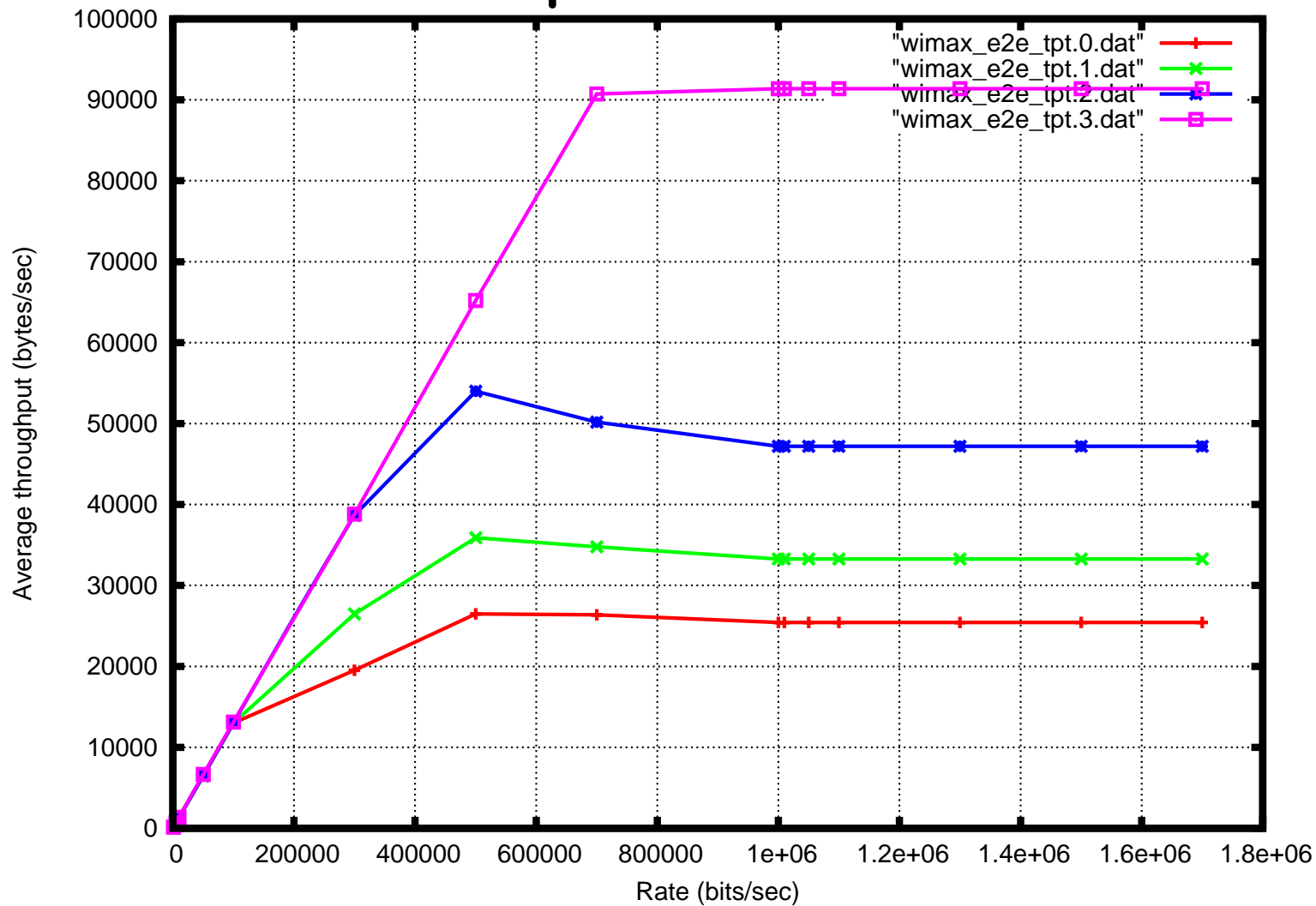
Studio caso basso

TCP Reno: confronto Ack-Sack

- ✓ Rate basso (100000 bits/s)
 - packet loss 0
 - tpt nel caso ack = tpt nel caso sack
 - tpt molto simile tra i flussi
- ✓ Rate intermedio (300000 bits/s)
 - Flussi vicini sono quasi invariati
- ✓ Rate alto (700000 bits/s)
 - Comportamento equivalente al caso FTP

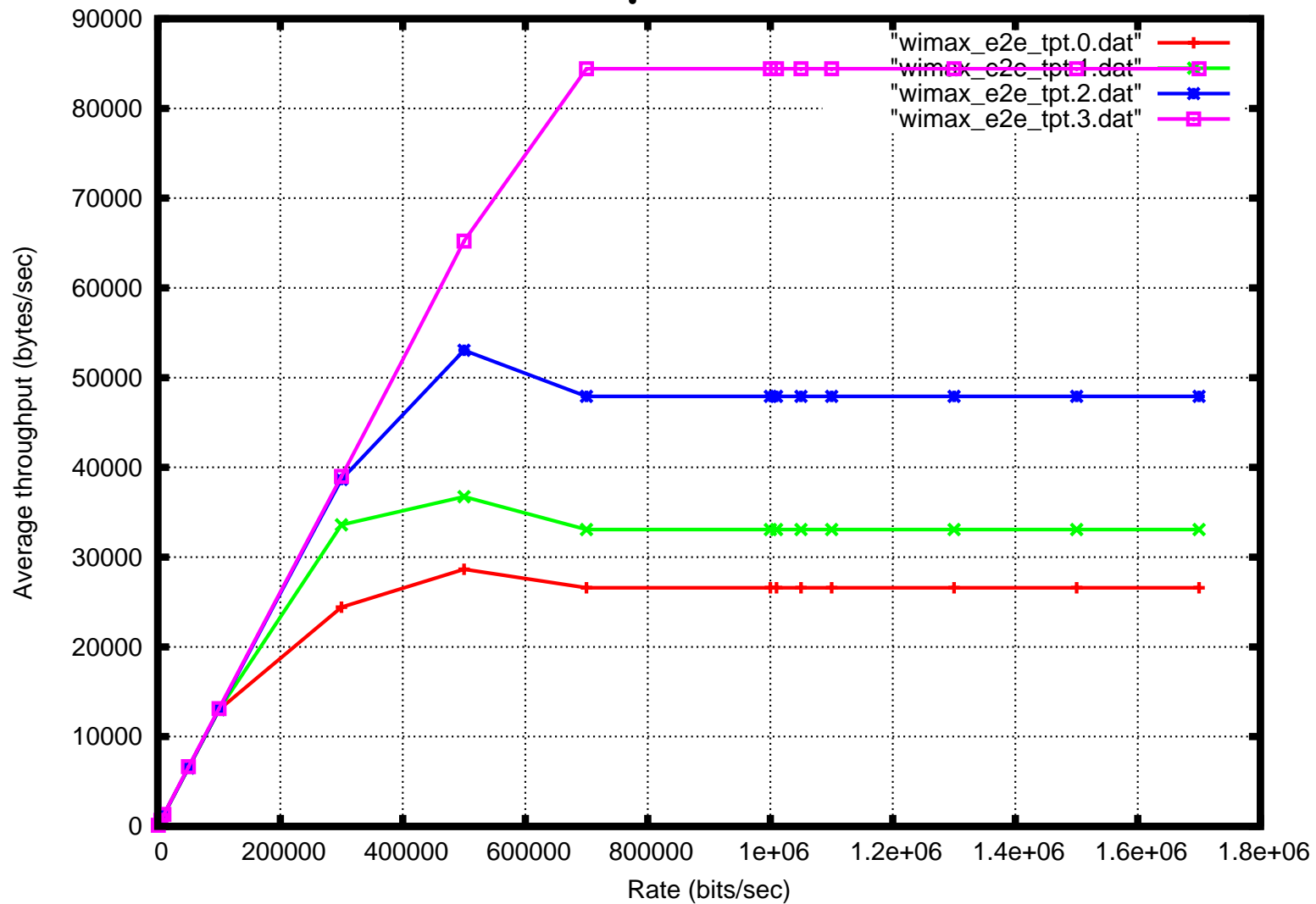
Studio caso basso

Tpt ACK New Reno



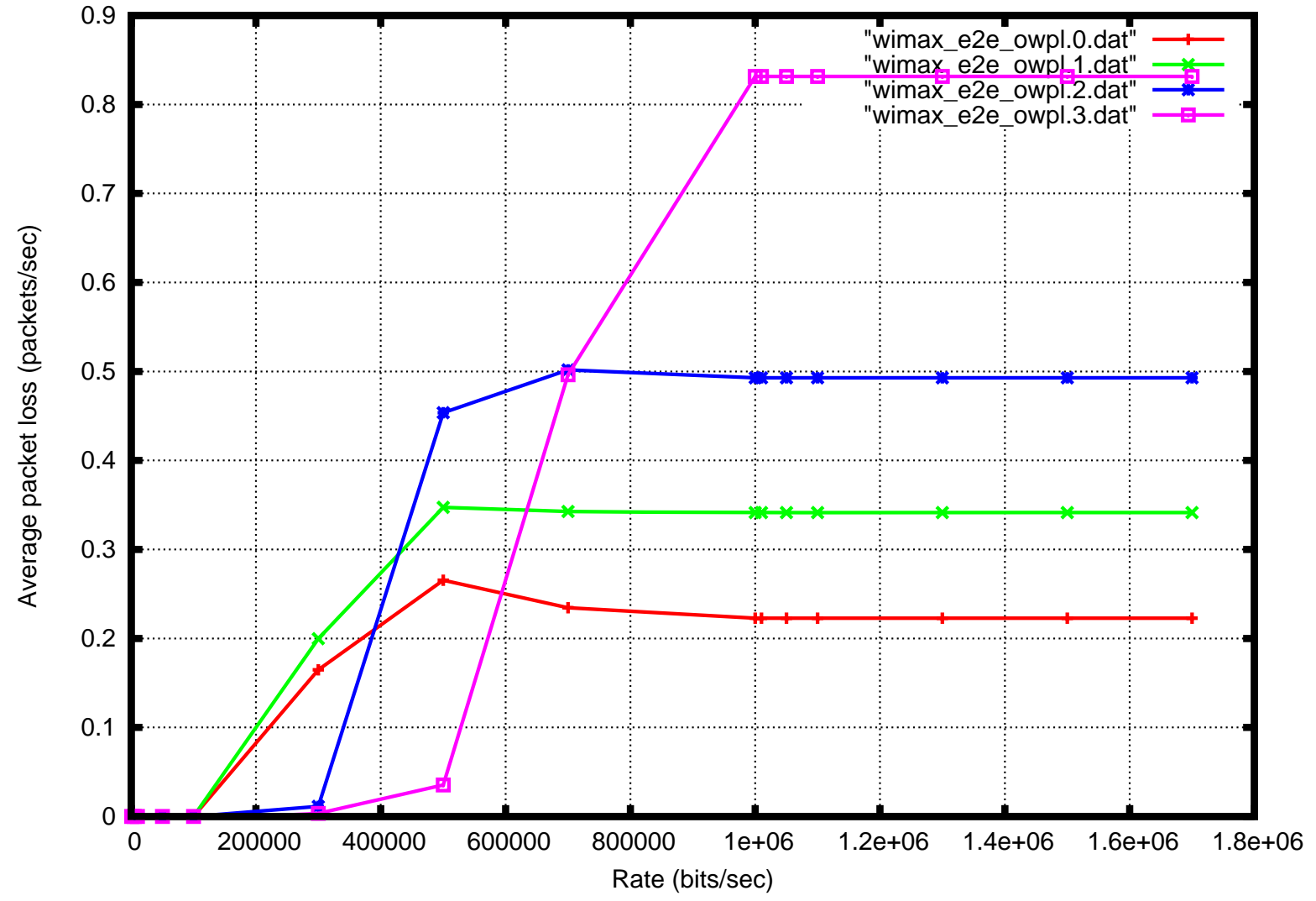
Studio caso basso

Tpt SNACK



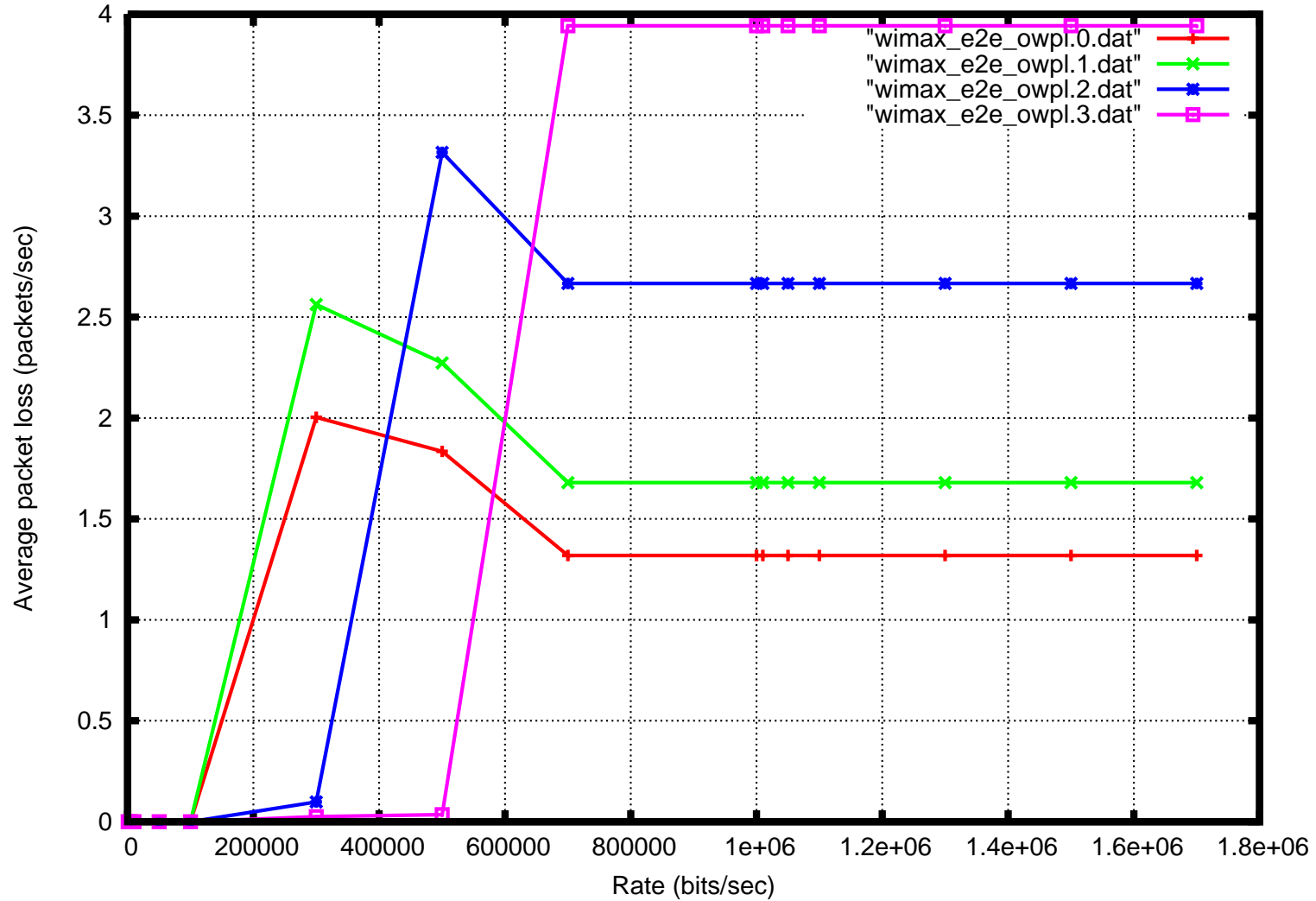
Studio caso basso

Packet loss ACK New Reno



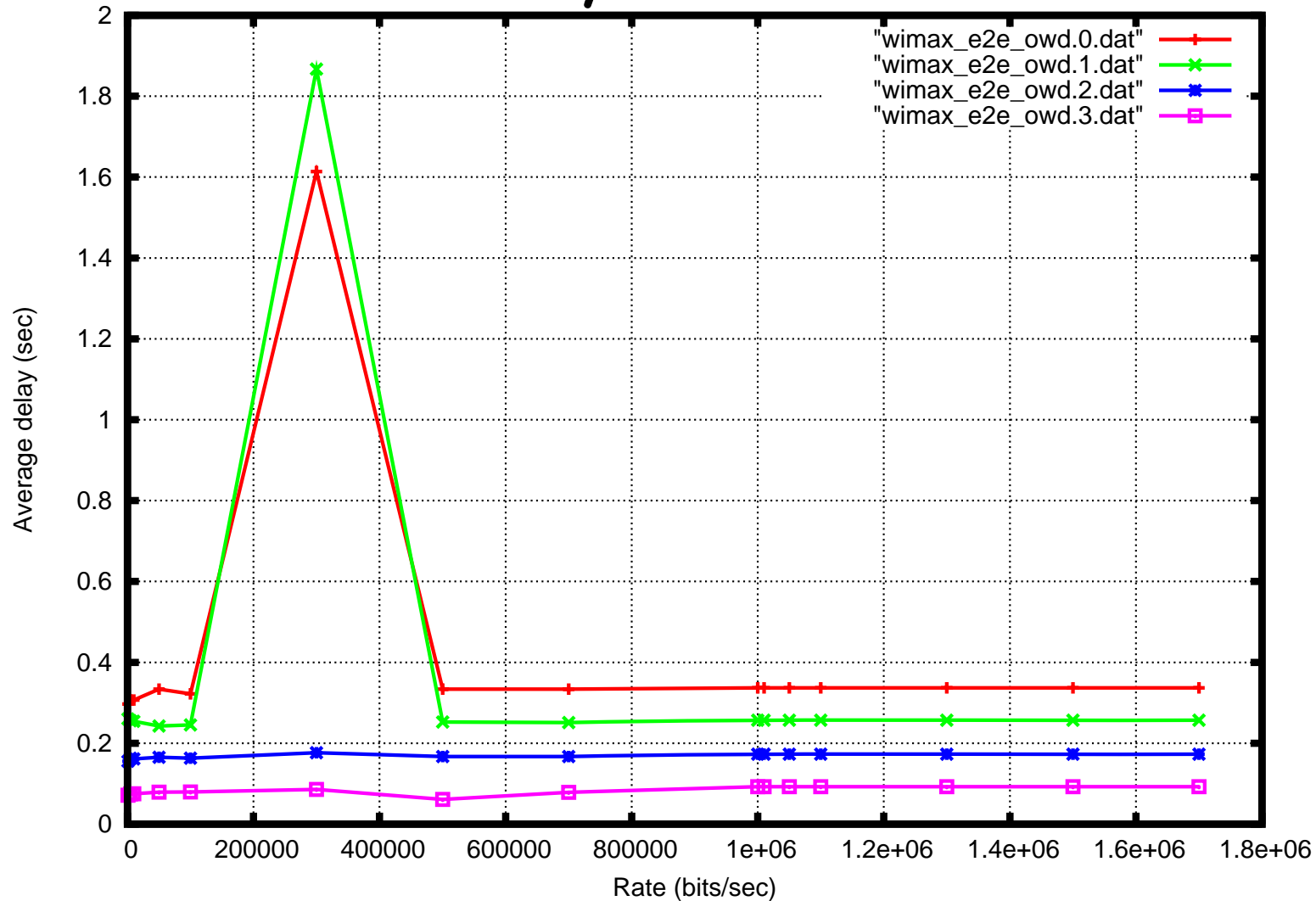
Studio caso basso

Packet loss SNACK



Studio caso basso

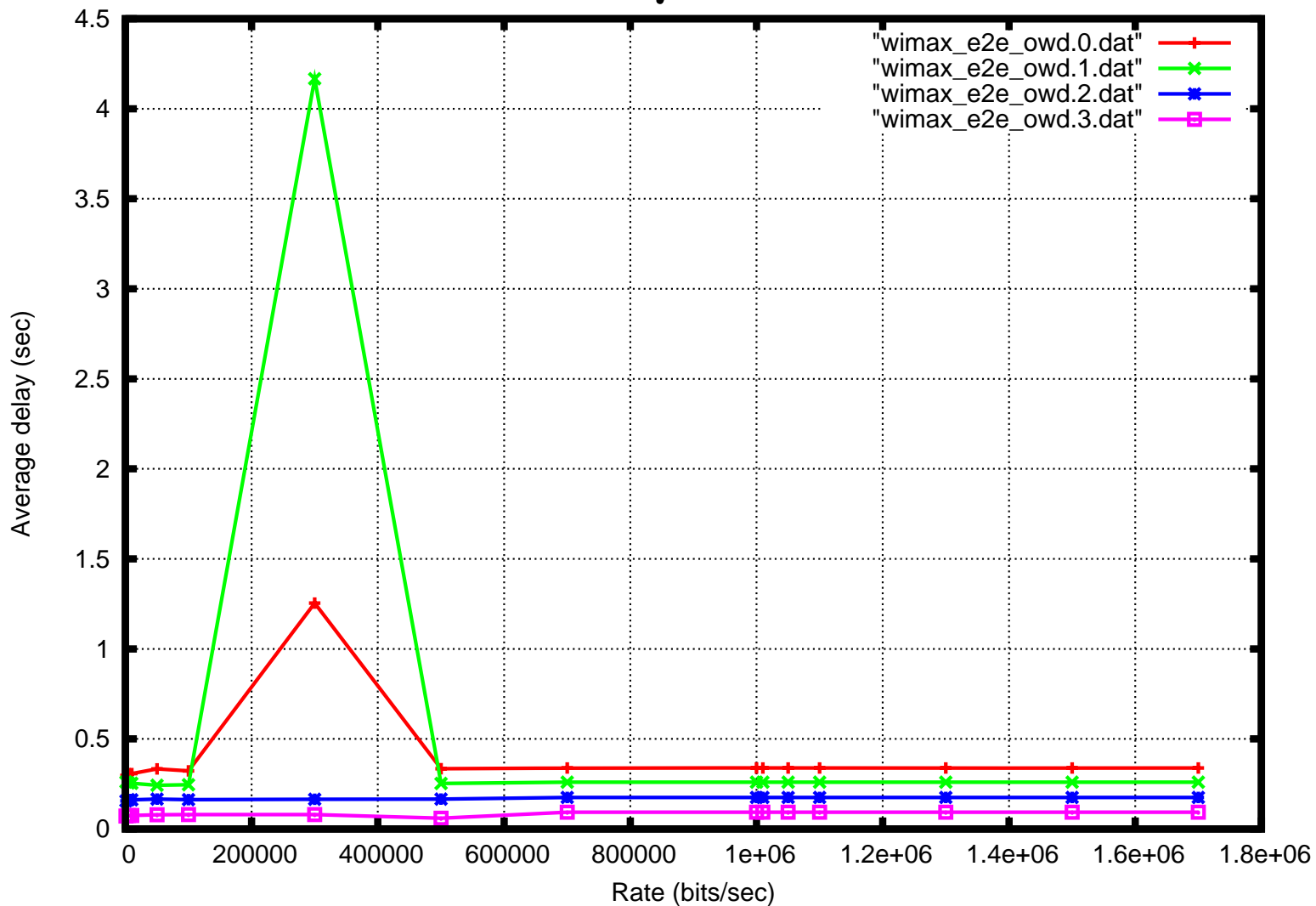
Delay ACK New Reno



* Intervalli di confidenza elevati

Studio caso basso

Delay SNACK



* Intervalli di confidenza elevati

Studio caso basso

TCP New Reno: confronto Ack-Snack

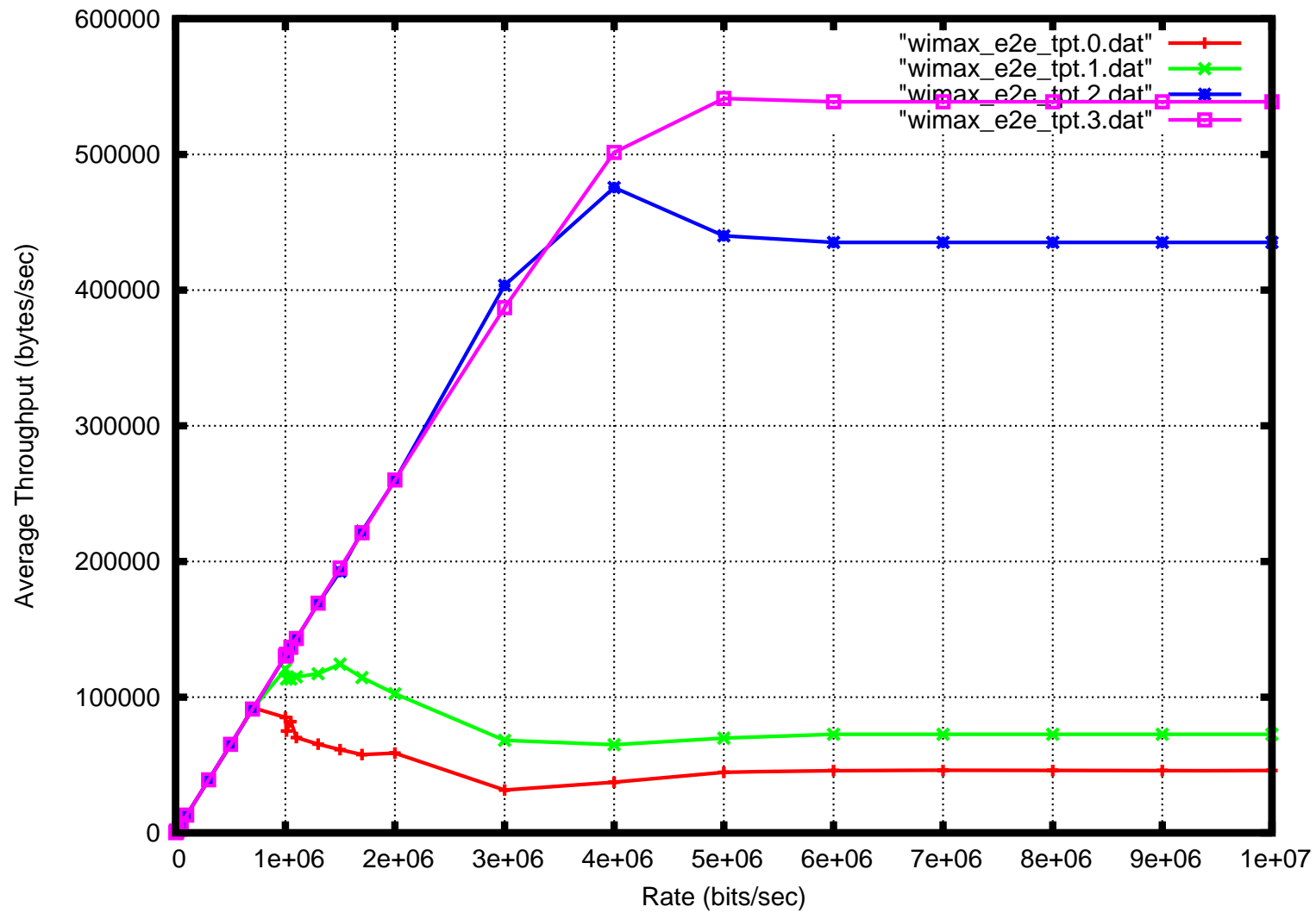
- ✓ Rate basso (100000 bits/s)
 - packet loss 0
 - tpt nel caso ack = tpt nel caso snack
 - tpt molto simile tra i flussi
- ✓ Rate intermedio (300000 bits/s)
 - Con SNACK i flussi lontani migliorano, quelli vicini sono quasi invariati
- ✓ Rate alto (700000 bits/s)
 - Comportamento equivalente al caso FTP

Traffico CBR

Studio caso alto

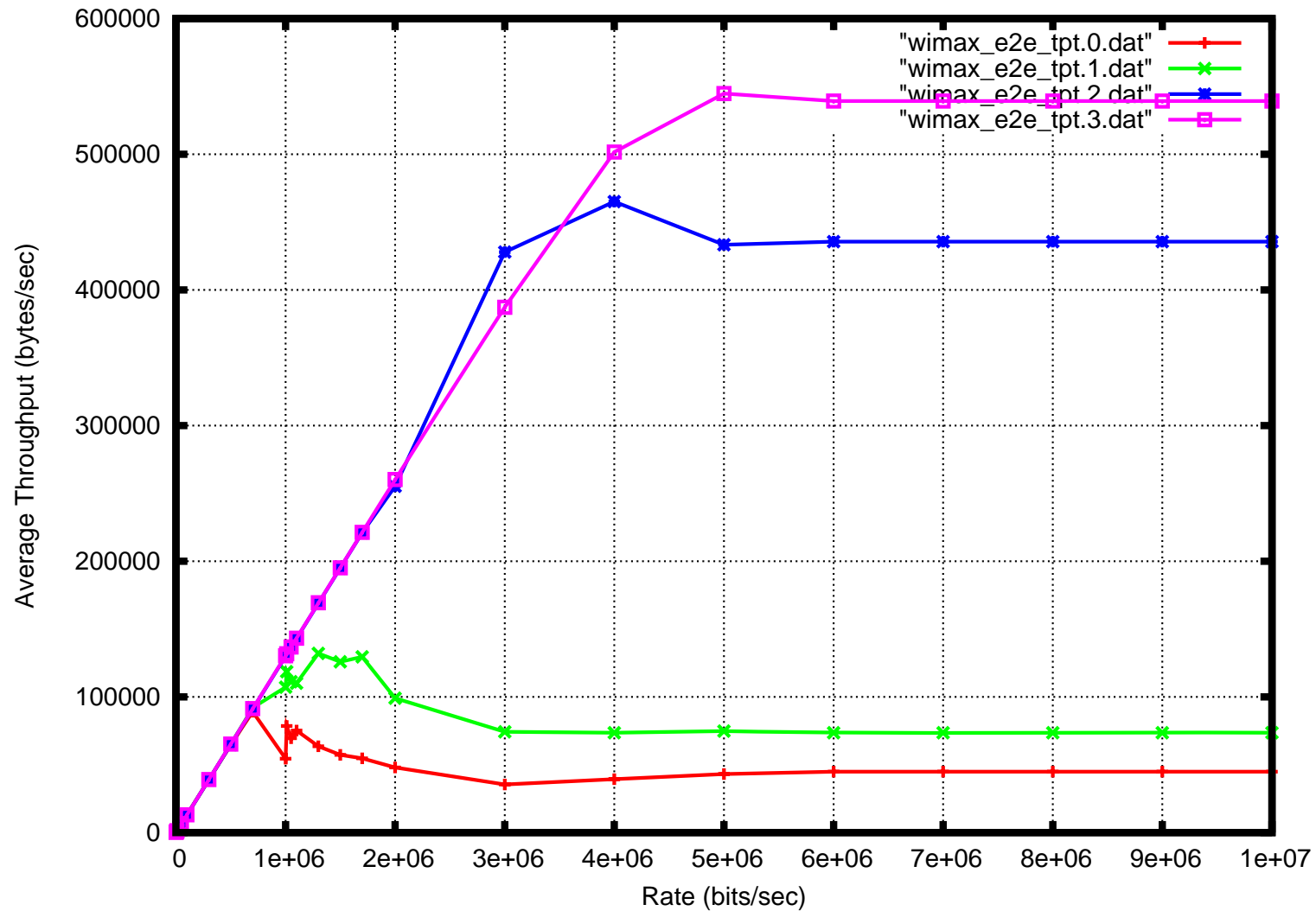
Studio caso alto

Tpt ACK Reno



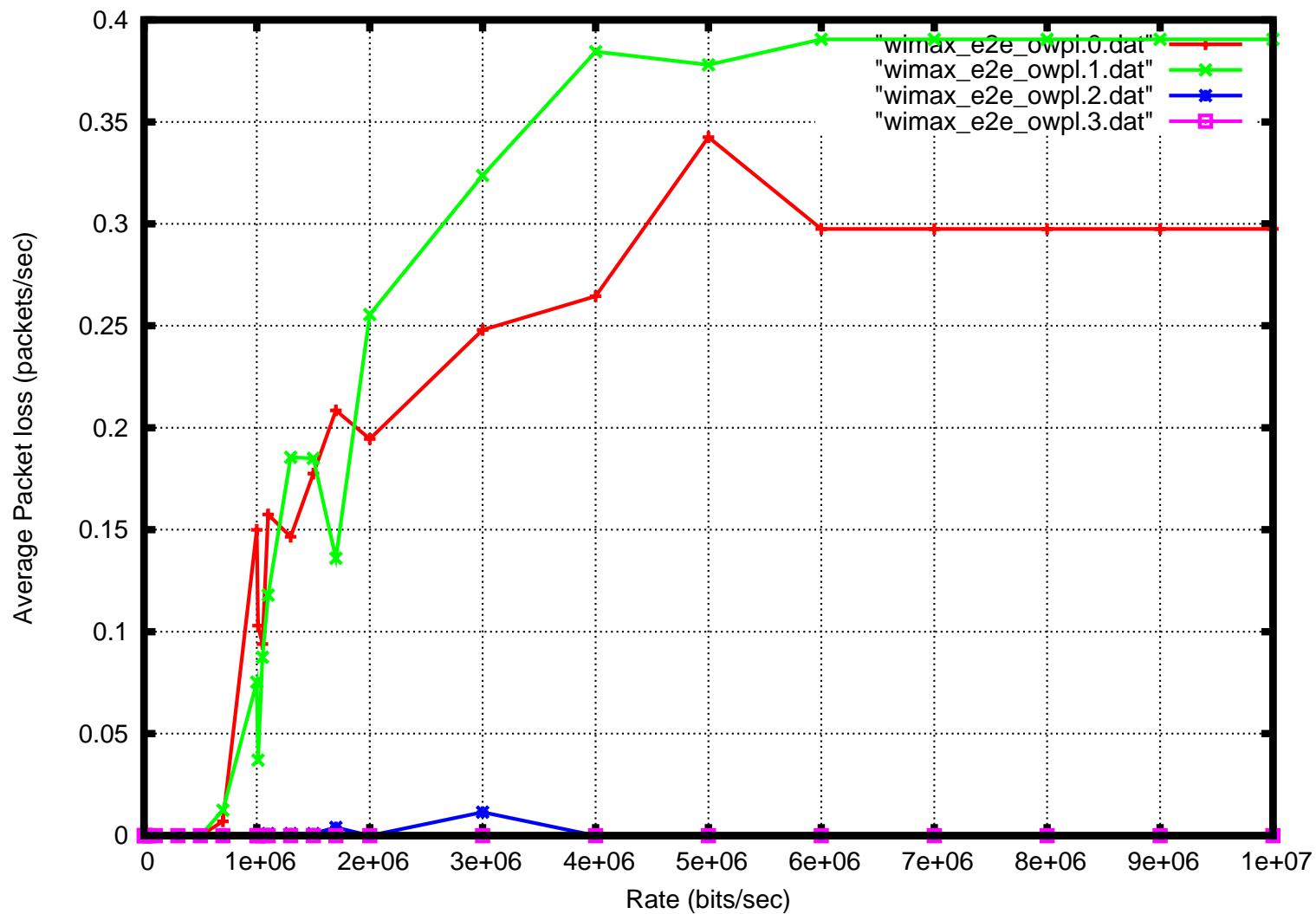
Studio caso alto

Tpt SACK



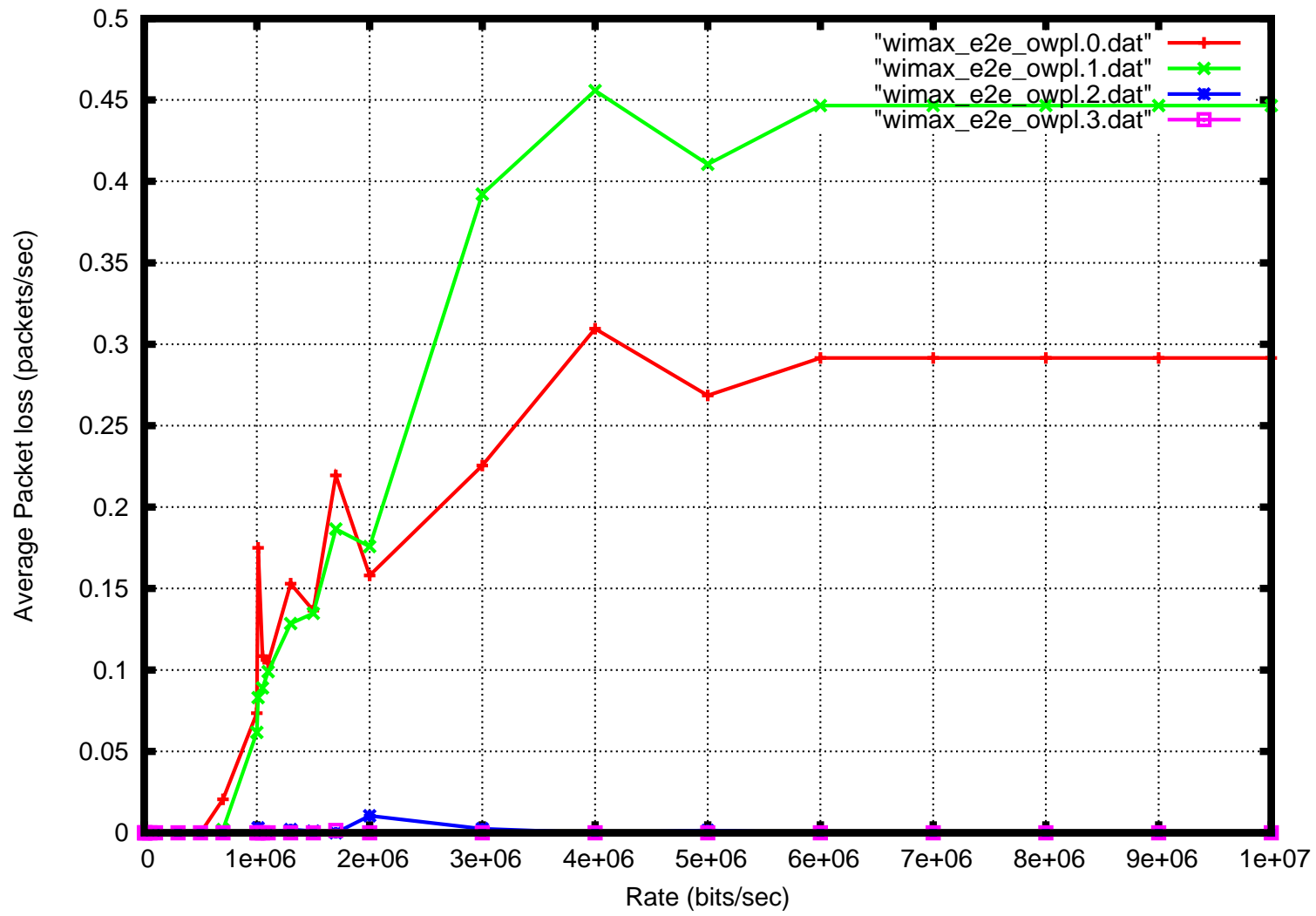
Studio caso alto

Packet loss ACK Reno



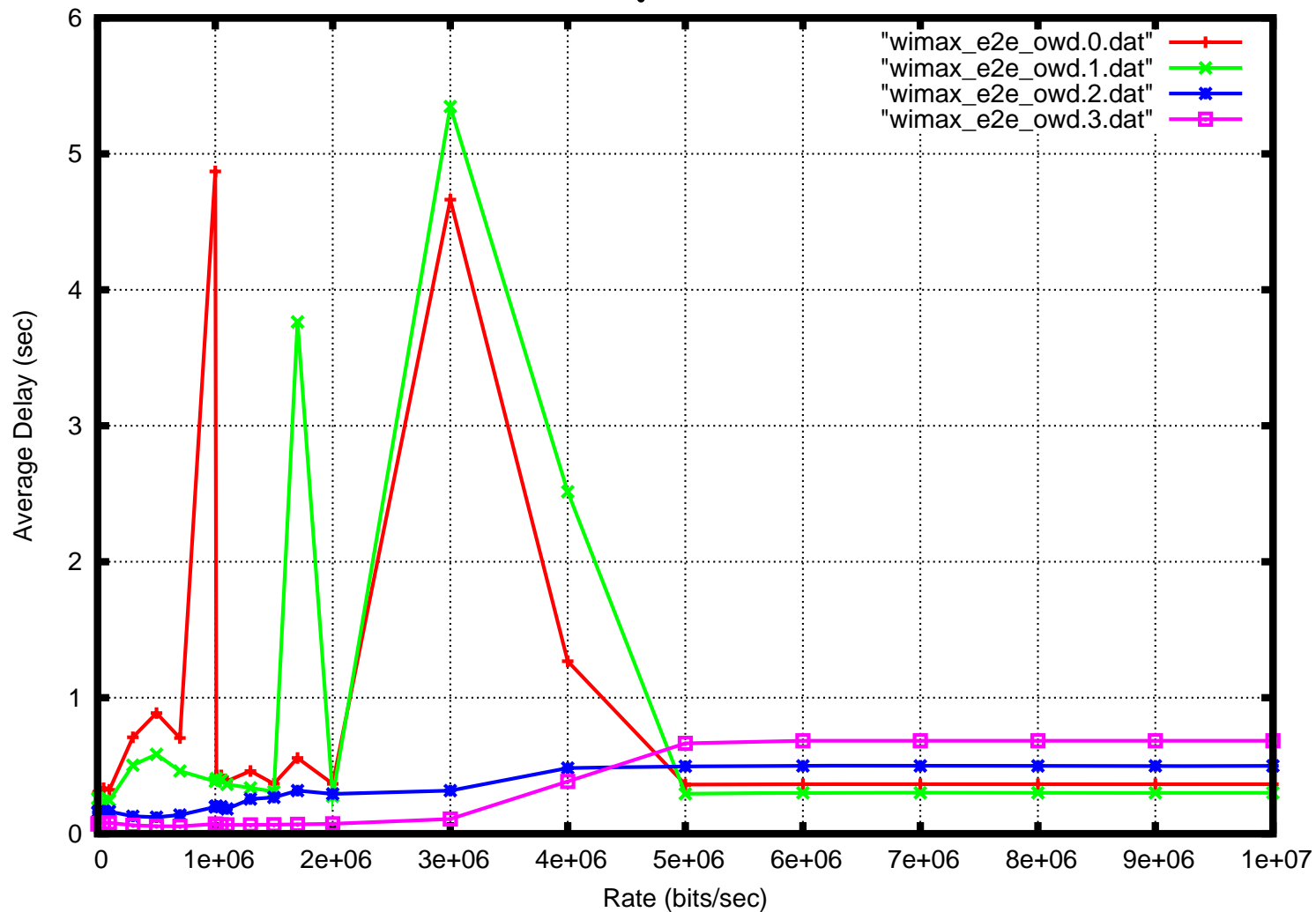
Studio caso alto

Packet loss SACK



Studio caso alto

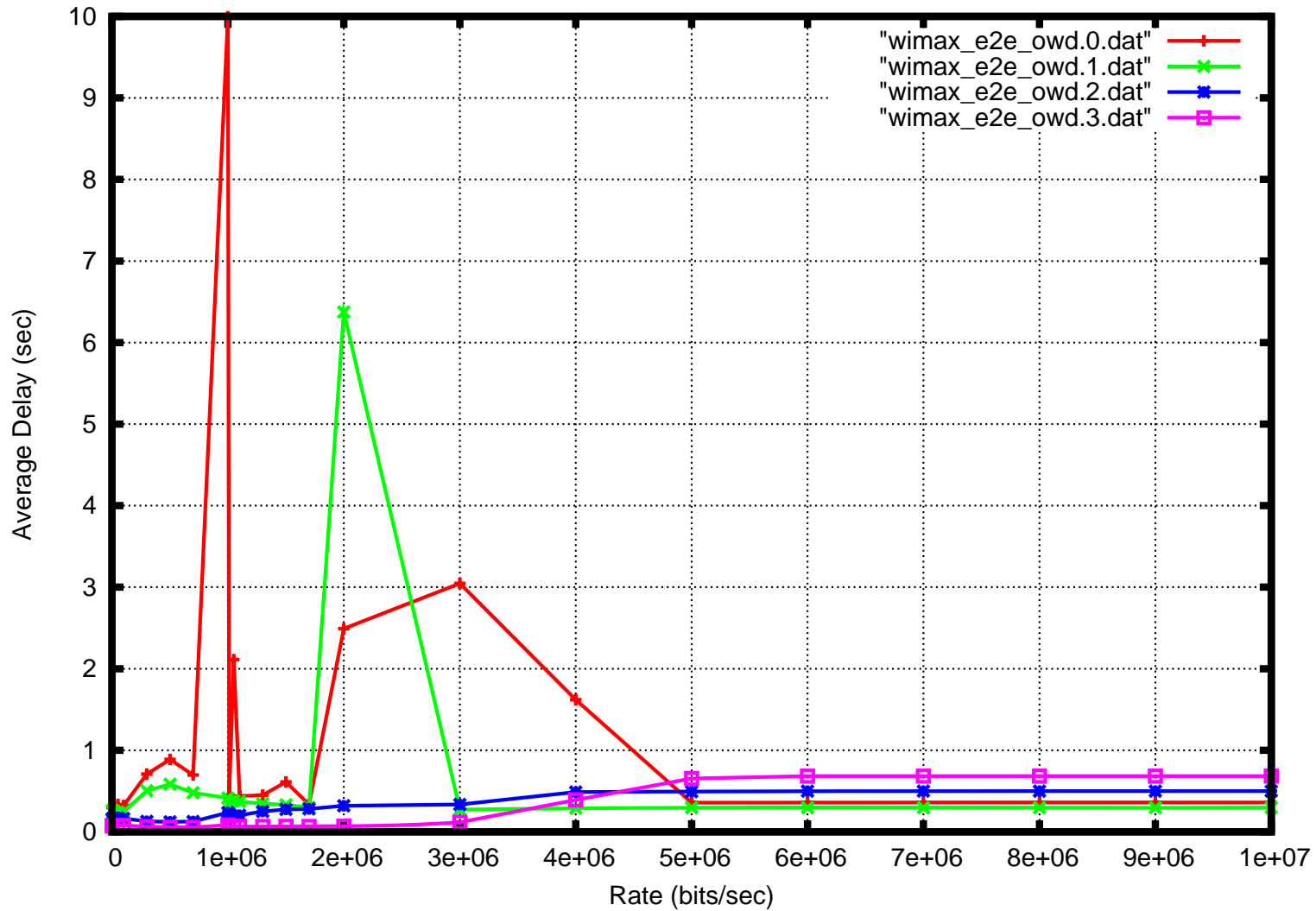
Delay ACK Reno



* Intervalli di confidenza elevati

Studio caso alto

Delay SACK



* Intervalli di confidenza elevati

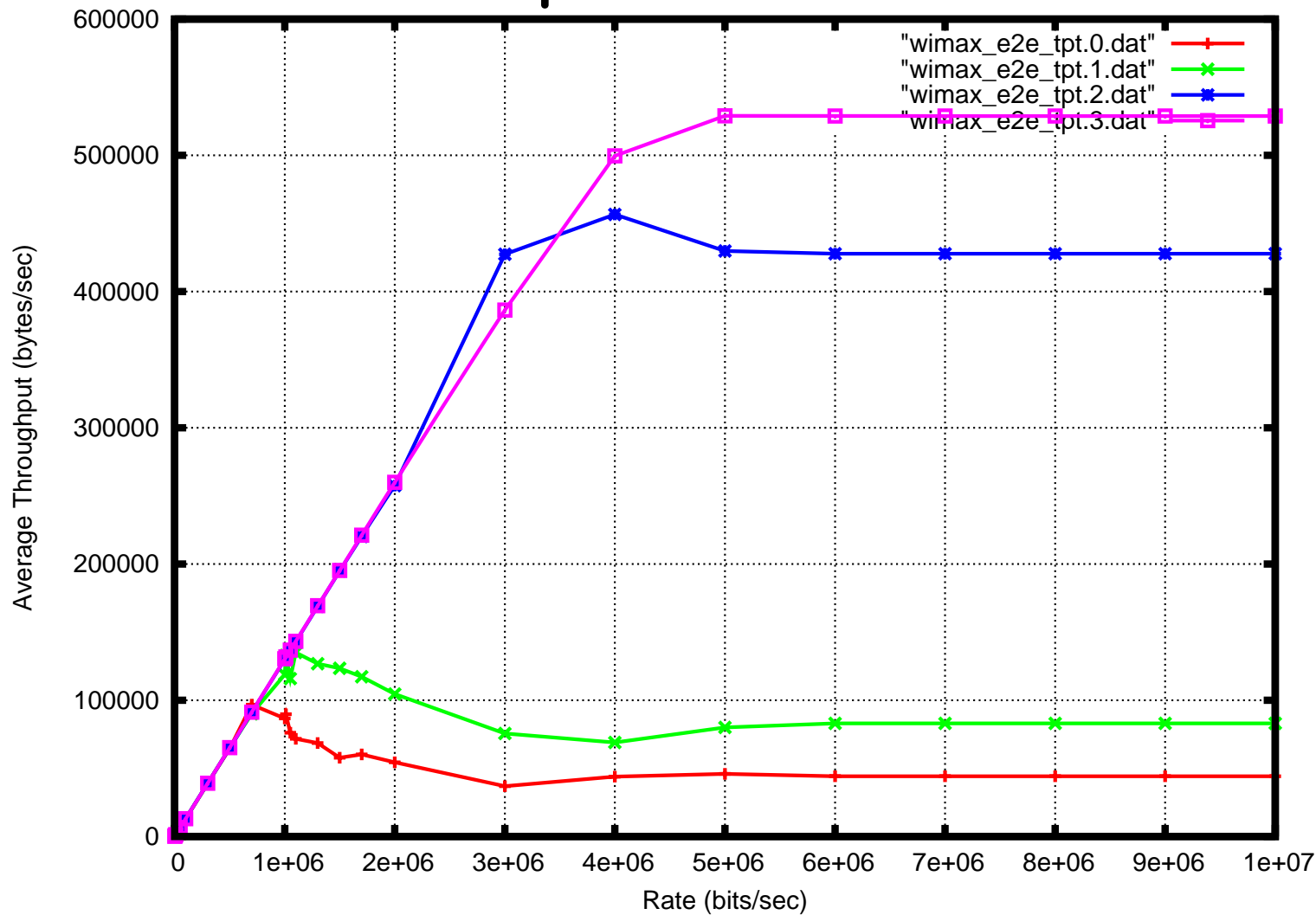
Studio caso alto

TCP Reno: confronto Ack-Sack

- ✓ Rate basso (500000 bits/s)
 - packet loss 0
 - tpt nel caso ack = tpt nel caso sack
 - tpt molto simile tra i flussi
- ✓ Rate intermedio (1700000 bits/s)
 - Flussi vicini sono quasi invariati
- ✓ Rate alto (6000000 bits/s)
 - Comportamento equivalente al caso FTP

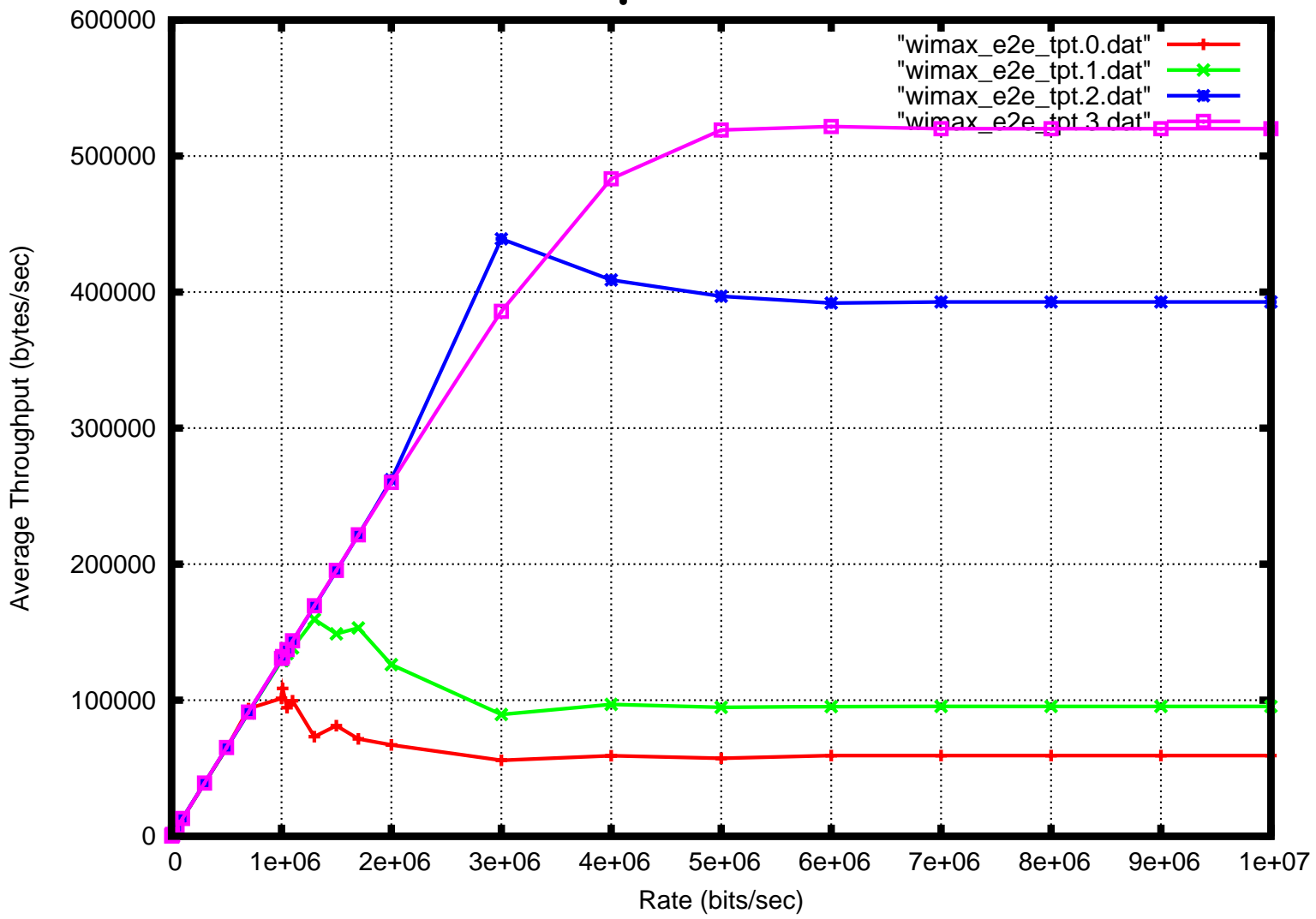
Studio caso alto

Tpt ACK New Reno



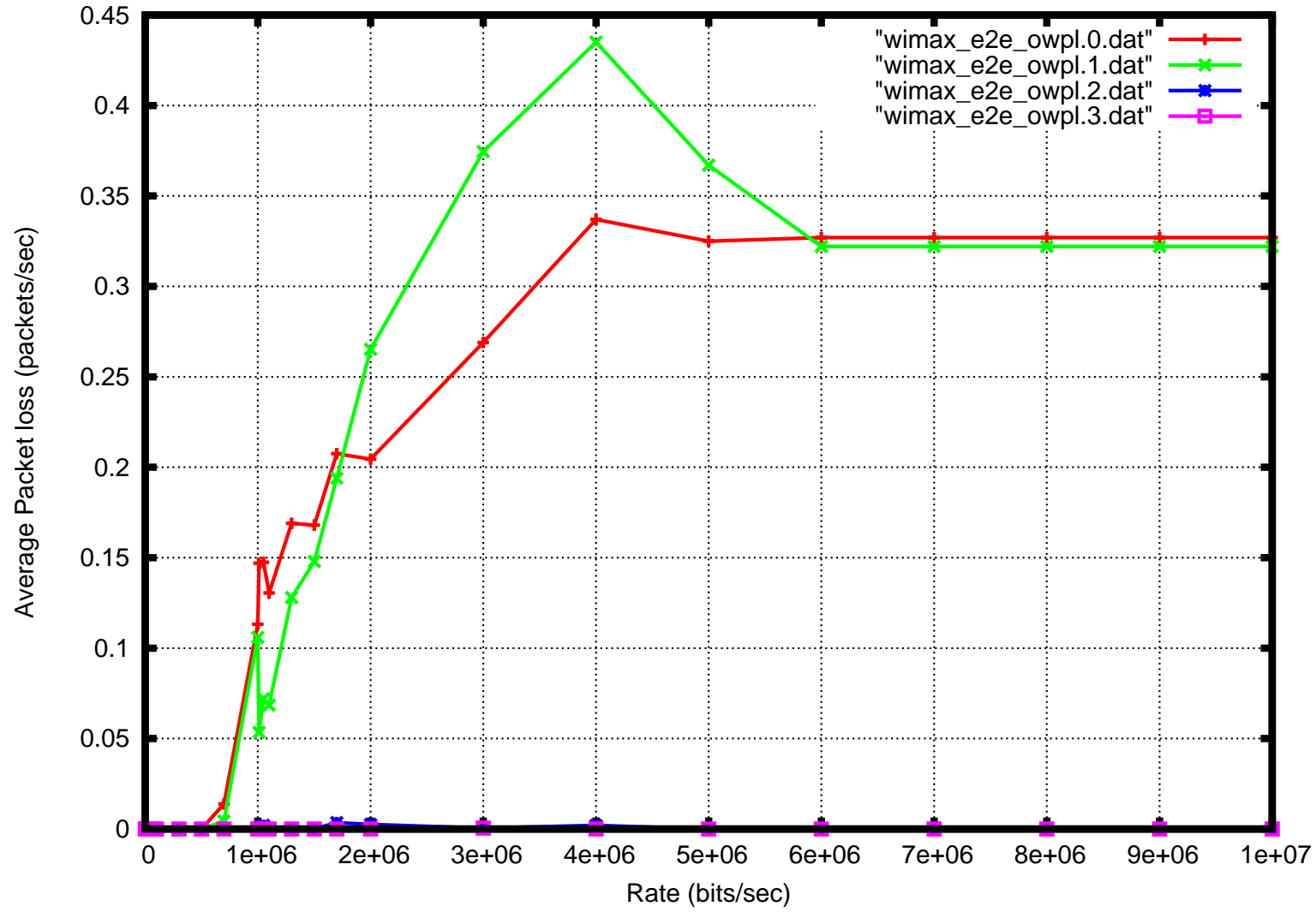
Studio caso alto

Tpt SNACK



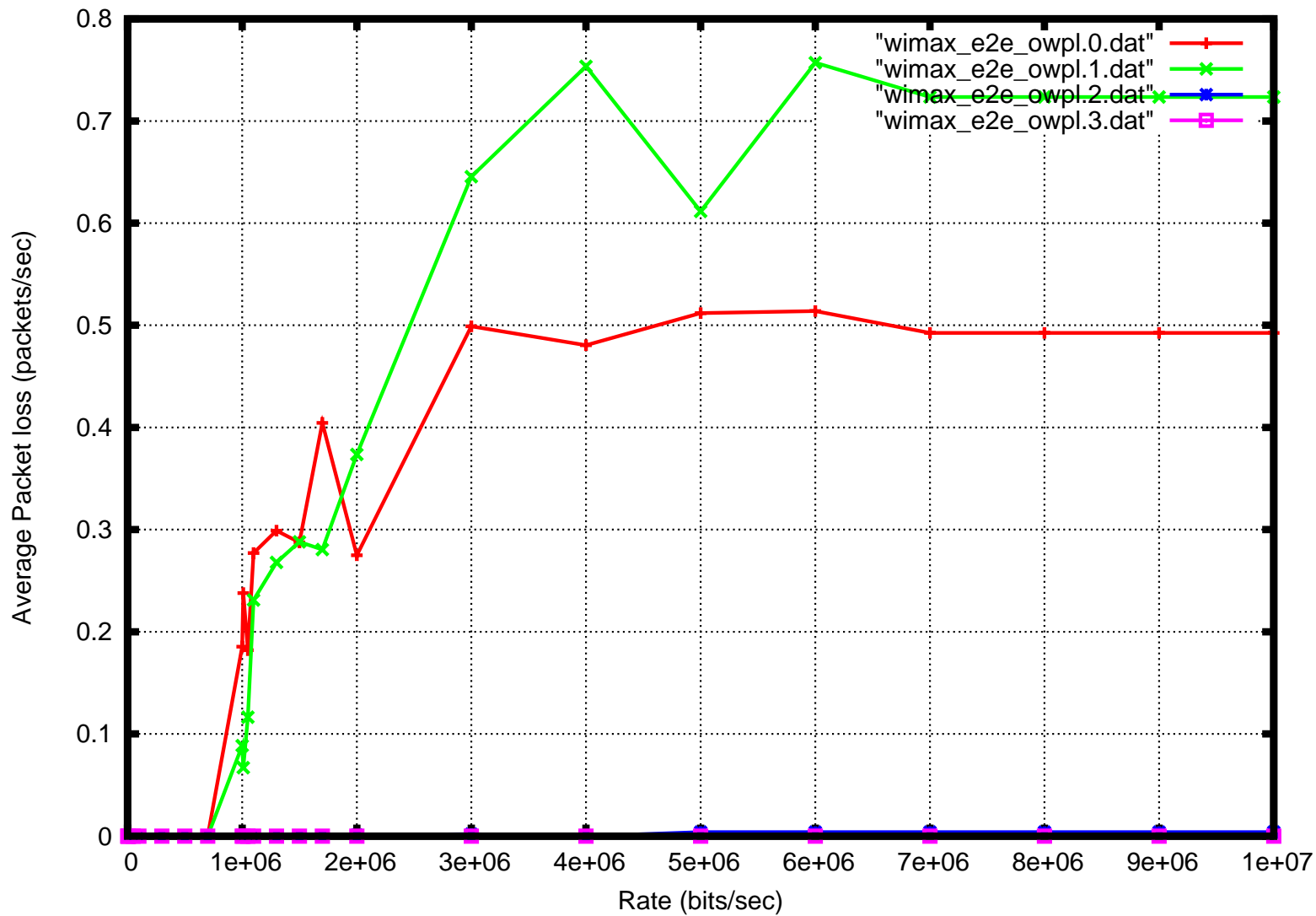
Studio caso alto

Packet loss ACK New Reno



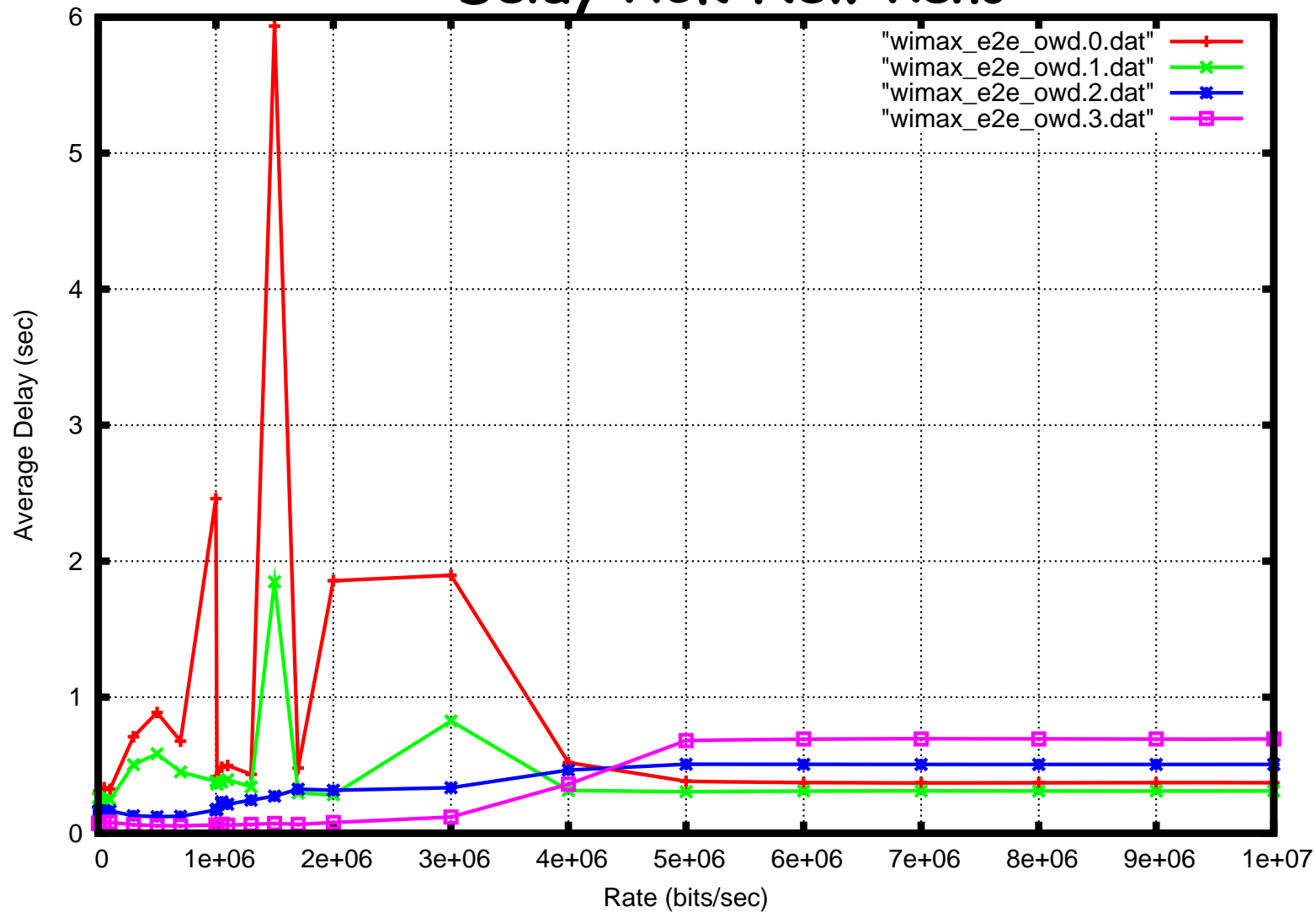
Studio caso alto

Packet loss SNACK



Studio caso alto

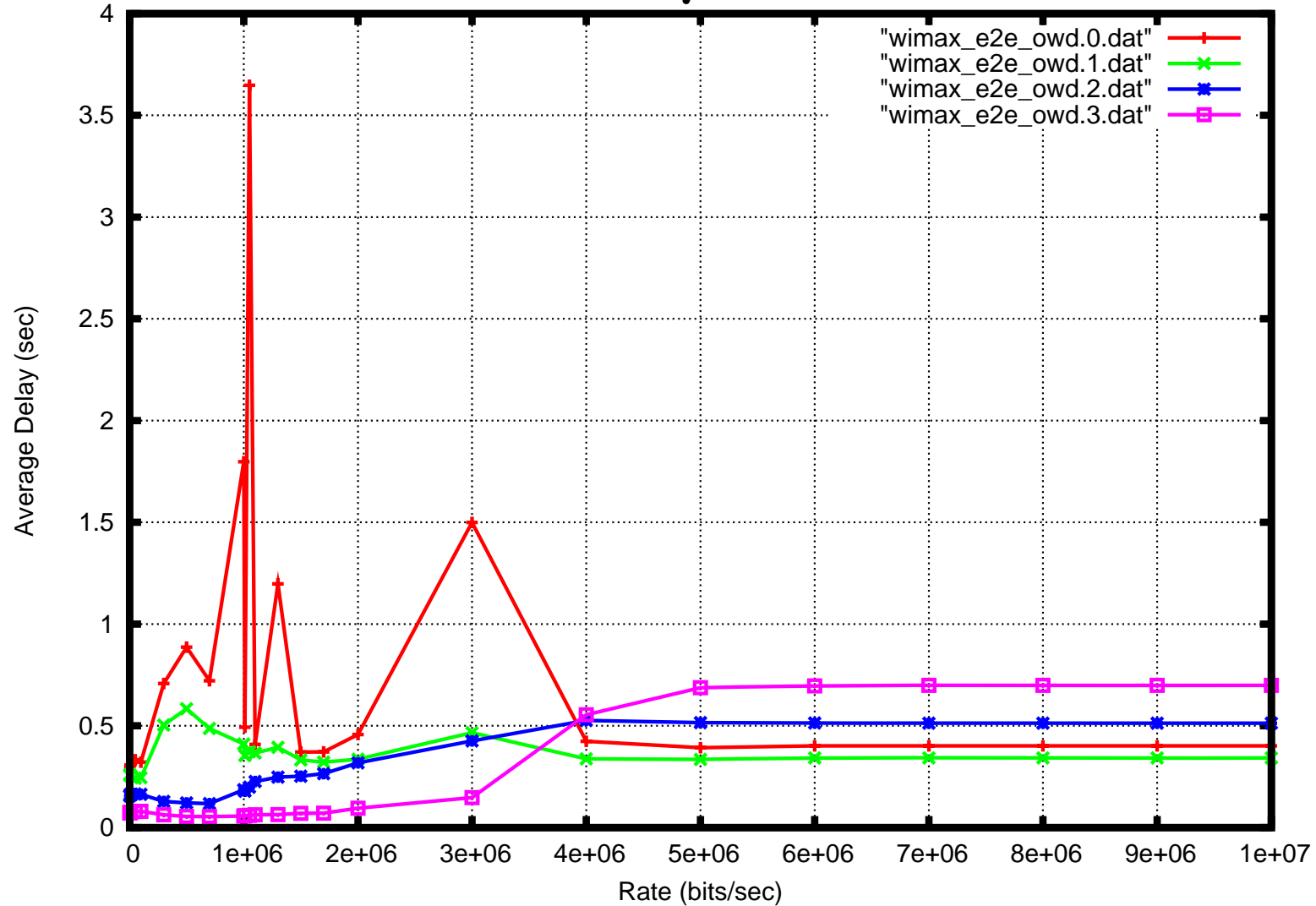
Delay ACK New Reno



* Intervalli di confidenza elevati

Studio caso alto

Delay SNACK



* Intervalli di confidenza elevati

Studio caso alto

TCP New Reno: confronto Ack-Snack

- ✓ Rate basso (500000 bits/s)
 - packet loss 0
 - tpt nel caso ack = tpt nel caso snack
 - tpt molto simile tra i flussi
- ✓ Rate intermedio (1700000 bits/s)
 - Con SNACK i flussi lontani migliorano, quelli vicini sono quasi invariati
- ✓ Rate alto (6000000-7000000 bits/s)
 - Comportamento equivalente al caso FTP

Conclusioni generali

Conclusioni generali

- ✓ Più perdite nella stessa finestra
 - Snack migliora rispetto all'Ack New Reno
 - New Reno migliora rispetto al Reno
- ✓ **New Reno è migliore del Sack** (dato sperimentale)
- ✓ **Snack privilegia i flussi lunghi** (dato sperimentale)
- ✓ Packet loss 0 determina nel CBR:
 - tpt simile tra i flussi,
 - FairRR e rate costante
 - tpt identico nei casi ack, sack, snack