Sistemi Ipermediali
Quality of Service (QoS) - I

Claudio E. Palazzi
Università degli Studi di Padova

Lucidi in parte a cura di: Marco Roccetti, Paola Salomoni, Stefano Ferretti

Quality of Service (QoS)

- *Significato intuitivo*: caratteristiche di sistema che influenzano la qualità dell'applicazione, così come percepita

- "A set of quality requirements on the collective behavior of one or more objects" (Reference Model for Open Distributed Processing - Part 3: Prescriptive Model", International Standard 10746 - 3, ITU - T Recommendation X.903, ITU - ISO, Geneva, 1995)

- "Set of quantitative and qualitative characteristics of a distributed multimedia system necessary to achieve the required functionality of an application" (A. Voegel et al., 1995)

- *Significato pragmatico*: contratto tra l'utente del servizio di rete e l'operatore di rete sulla fornitura di uno "*specifico*" servizio di comunicazione

QoS

- Che significa “specifico”?
- Ad esempio la visione di un utente di rete ATM (a prestazioni garantite) che usufruisse di applicazioni multimediali distribuite potrebbe essere la seguente, espressa in coppie *categoria/*parametro:

  - *Performance/affidabilità*: ritardo end-to-end, percentuale frame perse
  - *Formato*: risoluzione video, frame rate, schema di compressione
  - *Sincronizzazione*: skew tra flussi audio e video
  - *Sicurezza*: autenticazione del cliente, autenticazione delle frame
  - Costo: costi di connessione e trasmissione dati, copyright fees
  - *Percezione dell’utente*: qualità soggettiva di immagini e suoni
  - ….

QoS in MM

- Conseguentemente, i servizi offerti dai sistemi multimediali possono essere parametrizzati in base a certe caratteristiche che essi forniscono, con lo scopo di offrire servizi flessibili e personalizzati
- In generale la QoS definisce quanto buoni sono i servizi offerti (in particolare, nel nostro caso, quelli multimediali)

QoS in MM

- Tipici Requisiti di applicazioni MM distribuite si esprimono in termini di:
- *Prestazioni*: delay, delay jitter, throughput
- *Affidabilità*: percentuale di consegna corretta di messaggi multimediali
  - Questi requisiti possono essere tipicamente in trade off: error rate decrescente può indurre ritardi crescenti (o utilizzo di banda aggiuntiva), se ad esempio sono utilizzati meccanismi basati sulle ritrasmissioni
- *Equità*: network fairness (indipendenza del QoS percepito rispetto alle condizioni di rete di ciascun flusso dati)
- *Costo*: l'utente dovrebbe essere in grado di giudicare i costi proposti dall'erogatore del servizio per una data combinazione di valori relativi ai parametri prestazioni/affidabilità

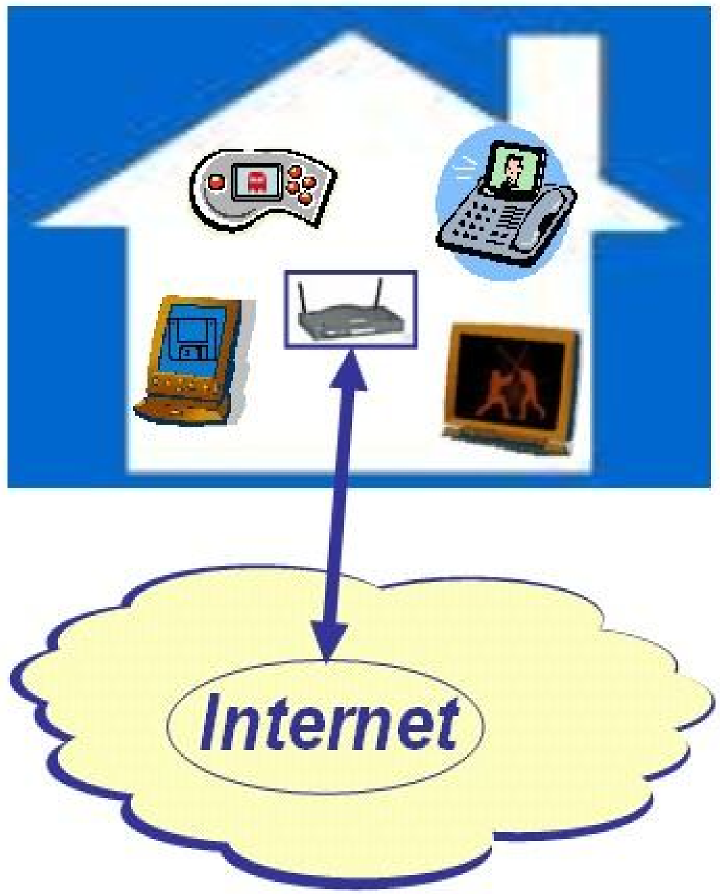
Caso di Studio (CS)

Coesistenza tra flussi eterogenei: analisi dell'impatto sulla QoS delle applicationi

C. E. Palazzi, S. Ferretti, M. Roccetti, G. Pau, M. Gerla

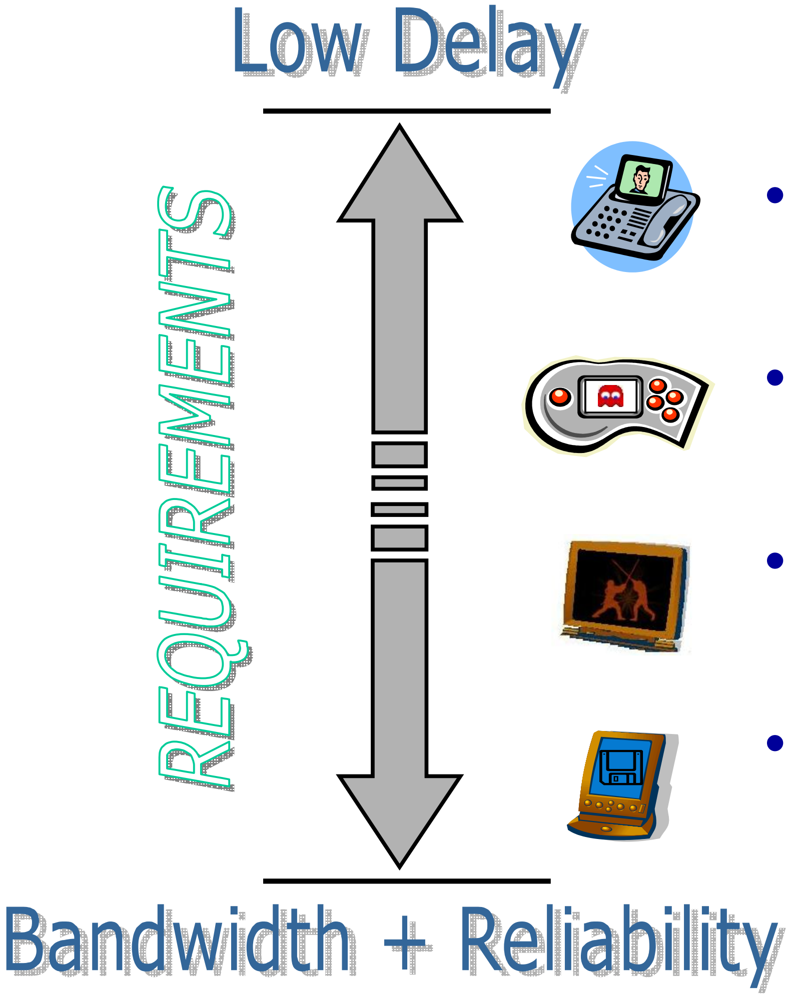
Background: Wireless Home

- Increasing bandwidth
- Connected appliances
- Online entertainment
  - Media Center



Background: Applications

(CS)



- Video Chat
- Online Game
- Video Stream
- Download

A Talk with a Friend

(CS)

"I use IPTV at home

[...]

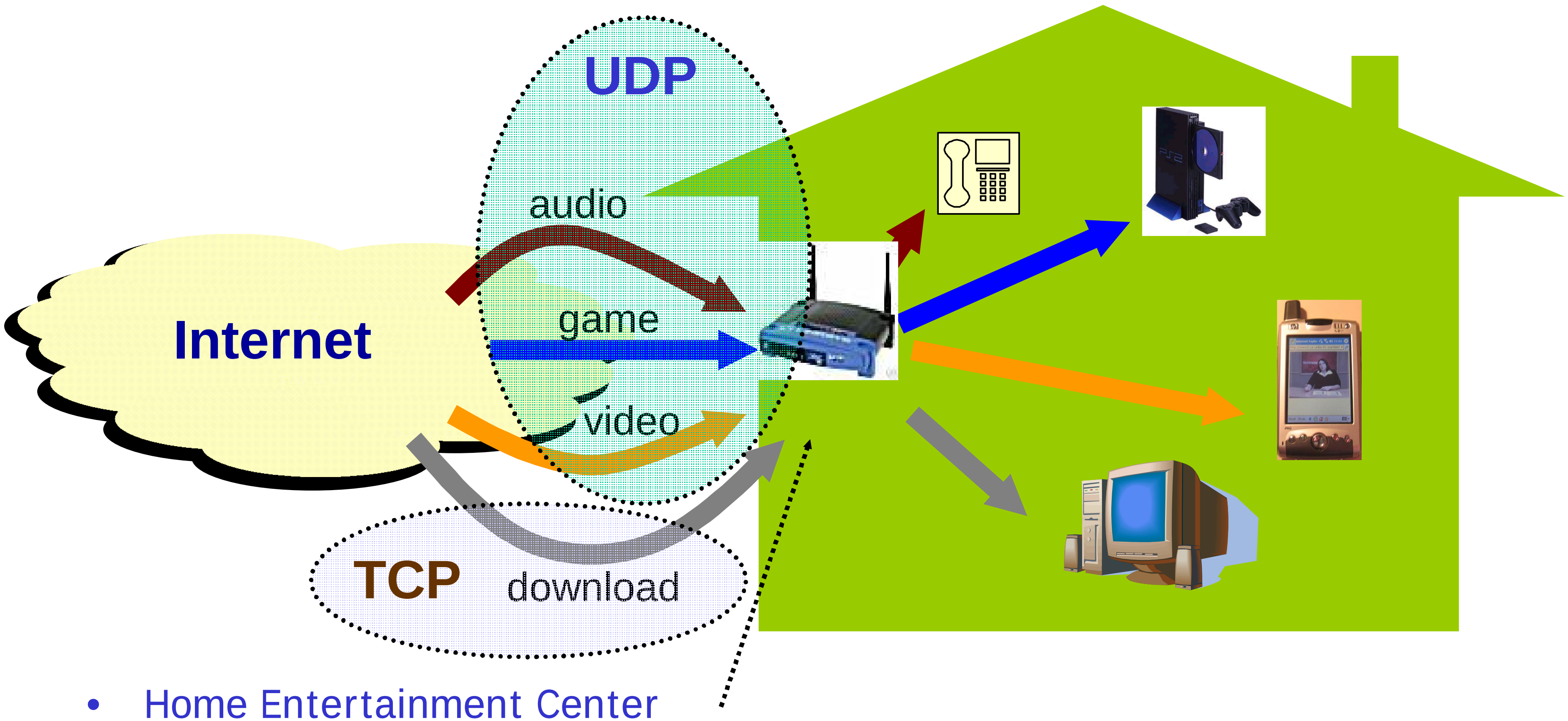
I had to lower my download/upload limit settings on eMule otherwise the video was scattering

[...]

There was no available bandwidth for the TV stream"

In-Home Wireless Scenario: Last Hop

(CS)



- Home Entertainment Center
  - Gateway between in-home devices and the outside world
  - Endowed with an Access Point to guarantee wireless connectivity
  - Shared channel → two nodes cannot transmit at the same moment
  - Often the bottleneck of the network traffic

(CS)

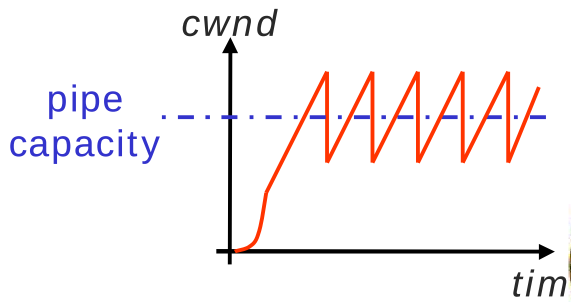
Multiple Streams on a Single Wireless Hop

- Study to observe the impact of several streams that share the same wireless hop
- Network protocols developed assuming mostly TCP-based traffics
- This assumption needs a radical reconsideration when (UDP-based) services for entertainment come into the picture
  - Extremely delay sensitive

TCP Overview

(CS)

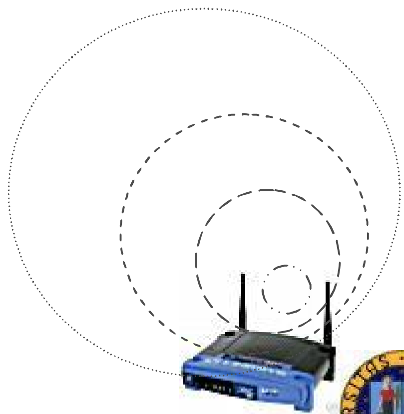
- Window based flow control mechanism
- Continuously probe the link for more bandwidth
- Can fill links and queues with its packets



IEEE 802.11g Overview

- High Bandwidth
  - 54Mbps nominal, ~20Mbps effective

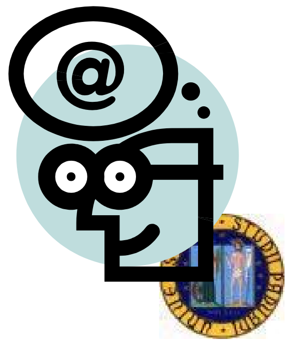
- Retransmission mechanism
  - hides wireless losses
  - increases delays



Aim of the Work **(CS)**

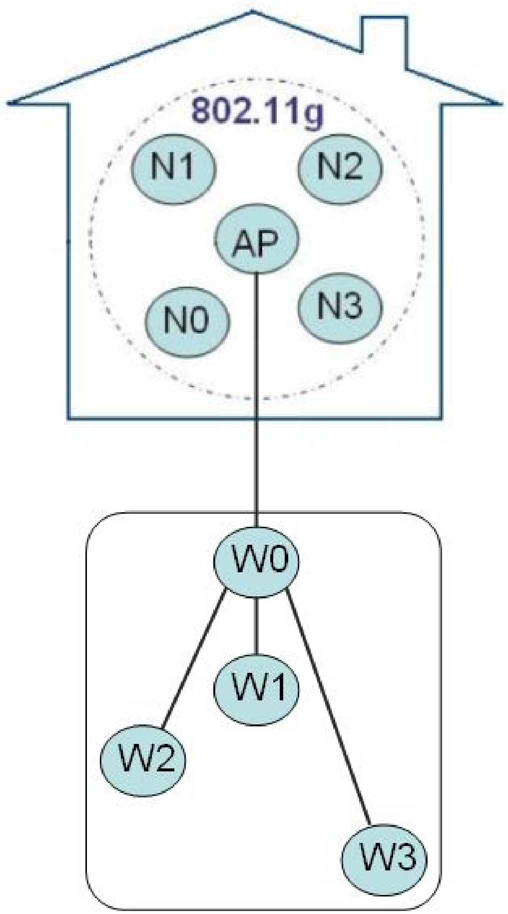
- Analyze the coexistence issues among TCP-based and UDP-based flows
  - Impact of TCP’s congestion avoidance on (UDP-based) real time flows

- Evaluate the interference among Wireless MAC and Transport Protocols
  - Impact of MAC Layer buffers and retransmissions on…
    - ✓ real-time applications (Jitter)
    - ✓ Best effort application (Throughput)



Simulation Environment: NS-2

(CS)



Node 1	Node 2	Delay	Capacity	Queue Size
W1	W0	10ms	100Mbps	140pkts
W2	W0	20ms	100Mbps	140pkts
W3	W0	30ms	100Mbps	140pkts
W0	AP	10ms	100Mbps	140pkts

SIMULATION CONFIGURATION (WIRED LINKS)

From	To	Type	Transport Protocol	Start	End
AP	N0	Movie Stream	UDP	0s	180s
W1	N1	Game Traffic	UDP	45s	180s
N1	W1	Game Traffic	UDP	46s	180s
W2	N2	Video Chat	UDP	90s	180s
N2	W2	Video Chat	UDP	91s	180s
W3	N3	FTP	TCP	135s	180s

SIMULATED APPLICATION LAYER TRAFFIC FLOWS

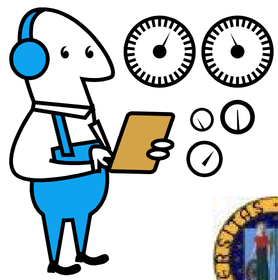
Parameters

(CS)

MAC parameters

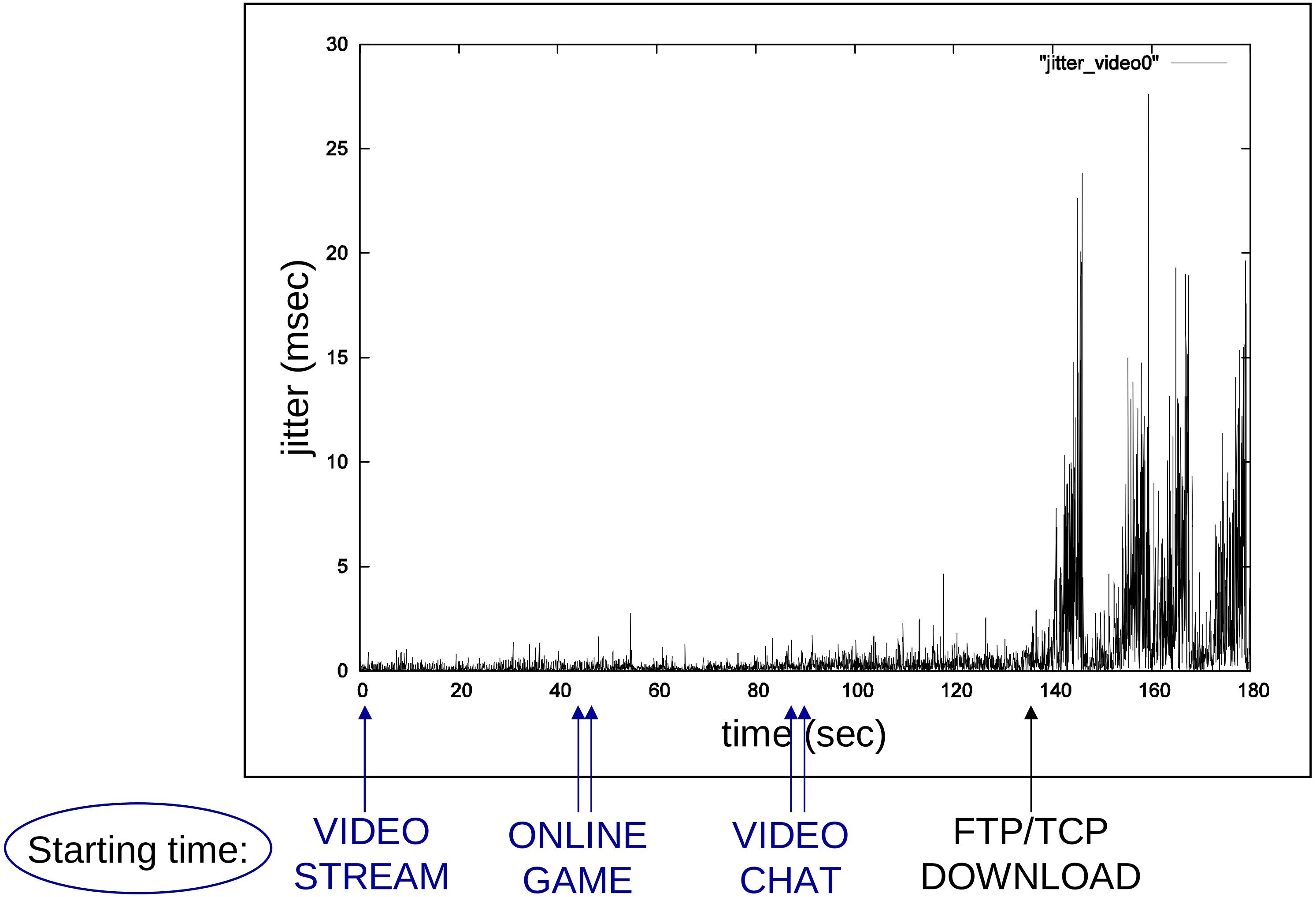
Parameter	Values	Comments
MAC data retransmissions	1, 2, 3, 4	default value = 4
MAC queue size (pkts)	50, 100	common values
Shadowing deviation	7, 9	medium, high
User-AP distance (m)	5, 10	same or other room

Environment parameters



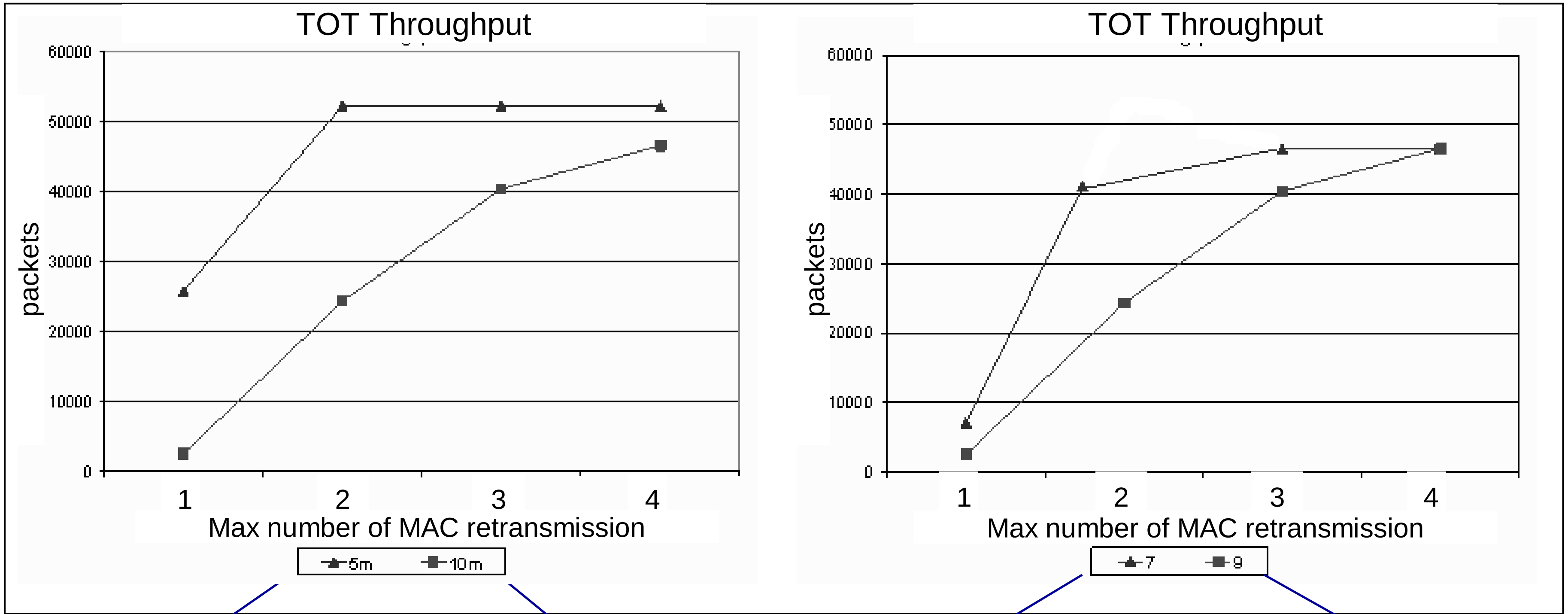
FTP Impact on Real-Time Applications

(CS)



Signal Attenuation vs Throughput

(CS)



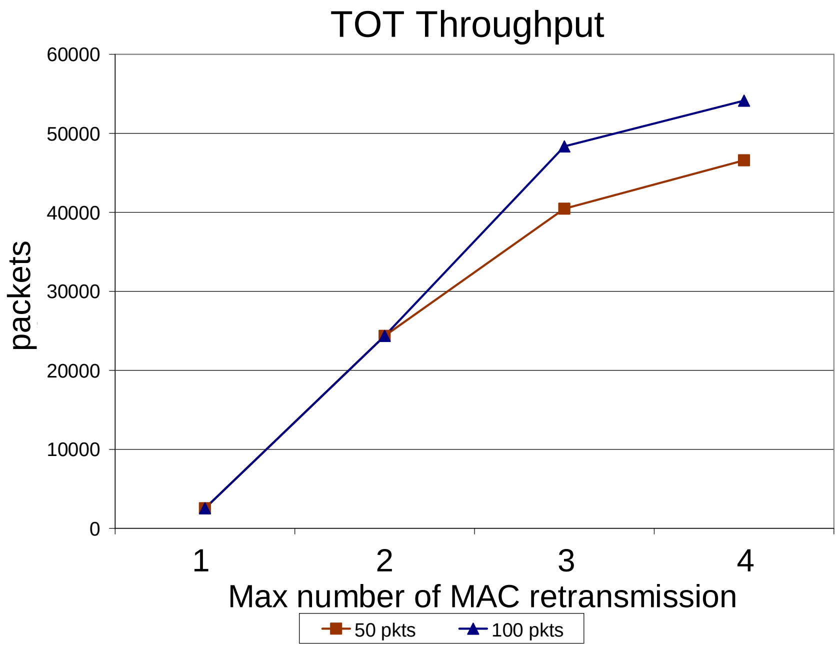
Distance between AP and device (m): 5 vs 10

Signal attenuation parameter (shadowing): 7 vs 9

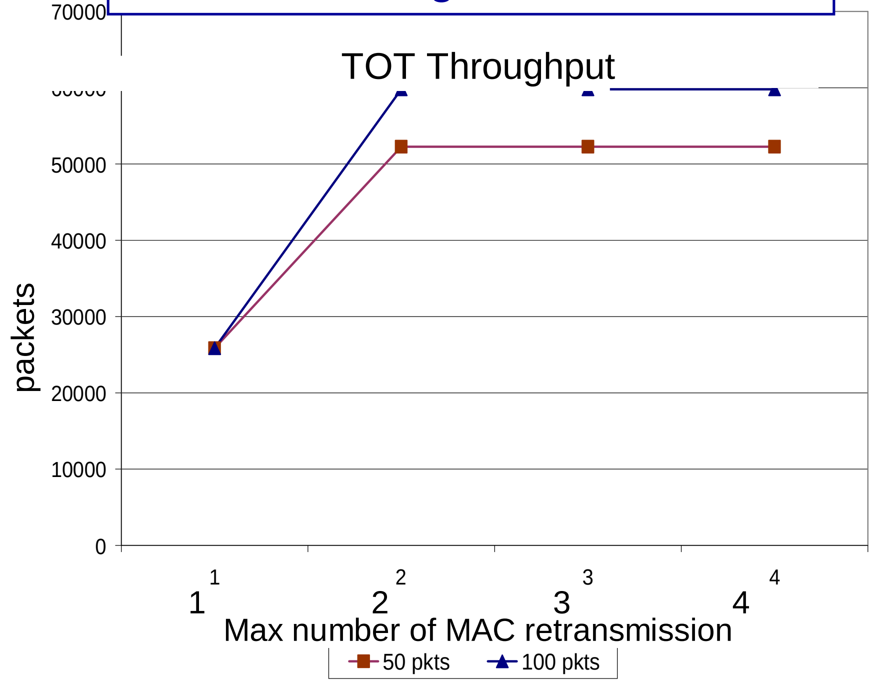
MAC Queue Size vs Throughput

(CS)

Device-AP distance: 10m
shadowing deviation: 9



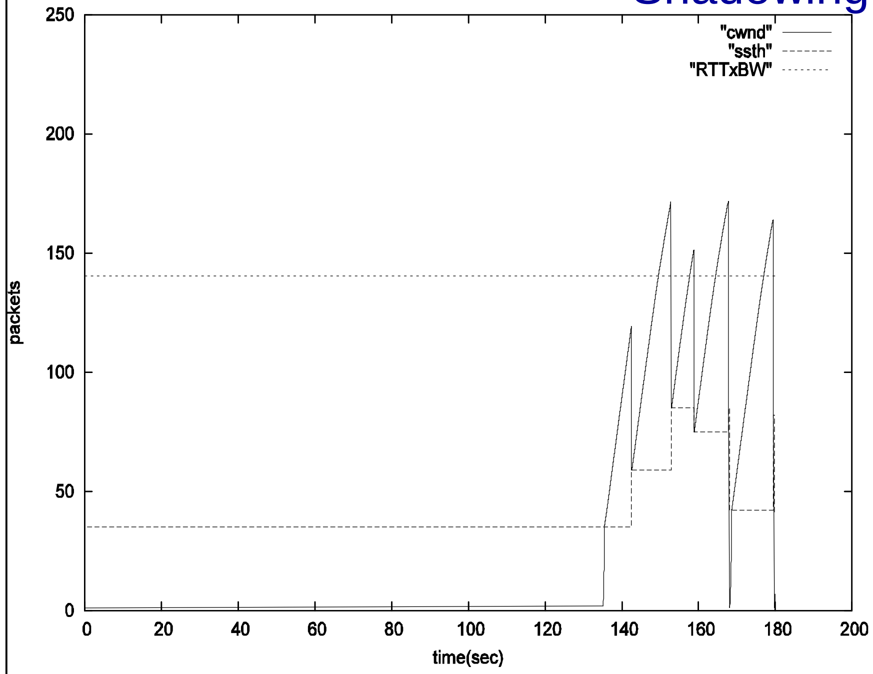
Device-AP distance: 5m
shadowing deviation: 9



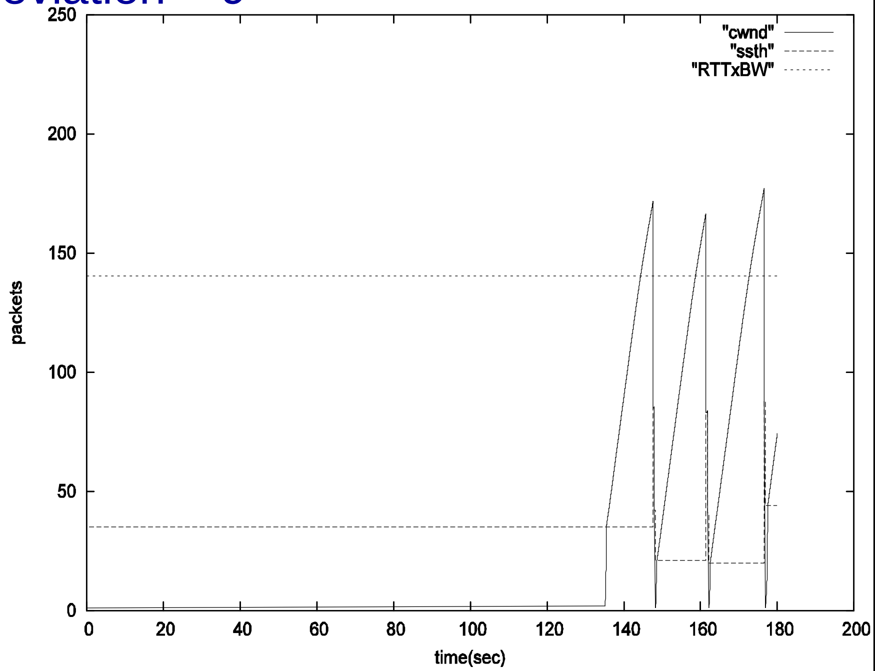
MAC queue size (pkts): 50 vs 100

Signal Attenuation vs Throughput

Shadowing Deviation = 9



MAC max retr = 2



MAC max retr = 3

Reverse game and chat
flow causes ACK losses: Timeouts!

MAC Queue Size vs Jitter (1/3)

(CS)

Jitter	50 pkts	100 pkts
maximum (ms)	33.740	108.36
average (ms)	1.306	2.041
variance	7.360	22.079
pkts received	2658	2658

GAMING FLOW JITTER: STATISTICS FOR VARIOUS MAC LAYER QUEUE SIZES; CONSIDERED PERIOD = [0-180S], MAX MAC RETR = 4, SHADOWING DEVIATION = 9

Jitter	50 pkts	100 pkts
maximum (ms)	33.740	108.36
average (ms)	3.056	5.229
variance	16.665	49.470
pkts received	899	899

GAMING FLOW JITTER: STATISTICS FOR VARIOUS MAC LAYER QUEUE SIZES; CONSIDERED PERIOD = [135-180S], MAX MAC RETR = 4, SHADOWING DEVIATION = 9

MAC Queue Size vs Jitter (2/3)

(CS)

Jitter	50 pkts	100 pkts
maximum (ms)	31.091	44.632
average (ms)	1.045	1.566
variance	4.833	11.034
pkts received	2654	2655

GAMING FLOW JITTER: STATISTICS FOR VARIOUS MAC LAYER QUEUE SIZES; CONSIDERED PERIOD = [0-180S], MAX MAC RETR = 3, SHADOWING DEVIATION = 9

Jitter	50 pkts	100 pkts
maximum (ms)	31.091	44.632
average (ms)	2.292	3.835
variance	11.502	24.431
pkts received	896	897

GAMING FLOW JITTER: STATISTICS FOR VARIOUS MAC LAYER QUEUE SIZES; CONSIDERED PERIOD = [135-180S], MAX MAC RETR = 3, SHADOWING DEVIATION = 9

MAC Queue Size vs Jitter (3/3) **(CS)**

Jitter	50 pkts	100 pkts
maximum (ms)	33.740	108.36
average (ms)	3.056	5.229
variance	16.665	49.470
pkts received	899	899

GAMING FLOW JITTER: STATISTICS FOR VARIOUS MAC LAYER QUEUE SIZES; CONSIDERED PERIOD = [135-180S], MAX MAC RETR = 4, SHADOWING DEVIATION = 9

Jitter	50 pkts	100 pkts
maximum (ms)	31.091	44.632
average (ms)	2.292	3.835
variance	11.502	24.431
pkts received	896	897

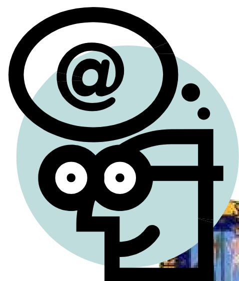
GAMING FLOW JITTER: STATISTICS FOR VARIOUS MAC LAYER QUEUE SIZES; CONSIDERED PERIOD = [135-180S], MAX MAC RETR = 3, SHADOWING DEVIATION = 9

Huge jitter difference

Summarizing

(CS)

- Long lasting FTP/TCP flows increase delays
- Need for queuing delay reduction
- Easy solution, appropriately setting MAC layer parameters:
  - Reducing MAC layer retransmissions to 3
  - Smaller MAC queue size (max 50 pkts)

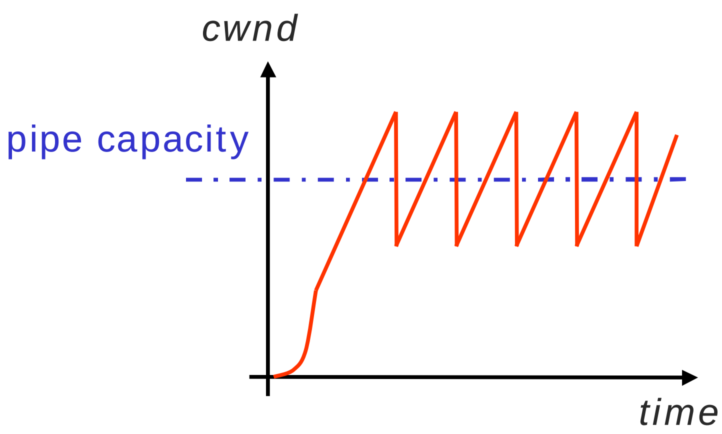


Smart Access Point with Limited Advertisement Window (SAP-LAW)

Delays caused by TCP Behavior

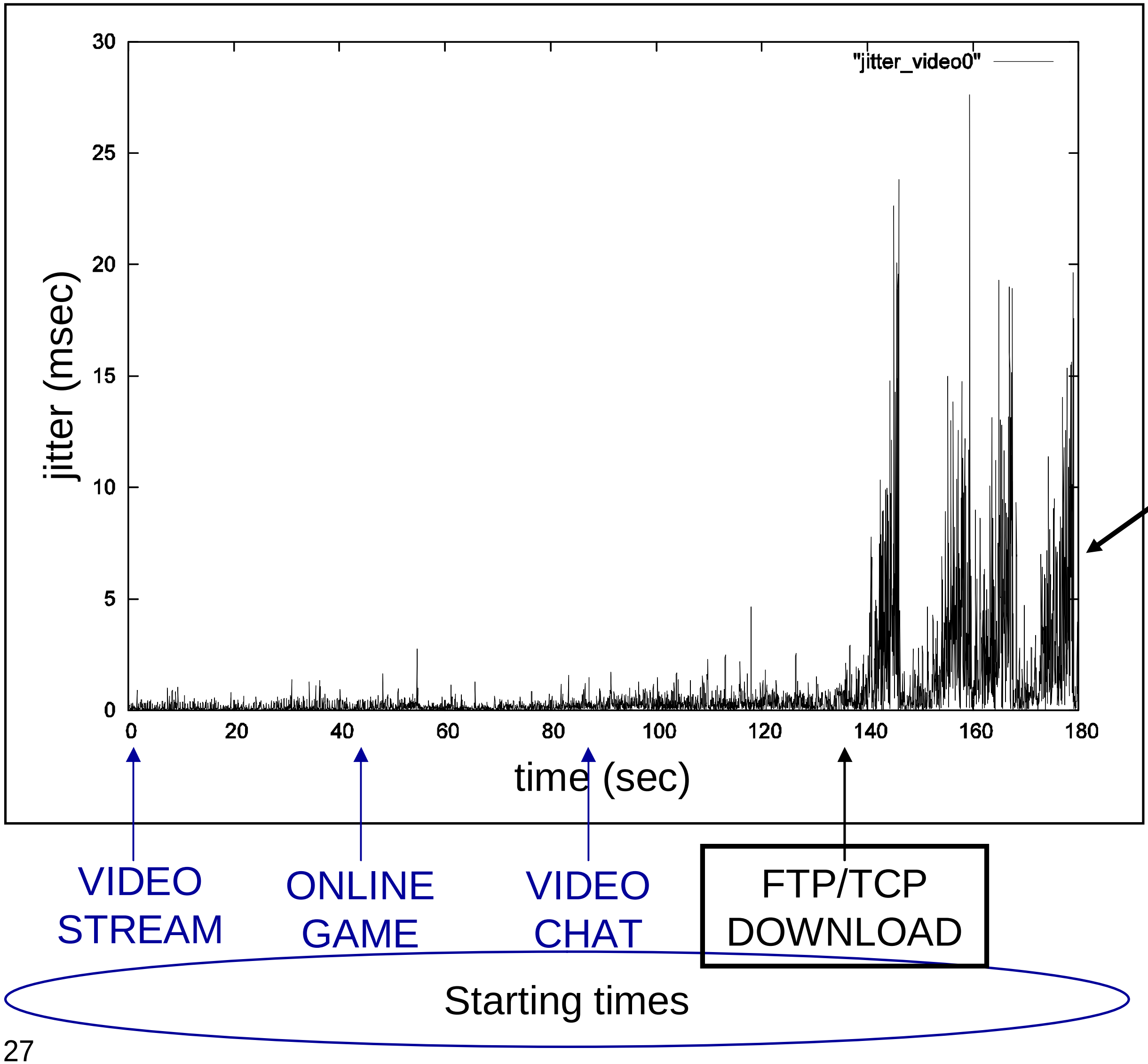
(CS)

- TCP has an aggressive behavior
- Window based flow control mechanism
- Continuously probes the link for more bandwidth
- Can fill up the AP buffer with its packets
  - This may increase delays and deteriorate performances of real-time streams



TCP Impact on Concurrent Real-Time Applications (CS)

Jitter of a video stream with other streams activated



Concurrent TCP traffic jeopardizes interactivity for online applications

Reversing the Problem (CS)

TCP vs **UDP**

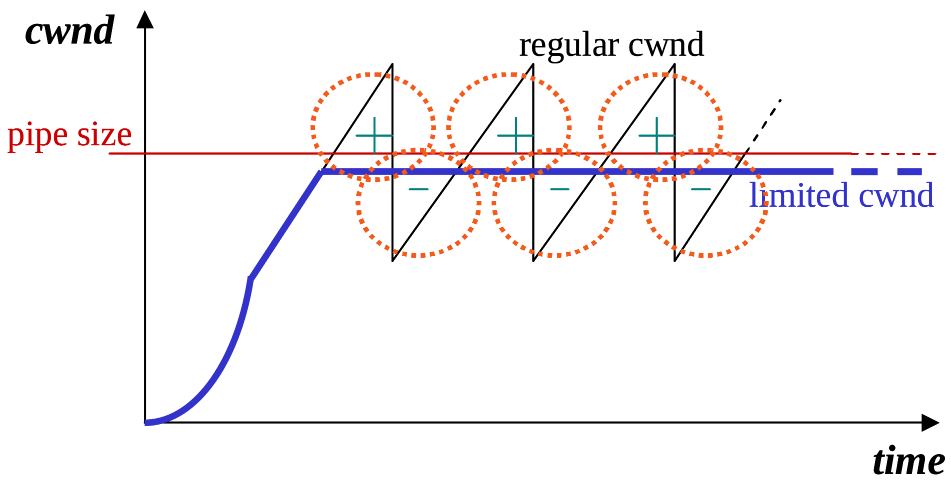
- Typically, UDP traffic is seen as a problem for TCP flows
- Here, TCP flows can jeopardize real-time requirements of UDP-based applications
- No improvements on TCP but act on TCP to not upset UDP performances
- Best tradeoff to provide:
  - Low per-packet delays for real time applications
  - High goodputs for downloading applications

“Smart” AP: SAP-LAW

(CS)

- IDEA: exploit the *advertised window* to limit the bandwidth utilized by TCP flows

Regular cwnd: regular TCP, typical saw tooth shape
Limited cwnd: window regulated by exploiting the advertised window

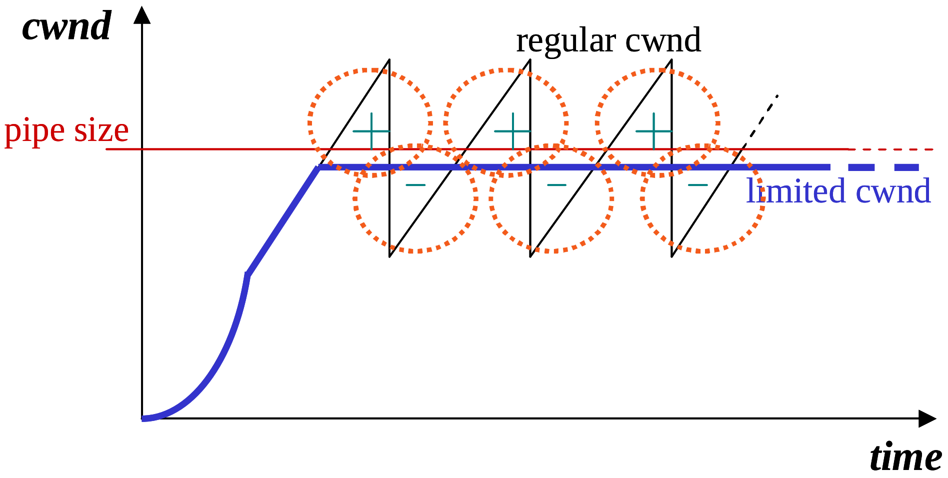


Regular window provides a sending rate that oscillates (+ and – in the picture) around the one ensured by the *limited* window

A balance can be reached on the final throughput

"Smart" AP: SAP-LAW *(CS)*

- IDEA: exploit the *advertised window* to limit the bandwidth utilized by TCP flows
- Avoid buffer utilization at the AP
  - Exploits information available at the AP to determine
    - ✓ Amount of bandwidth occupied by UDP-based traffic
    - ✓ Number of active TCP flows
  - On-the-fly modification of the *advertised window* of TCP flows ***at the AP***



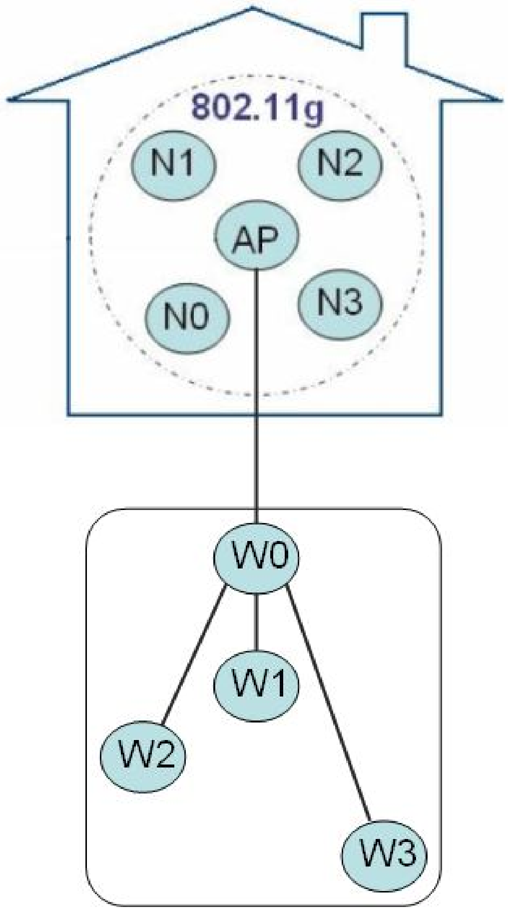
$$maxTCPrate(t) = \frac{(C - UDPtraffic(t))}{\#TCPflows(t)}$$

No queues:

- lower delays (for UDP flows)
- smoother throughput (no losses and reductions of the sending window for TCP flows)

Simulation Environment: NS-2

(CS)



Node 1	Node 2	Delay	Capacity	Queue Size
W1	W0	10ms	100Mbps	140pkts
W2	W0	20ms	100Mbps	140pkts
W3	W0	30ms	100Mbps	140pkts
W0	AP	10ms	100Mbps	140pkts

SIMULATION CONFIGURATION (WIRED LINKS)

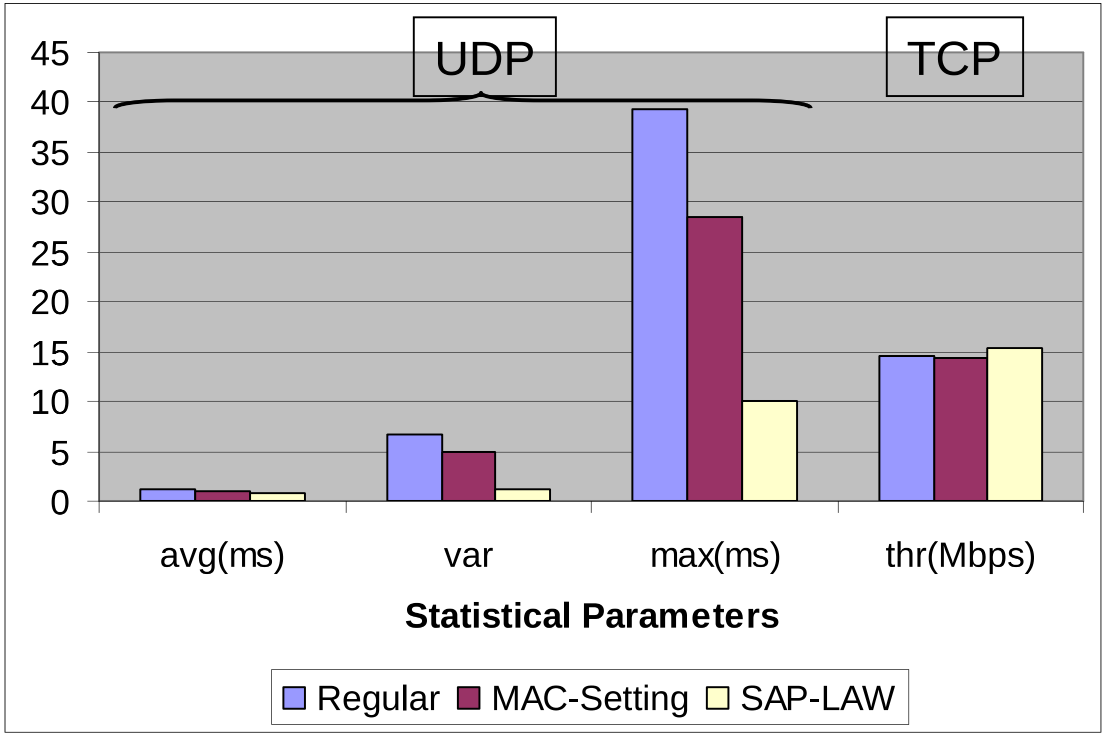
From	To	Type	Transport Protocol	Start	End
AP	N0	Movie Stream	UDP	0s	180s
W1	N1	Game Traffic	UDP	45s	180s
N1	W1	Game Traffic	UDP	46s	180s
W2	N2	Video Chat	UDP	90s	180s
N2	W2	Video Chat	UDP	91s	180s
W3	N3	FTP	TCP	135s	180s

SIMULATED APPLICATION LAYER TRAFFIC FLOWS

Summarizing Results

(CS)

- Regular: classic TCP
- MAC-Setting: classic TCP, reduction of the buffer size at the AP
- SAP-LAW: Our approach



- SAP-LAW has good performances

Real Testbed Assessment

(CS)

- We have now created a prototype (SLUS: SAP-LAW in User Space), a user space OpenWRT solution on a NETGEAR WGT634U AP and performed real testbed experiments

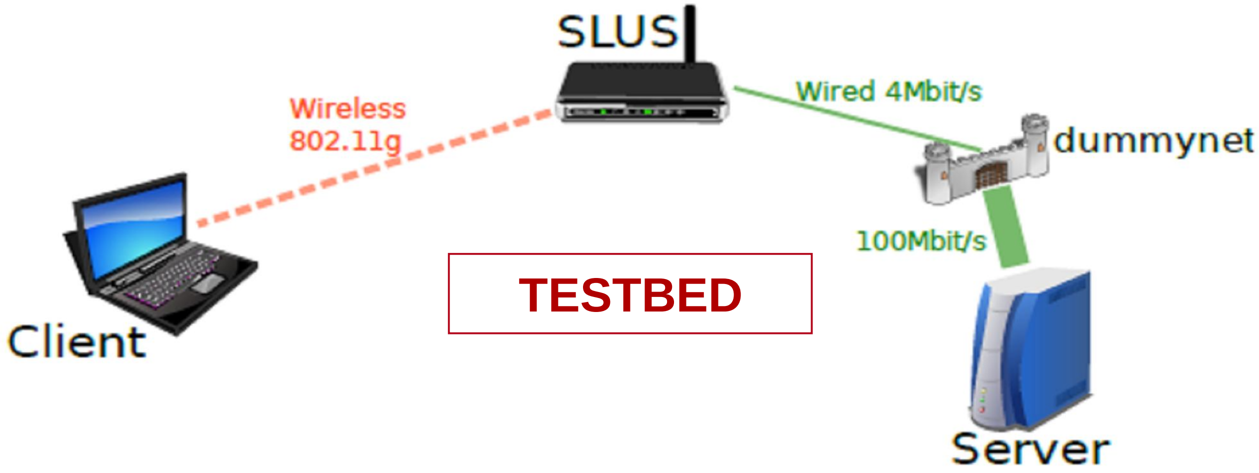


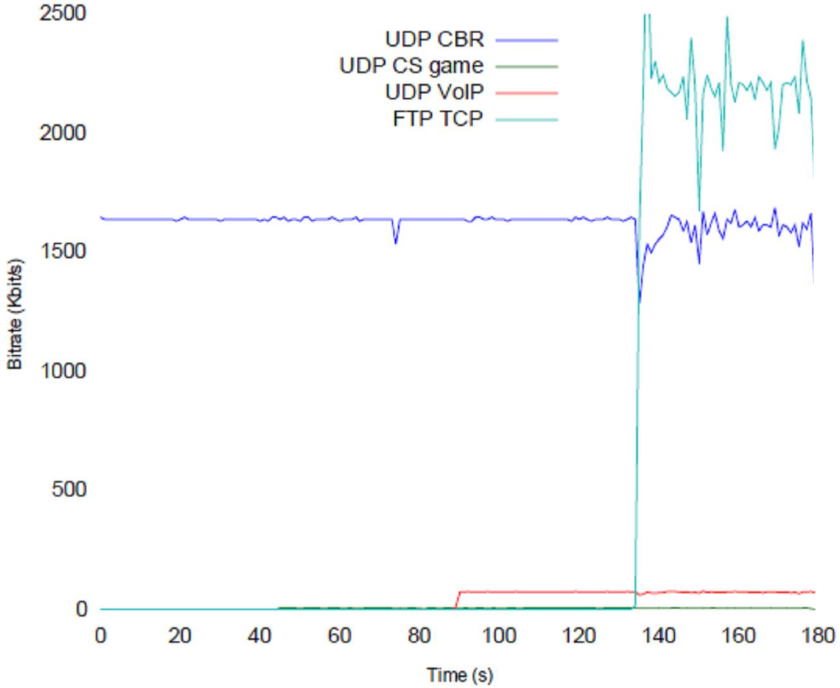
Table 1. Traffic flows generated by ITGSend (D-ITG suite).

Flow type	Protocols	Start time	End time	RTT
CBR flow	UDP	0s	180s	60ms
Online Game	UDP	45s	180s	60ms
VoIP	UDP	90s	180s	60ms
FTP flow	TCP	135s	180s	60ms

Throughput Evaluation

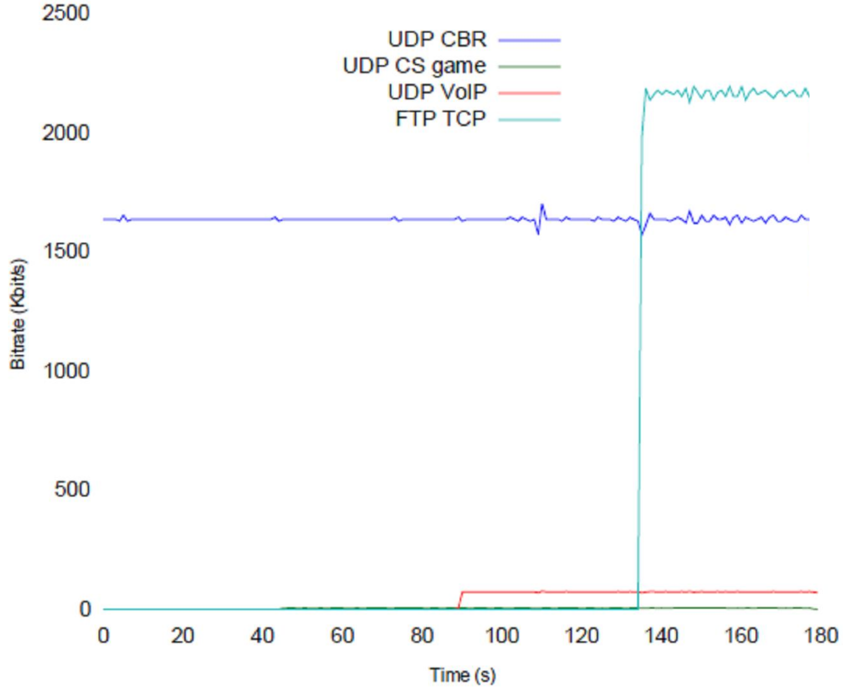
(CS)

Regular



High oscillations

SLUS

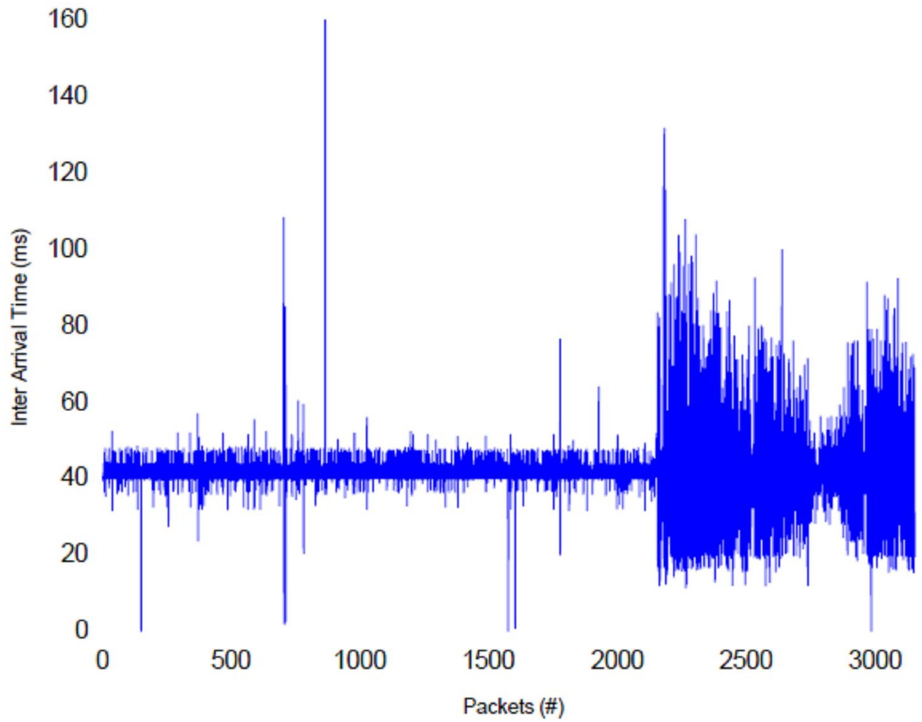


Smooth progression
(same final throughput)

Interarrival Time Evaluation

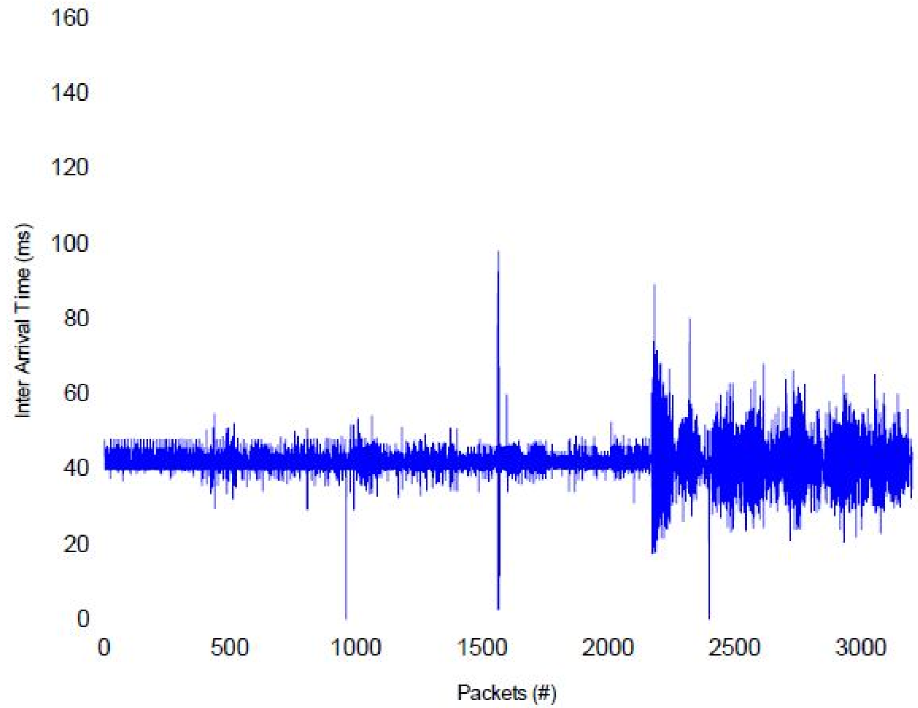
(CS)

Regular



packet queuing: high oscillations

SLUS



limited queuing/oscillations
(better jitter)

Conclusions

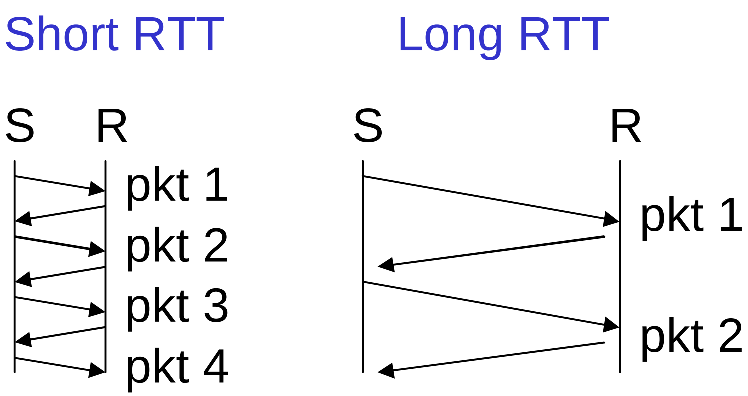
(CS)

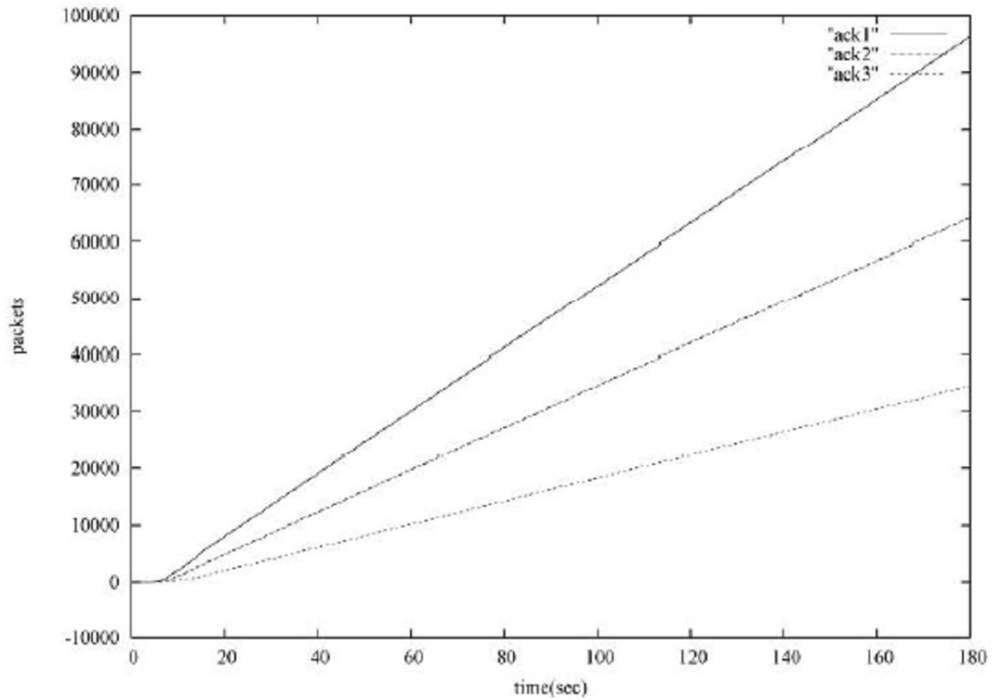
- In in-home wireless scenarios, concurrent long lasting TCP flows increase delays
  - Especially on wireless links
- Real-time online applications need a reduction of the queuing delay
- SAP-LAW: on-the-fly modification of the *advertised window* of TCP flows ***at the AP*** to
  - Augment UDP flows performances
  - Maintain a high goodput for TCP downloads
- SAP-LAW is easily deployable as it requires modifications *only* at the AP

Enhancing SAP-LAW for RTT-fairness

TCP's RTT-unfairness

- The throughput of a connection is inversely proportional to the RTT length
  - Short RTT flows capture the channel
  - Long RTT flows starve

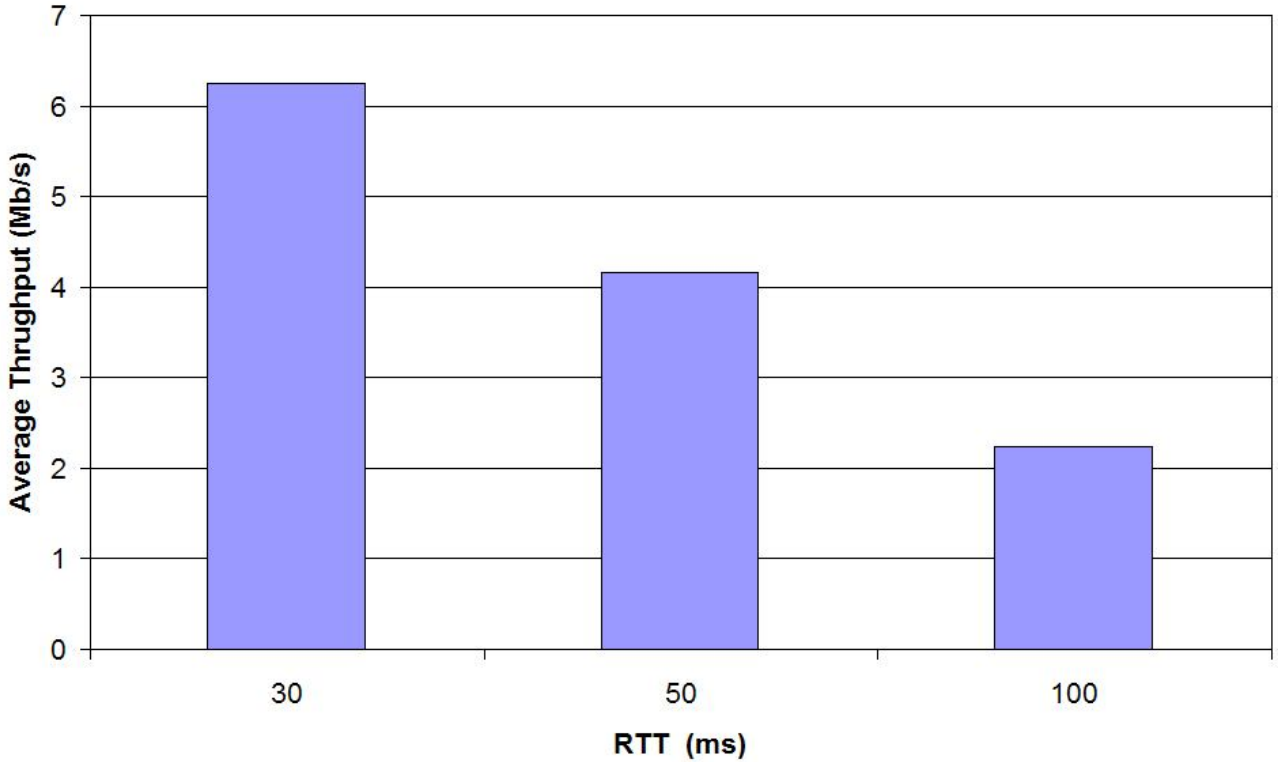




TCP's RTT-unfairness

(CS)

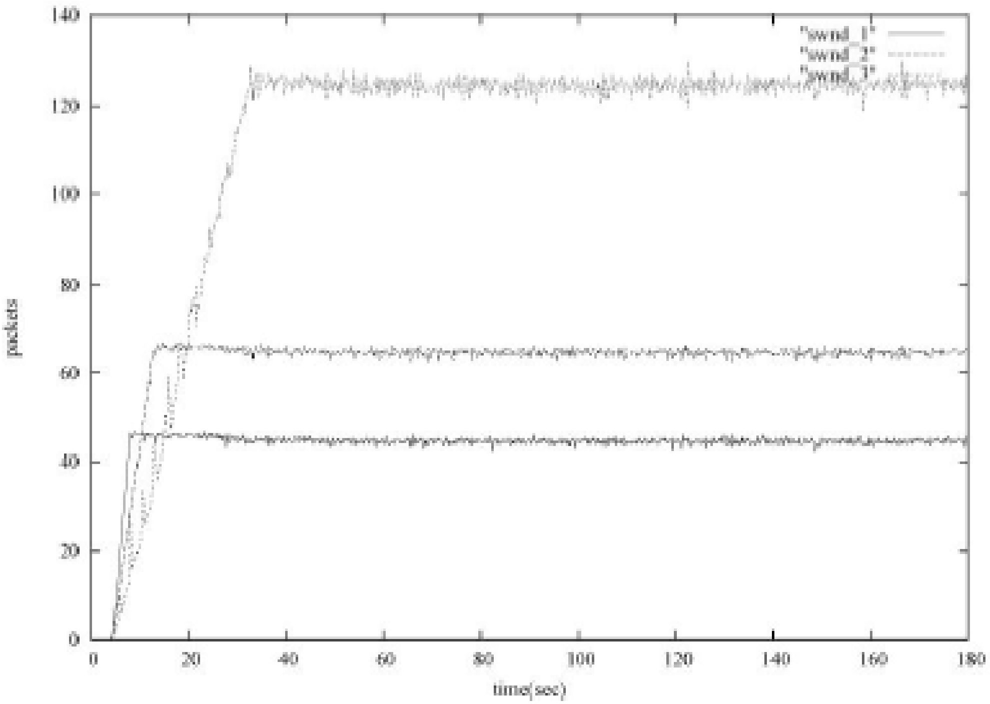
- Even SAP-LAW does not solve it:
  - Same windows results in different throughput (if different RTTs)



Advertised Window Computation *(CS)*

$$maxTCPrate_i(t) = \frac{(C - UDPtraffic(t)) * RTT_i}{\sum \frac{RTT_i}{avg\_RTT_{min}}}$$

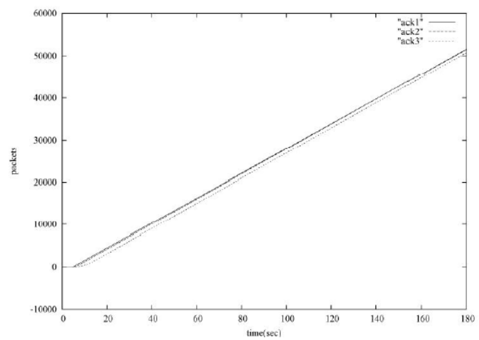
Sending window



Three TCP flows with

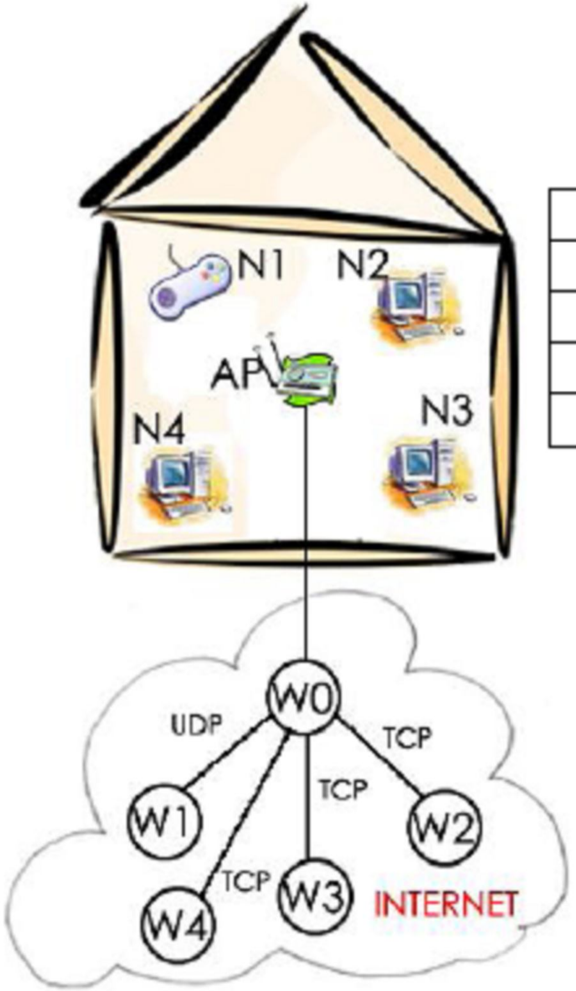
30, 50, 100 ms of RTT

Throughput



Different sending windows
produces same throughput

Simulation Assessment (CS)



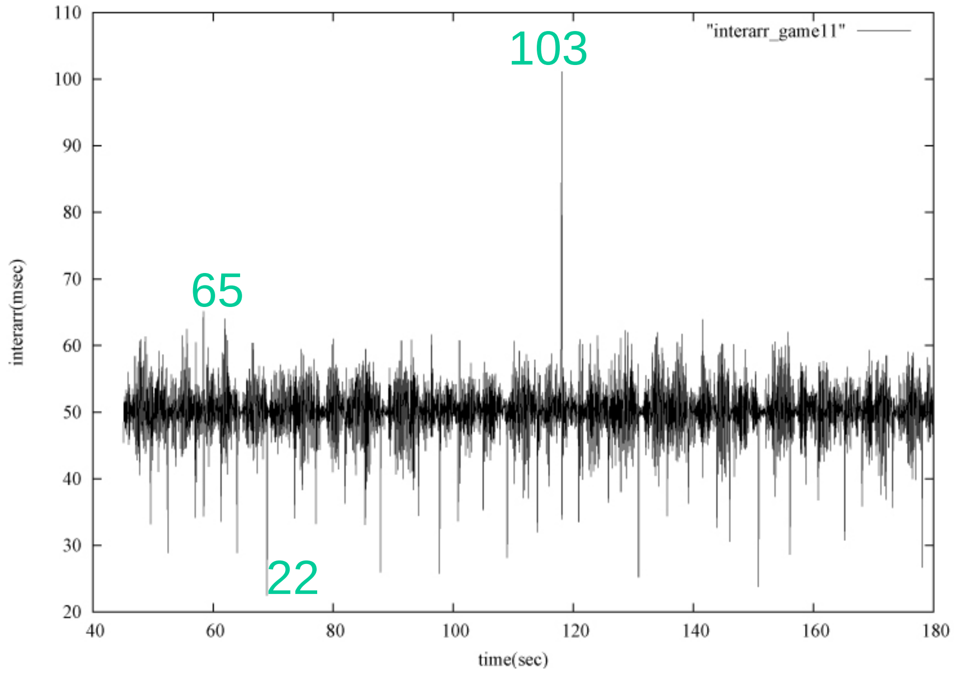
Flow Type	Protocol	From	To	Start	End	RTT
FTP 1	TCP	N2	W2	1 s	180 s	30 ms
FTP 2	TCP	N3	W3	1 s	180 s	50 ms
FTP 3	TCP	N4	W4	1 s	180 s	100 ms
Online Game	UDP	N1	W1	45 s	180 s	20 ms

- NS-2 simulator
- Wireless bottleneck (802.11g)
- Various configurations

Main contribution of this work is the comparison of our SAP-LAW with other protocols/solutions.

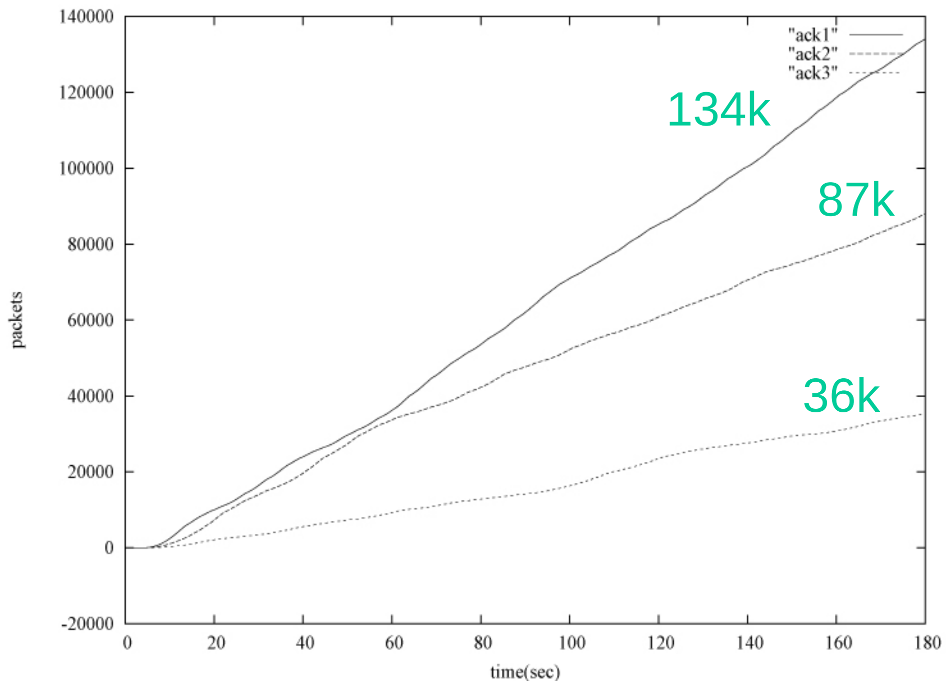
Comparison Among Transport Protocols: TCP SACK

Interarr time of game events



High oscillations due to queuing

ACK transmission of three TCP flows

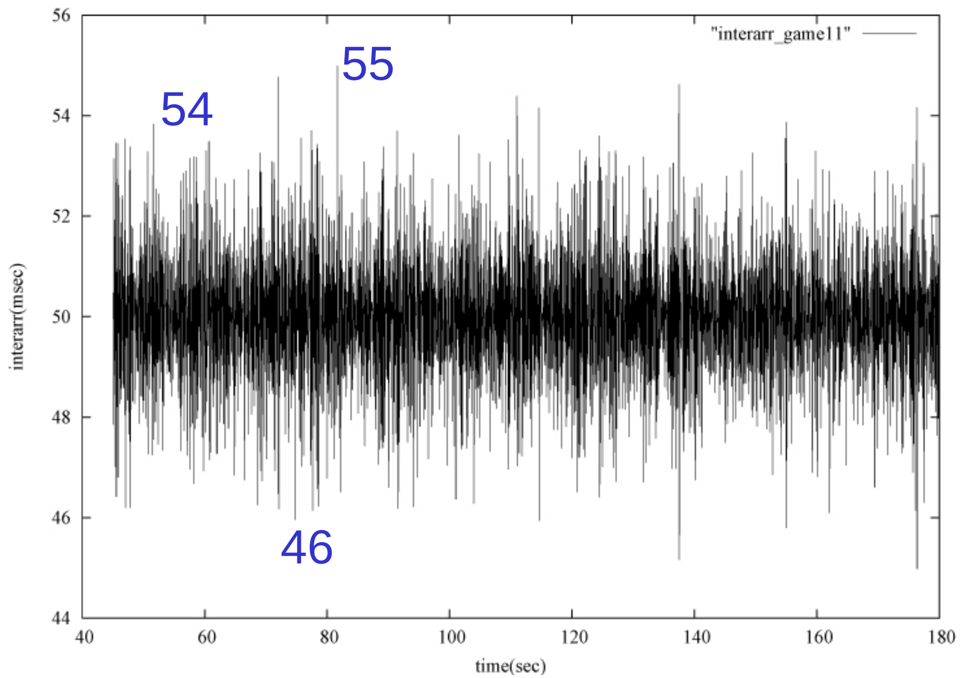


Unfair bandwidth usage

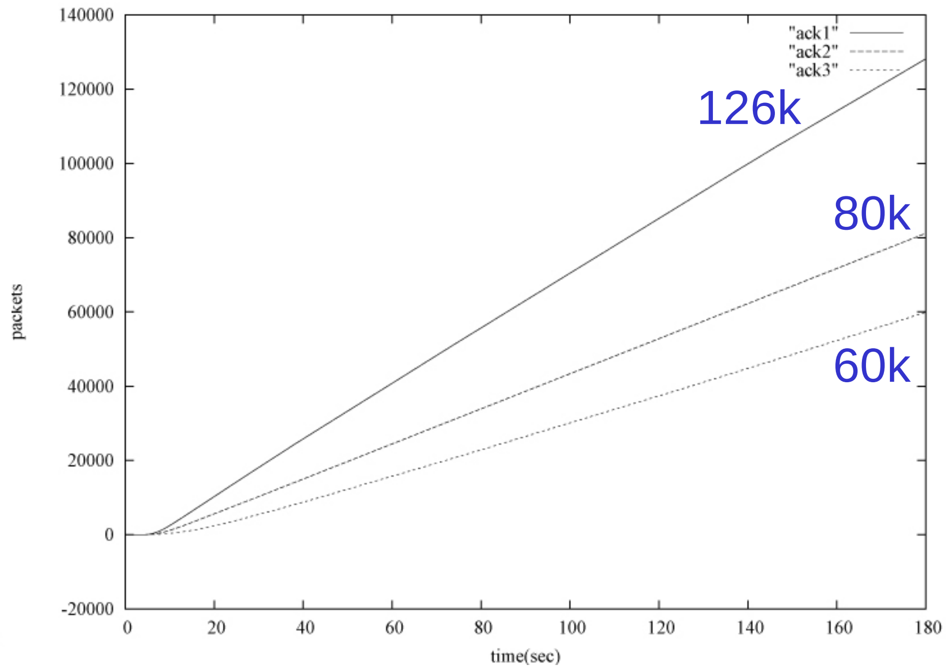
(CS)

Comparison Among Transport Protocols: TCP Vegas

Interarr time of game events



ACK transmission of three TCP flows



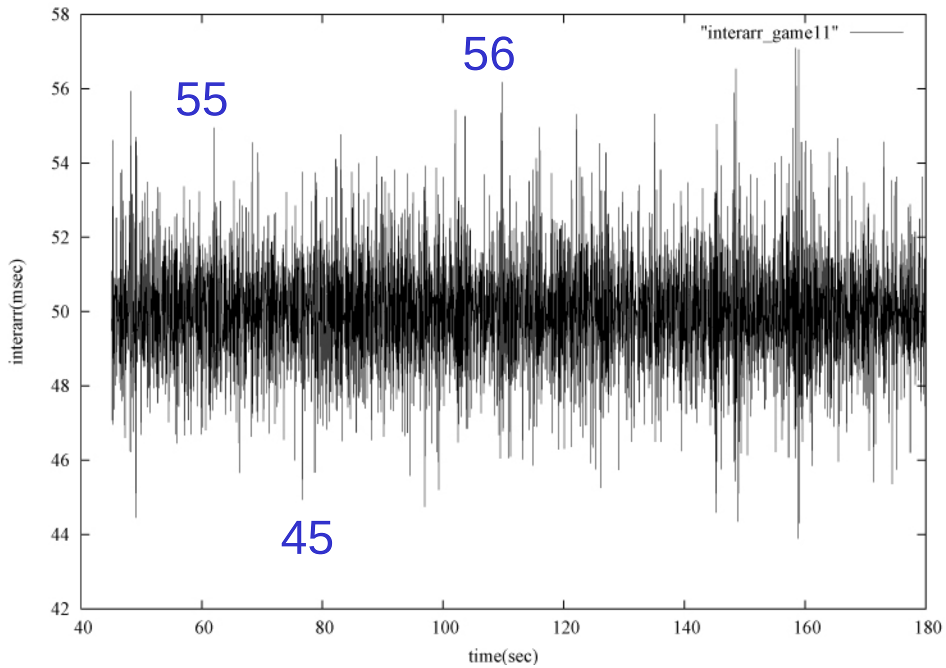
No queuing: low oscillations

Still unfair bandwidth usage

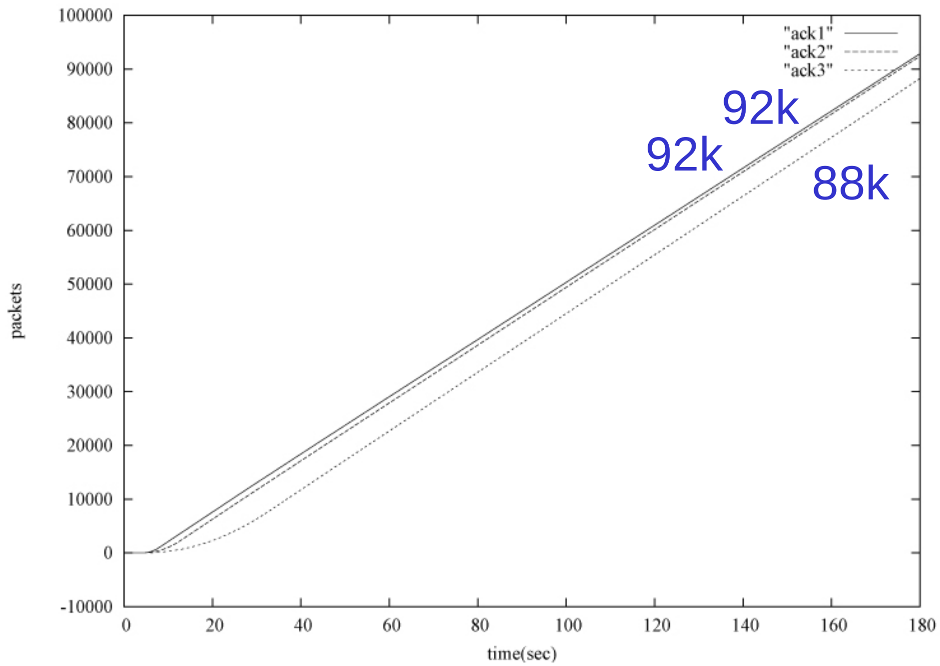
Plus, TCP Vegas cannot coexist with legacy TCP

Comparison Among Transport Protocols: SAP-LAW

<u>Interarr time of game events</u>



<u>ACK transmission of three TCP flows</u>

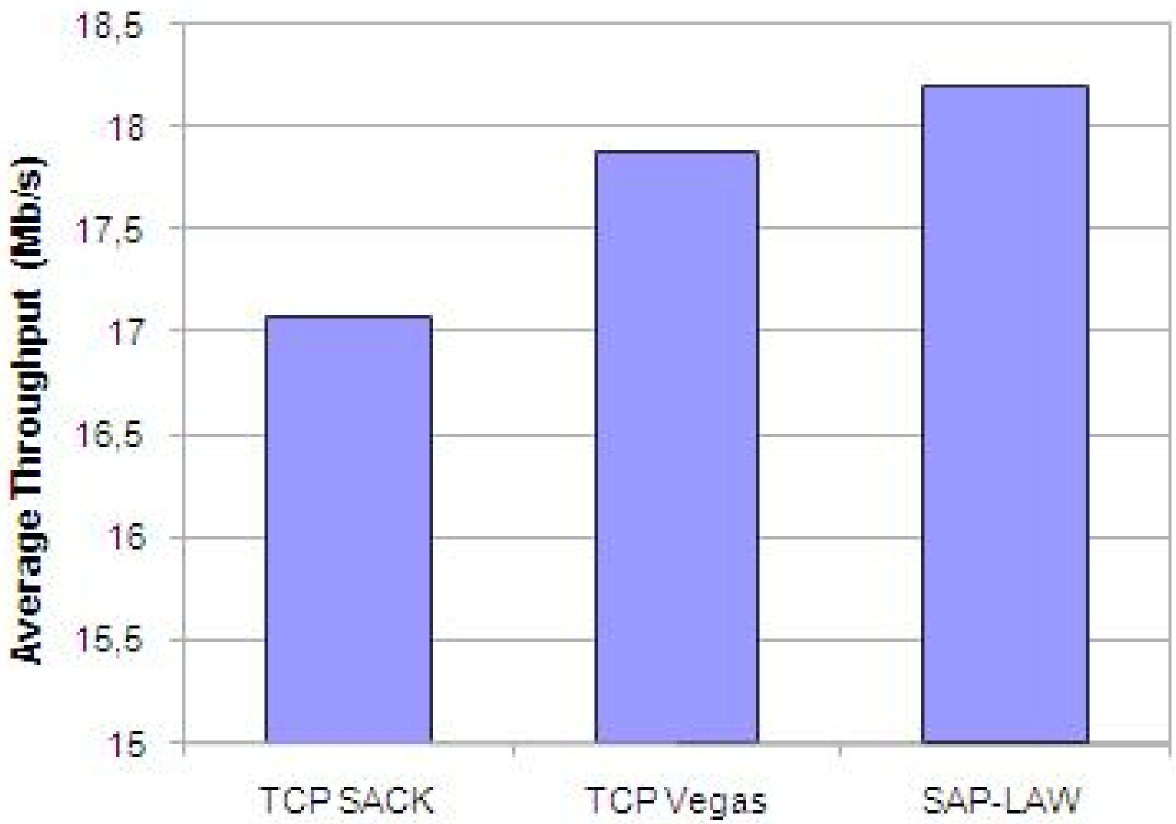


<u>No queuing: low oscillations!</u>

<u>Fair bandwidth usage!</u>

<u>Plus, regular TCP used under SAP-LAW to preserve compatibility versus legacy TCP</u>

Comparison Among Transport Protocols: Efficiency

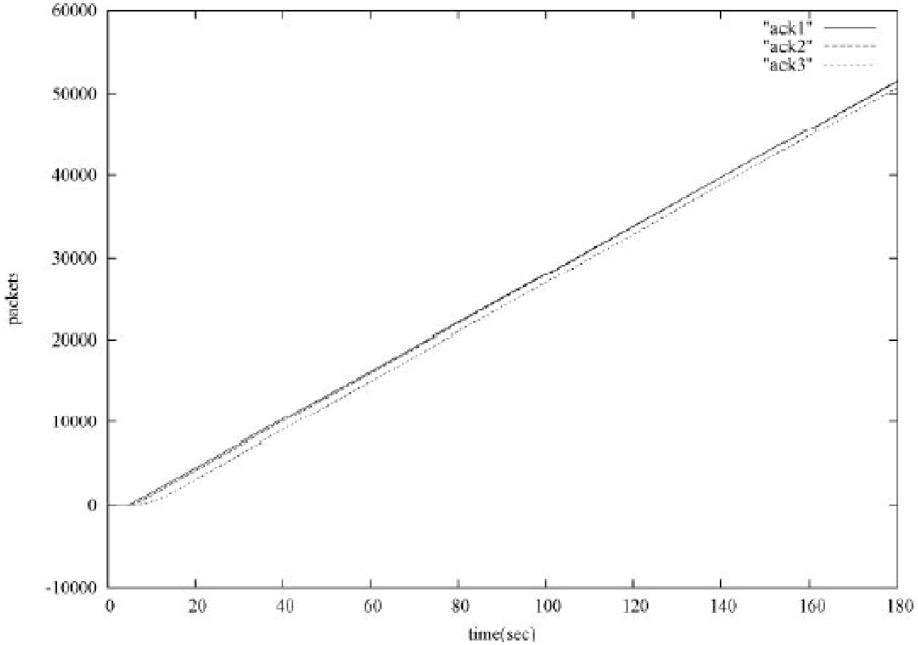


No losses also means higher efficiency in terms of throughput

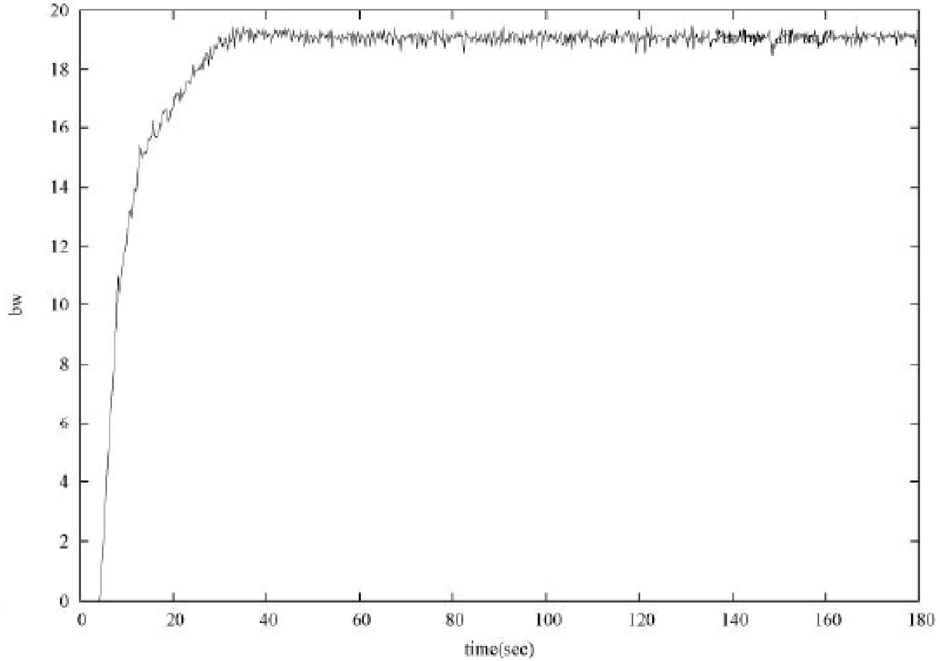
(CS)

Results: High and Fair TCP's Throughputs

Single throughputs



Aggregate throughput



The three TCP flows have the same throughput regardless of different RTTs

The aggregate throughput consumes all the available bandwidth

Conclusions

(CS)

- Two problems:
  1. Long lasting TCP flows increase delays
     - ✓ Real-time applications need a reduction of the queuing delay
  2. TCP's throughput depends on RTT

- <u>SAP-LAW solution:</u> on-the-fly modification of the *advertised window* of TCP flows *at the AP*

- SAP-LAW is easily deployable as it requires modifications *only* at the AP
  - Utilized on top of legacy transport protocols

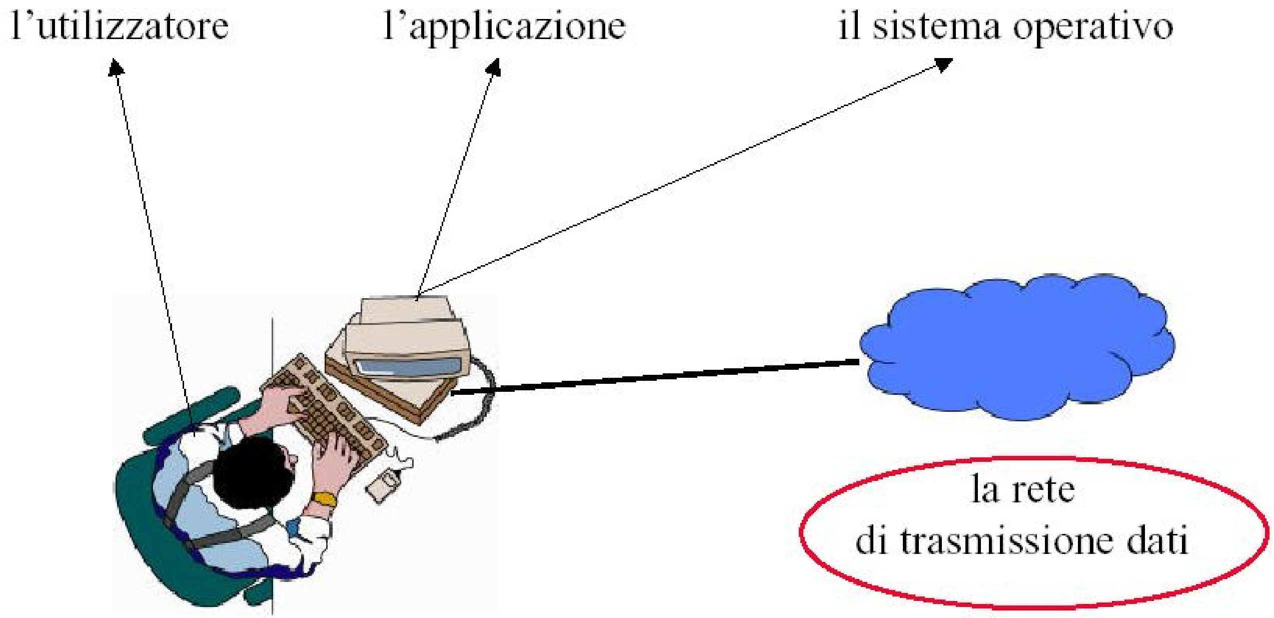
FINE Caso di Studio (CS)

Specifica di QoS in MM

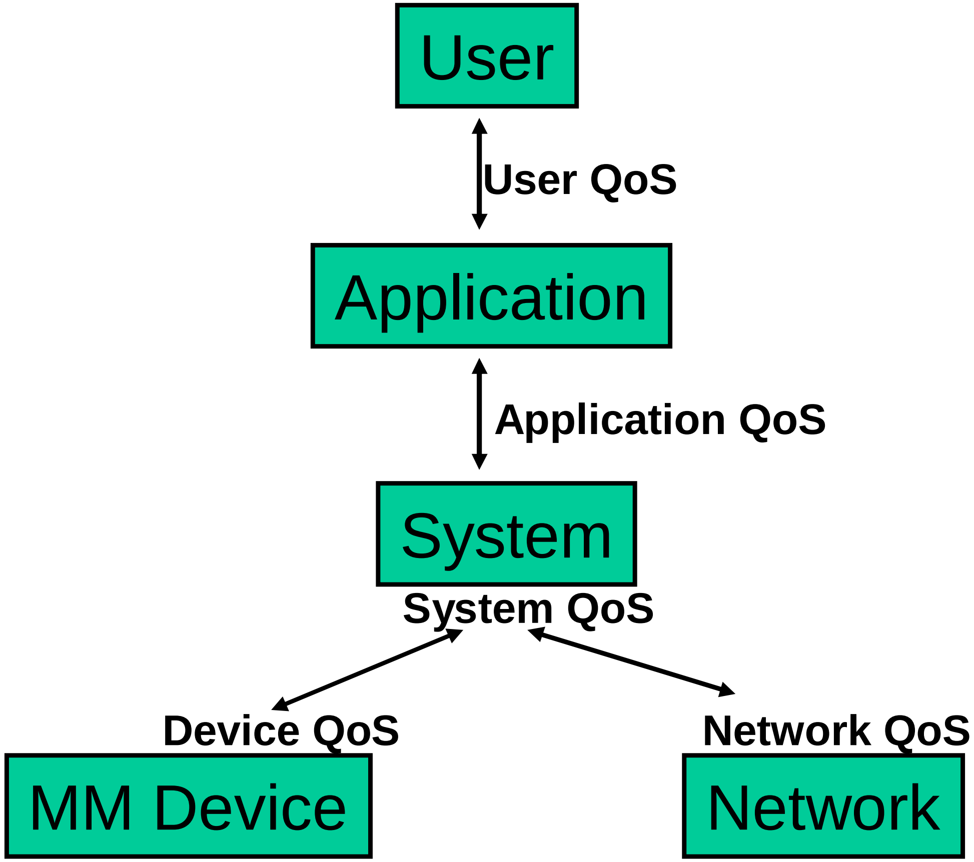
- Specifica di requisiti di QoS per applicazioni MM distribuite può avvenire attraverso:
  1. Utilizzo di bound o intervalli
  2. Uso di singoli parametri
  3. By example
- Come si processano i requisiti di Qos per applicazioni MM distribuite:
  1. Assessment dei requisiti
  2. Mapping del risultato dell'assessment sui livelli inferiori dell'architettura
  3. **Negoziazione** tra le componenti del sistema preposte alla garanzia del soddisfacimento dei requisiti di QoS

Componenti per la gestione della QoS

La QoS può essere gestita a vari livelli:



Quale QoS?


User
User QoS
Application
Application QoS
System
System QoS
Device QoS
Network QoS
MM Device
Network

QoS: Rete e OS

- Un ruolo fondamentale per la garanzia del soddisfacimento dei requisiti di QoS è giocata a livello di rete
- La rete di trasmissione dati può offrire qualità di servizio in maniera tale da soddisfare le richieste provenienti dall'utilizzatore, se è in grado di "trattare e gestire" il traffico prodotto dalle varie applicazioni
- In questo caso, differenti richieste di qualità di servizio determinano l'offerta di qualità diverse a differenti flussi di dati
- Tuttavia, la qualità di servizio deve essere "gestita" lungo l'intero percorso del flusso dati (end to end QoS)
- Non irrilevante è il ruolo del SO

QoS: osservazioni

- L'utente deve essere informato sui possibili dinieghi alle sue richieste di QoS (*exception handling requirement*)
- Il cliente può essere o umano o un'altra applicazione (*standard interfaces requirement*)
- La rinegoziazione della QoS può determinare cambiamenti nel carico di sistema complessivo (*QoS constant monitoring requirement*)

QoS: responsabilità dell'OS

- Le applicazioni MM distribuite richiedono garanzie temporali per ogni componente hw/sw per gestire flussi di dati MM continui
- Il soddisfacimento dei requisiti di QoS dipendono così dalla capacità dell'OS di gestire il RT (Real Time)
- I parametri di QoS coinvolti ai vari livelli di astrazione dell'OS possono essere di due tipi:
  1. High level: le interfacce dell'OS (e.g., chiamate di sistema) devono offrire servizi QoS oriented in termini, per esempio, di gestione del throughput e del ritardo
  2. Low level: il sistema operativo deve includere meccanismi per la gestione in RT delle prestazioni, ad esempio tecniche di scheduling RT per CPU e bus

Sostanzialmente, si tratta della gestione delle risorse in un contesto guidato dal tempo

QoS: responsabilità dell'OS

- Gestione delle risorse:
  - bisogna *mappare* i requisiti di QoS delle applicazioni MM distribuite sulle capacità delle risorse richieste (cioè, essere in grado di eseguire in un dato intervallo di tempo con quelle risorse)
  - bisogna *allocare/riservare* risorse
- Mapping dei requisiti di QoS sulle capacità di risorse richieste: si tratta di gestire meccanismi di rinegoziazione tra cliente e resource managers
- Allocare/riservare risorse:
  1. Schedulability test (testare la capacità delle risorse a soddisfare il QoS richiesto/assegnato dall'/all'utente)
  2. QoS calculation (valutare le migliori performance possibili che le risorse a disposizione possono garantire)
  3. Resource reservation (allocare risorse necessarie per quel livello di QoS)
  4. Resource scheduling (ordinare le richieste secondo garanzie QoS)

Gestione conflitti in OS

- Riservare/allocare risorse:
  - bisogna *gestire* i conflitti tra le richieste delle differenti applicazioni

- Approccio Pessimistico:
  1. assunzione: maximum work-load (MWL), i.e. i conflitti non sono tollerati
  2. evita sovraccarico e conflitti per risorse
  3. può causare bassa utilizzazione di risorse, se MWL non capita
- Approccio ottimistico:
  1. assunzione: WL medio , i.e. conflitti tollerati
  2. miglior utilizzazione risorse, potenzialmente
  3. Rischio di overbooking, con conseguente decadimento/degrado dei services
  4. <u>Richiede monitor che scopra e risolva i conflitti, basata sull'importanza delle applicazioni</u>

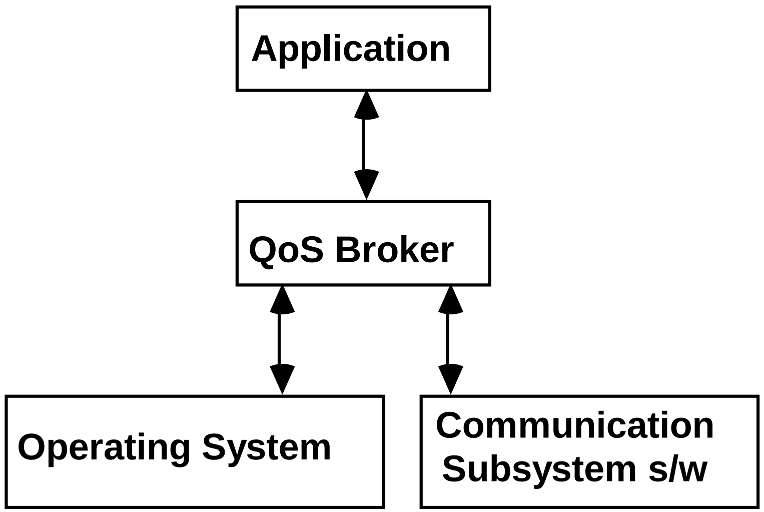
QoS Broker

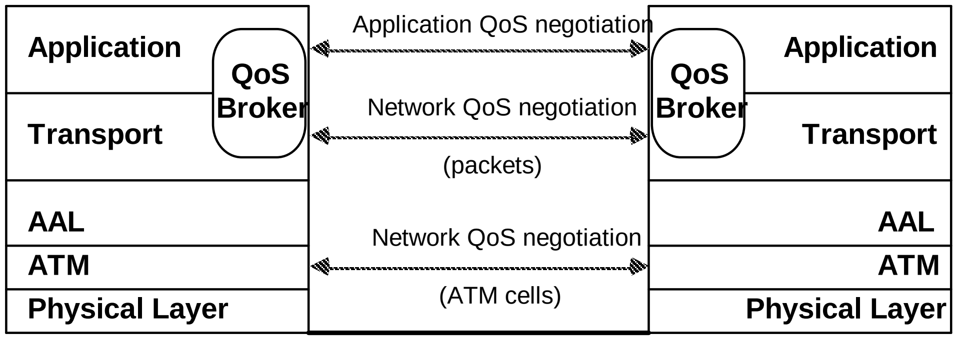
Concetto di QoS Broker (K. Nahrstedt, J. M. Smith, 1995):

• Coopera con OS ed il communication subsystem per ottenere un bilanciamento tra requisiti di QoS di applicazioni MM distribuite, risorse di OS e di rete

• Negozia con altri brokers remoti sull'uso delle risorse

• Può essere implementato come estensione sw al sottosistema di comunicazione





QoS: ruolo della rete

- Per lo standard ISO, la QoS viene considerata nell'ambito di connessioni a livello di "trasporto"
- I parametri che definiscono la QoS si riferiscono quindi essenzialmente a parametri di rete
- Esiste dunque una fase di negoziazione per definire i valori dei parametri che specificano la QoS a livello di rete

Parametri di QoS su rete

- Ritardo della creazione di connessione
- Probabilità di fallimento nella creazione di una connessione
- Capacità (Throughput): numero di byte trasferiti in un secondo
- Ritardo di trasmissione (Transit Delay): tempo trasmissione-ricezione
- Tasso di errori
- Protezione: sicurezza contro terze parti non autorizzate
- Priorità: possibilità di indicare che alcune connessioni sono più importanti di altre
- Resilienza: probabilità che lo stesso strato di trasporto interrompa spontaneamente una connessione

Tipi di Servizio

- Fondamentalmente possono essere individuate tre tipologie di servizi in base a differenti QoS:
  - Guaranteed Service: vengono fornite garanzie sui parametri di QoS
  - Predictive Service: sono fatte stime dei parametri di QoS
  - Best-effort Service: il comportamento del servizio non è predicibile (non possono essere fatte nemmeno stime)

Guaranteed Service

- Forniscono garanzie di QoS in base ai valori di certi specifici parametri considerati,
  - Deterministicamente:
    - ✓ singolo valore: valore medio, soglie, …
    - ✓ coppia di valori: (min, avg), (lowest, target),…
    - ✓ intervallo di valori: [lower_bound, upper_bound]
  - Statisticamente:
    - ✓ valore statistico sulla rate
      - – che verrà certamente assicurato

Predicted Service

- In alcuni casi non si è in grado di fornire garanzie ma si può comunque offrire una stima dei valori dei parametri di QoS che si offriranno
- I predictive service considerano il comportamento passato (<u>euristica</u>) per stimare i parametri di QoS, ovvero i parametri possono essere calcolati sulla base dei valori rilevati precedentemente
- Sulla base di stime passate, posso fornire un certo tipo di servizio
  - Probabile ma non certo

Best-effort Service

- Non sono fornite garanzie (oppure solo garanzie parziali)
- Non vengono (specificati)/garantiti i parametri di QoS, ma c'è adattamento delle applicazioni MM alle condizioni offerte dalla rete
- Sono in realtà utilizzati nella maggior parte delle situazioni sull'attuale IP che è un protocollo di rete di tipo best-effort

Risorse

- Possono essere classificate in base a tipo
  - attiva: fornisce un servizio
    - ✓ CPU, network adapter
  - passiva: ha certe capacità richieste da risorse attive
    - ✓ RAM, bandwidth
- In base all'uso
  - esclusivo
    - ✓ speaker
  - condiviso
    - ✓ Bandwidth
- In base al numero
  - singola
  - multipla
    - ✓ multiprocessore

Risorse e Multimedia

- Le applicazioni multimediali distribuite hanno ovviamente necessità di utilizzare risorse, in modo intensivo, a causa dell'esistenza di flussi MM continui
- Spesso, tali risorse possono essere condivise:
  - banda
  - CPU
  - buffer space

Risorse e Multimedia

- Necessità di controllare l'uso delle risorse per avere una certa QoS
- Azioni possibili:
  - riservare / allocare risorse
  - fornire risorse in accordo a certi parametri di QoS
  - adattare l'uso delle risorse durante il processing dei dati multimediali
- Possono quindi entrare in gioco:
  - Protocolli di Negoziazione
  - Protocolli per la Gestione delle Risorse
  - Resource Managers

Negoziazione

- Utente necessita di certe caratteristiche di QoS per una data applicazione MM
- Non sempre, tuttavia, tali richieste possono essere soddisfatte
- Fasi possibili:
  - Negoziazione dei parametri
  - Traduzione della richiesta in un insieme di specifiche relative a differenti componenti del sistema MM

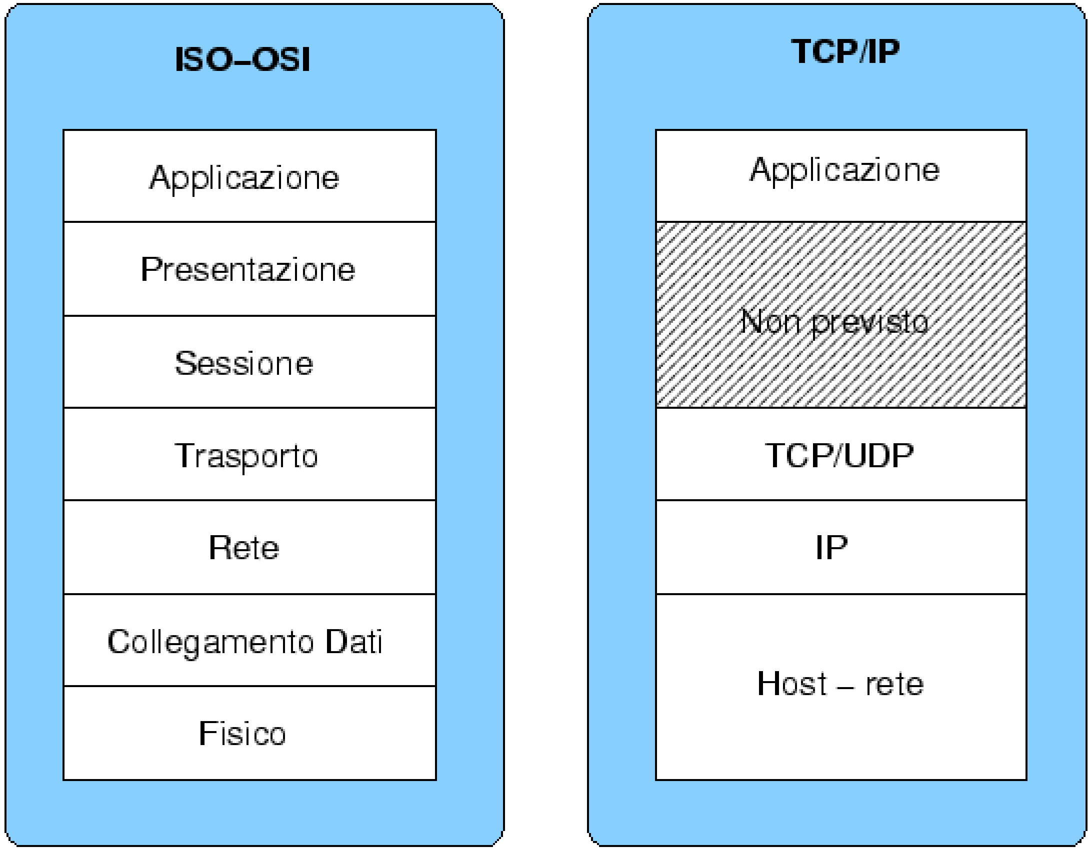
Negoziazione

- Negoziazione:
  - Quali sono le entità che negoziano?
  - Come lo fanno?

- Sono possibili diverse situazioni:
  - peer-to-peer negotiation
  - layer-to-layer communication

Layer-to-Layer Communication

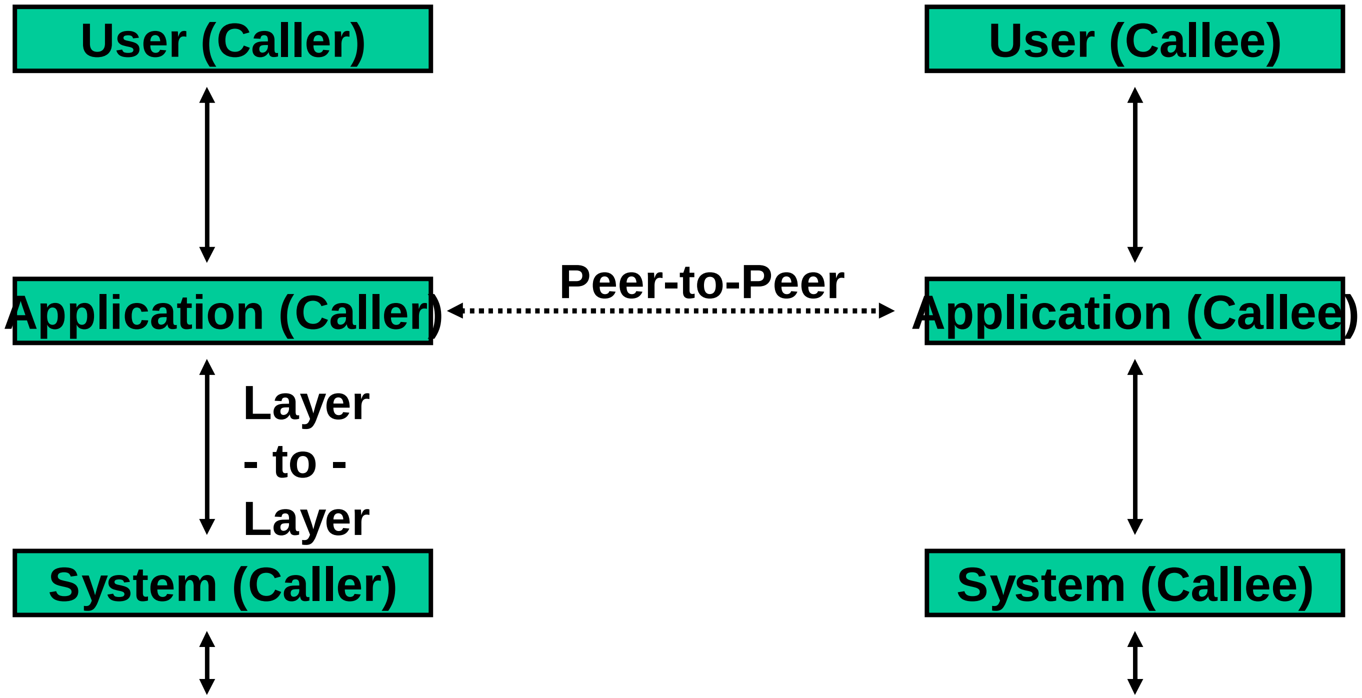
- Detta anche (in ISO) service-user-to-service-provider negotiation
- Comunicazione tra strati di protocollo di diverso livello
- Ogni livello comunica col livello immediatamente sottostante e sovrastante



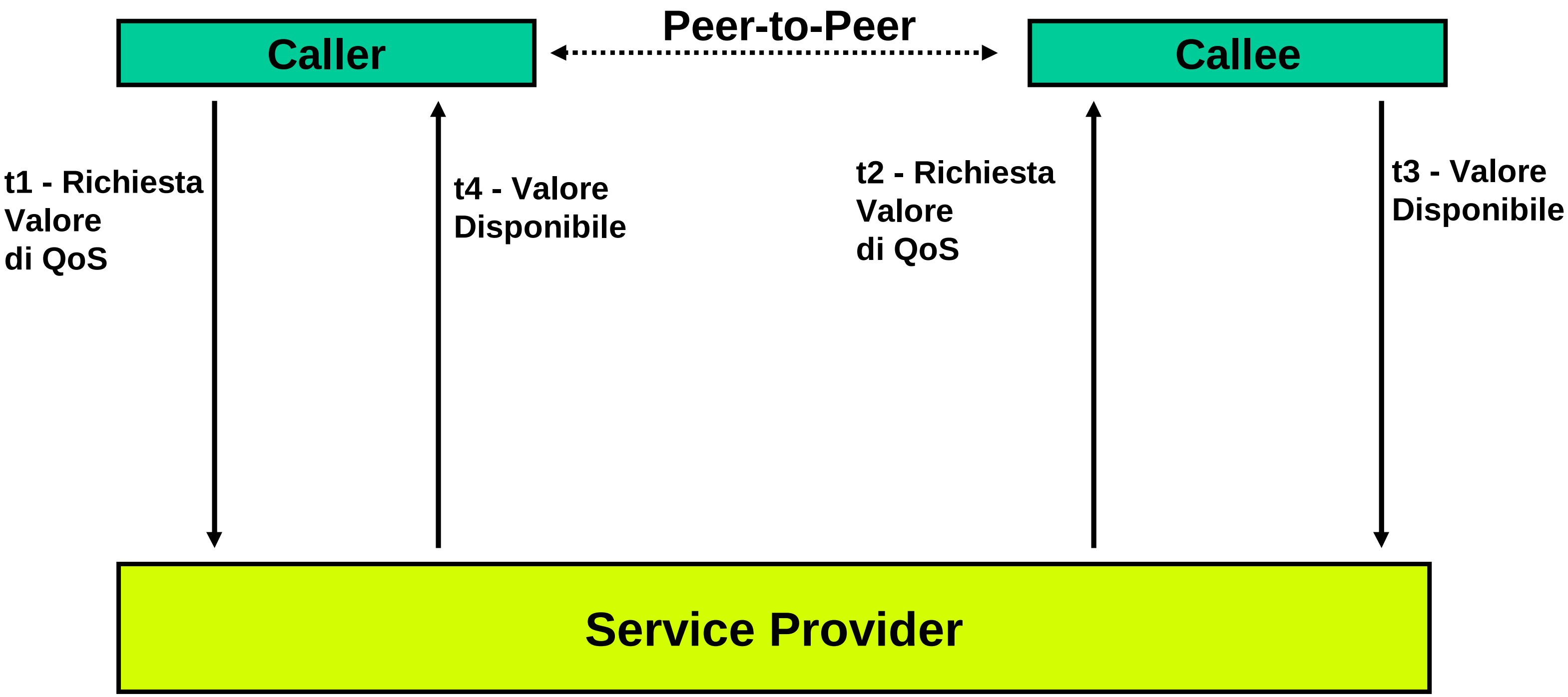
Peer-to-Peer Negotiation

- Detta anche (in ISO) caller-to-callee negotiation
- E' in linea di principio una negoziazione tra 2 entità (ex: applicazioni) dello stesso livello (o tipo)
- In realtà è una modalità emulata, per ottenere la quale si usa la comunicazione layer-to-layer

Negotiation



Bilateral peer-to-peer



Negoziazione tra i 2 peer, il Service Provider non interviene

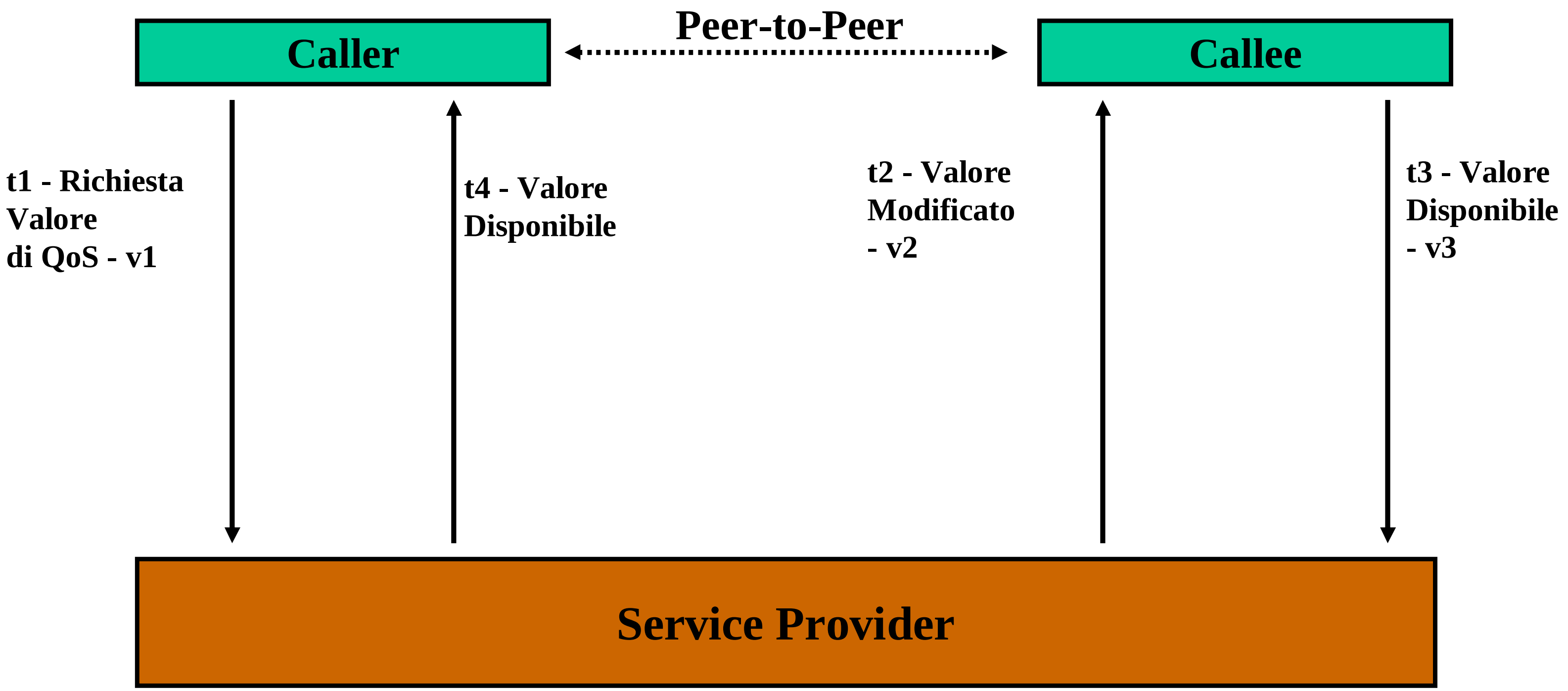
Bilateral Layer-to-Layer

- Negoziazione tra layer adiacenti
- Utente negozia col Provider
  - locale
    - ✓ applicazione con Sistema Operativo
  - in rete
    - ✓ host con network

Unilateral Negotiation

- Il Caller propone
- né Service Provider, né Callee hanno la possibilità di modificare i parametri richiesti
- Il Callee è messo di fronte a due sole possibilità (*take it* or *leave it*):
  - Accettare
  - Rifiutare

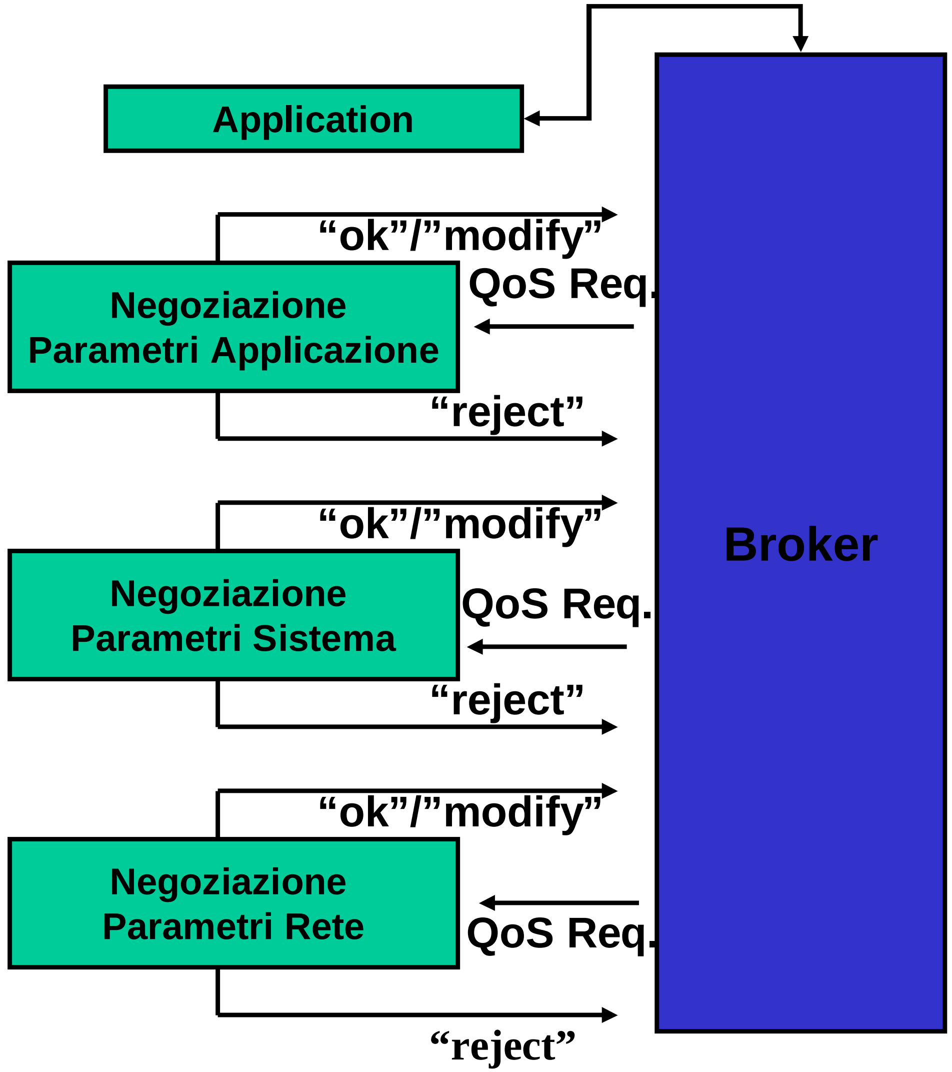
Triangular Negotiation



Caller, Callee e Service Provider partecipano

QoS Broker

- Bilateral Negotiation a livello Applicazione tra peers
- Unilateral Negotiation col Sistema Operativo
- Triangular Negotiation a livello di trasporto



Translation

- Differenti componenti richiedono differenti caratteristiche di QoS
- Nel passaggio da layer a layer può essere necessario tradurre le caratteristiche richieste in parametri specifici del layer che “eredita”
- La traduzione è un servizio fondamentale nella comunicazione layer-to-layer

Translation

- User - Applicazione
  - tuning service
- Applicazione - Sistema
  - esempio: video frame size -> transport frame size
- Sistema - Rete
  - esempio: ritardo minimo scheduling del pacchetto -> minimizzare tempo di spedizione

Scaling

- Possibilità di dover/poter considerare separatamente solo un sub-sample dello stream di dati, in modalità:
  - trasparente: drop di alcuni dati, gli strati superiori non sono informati
  - non-trasparente: necessaria interazione tra livello di trasporto e strati superiori
  - Esempio: modifica di parametri in algoritmi di codifica audio/video

Scaling: Audio

- Audio
  - può causare alto degrado del segnale
  - facilmente notato dall'orecchio umano
  - in genere di tipo non-trasparent
    - ✓ se ad esempio si decide di eliminare un canale stereo -> mono

Scaling: Video

- Video
  - applicabilità dipende dal tipo di compressione
- domini di applicabilità
  - Temporal Scaling: in video stream si riduce temporaneamente il numero di frame inviati
  - Spatial Scaling: riduzione del numero di pixel per immagine
  - Frequency Scaling: riduzione del numero di coefficienti in DCT (Discrete Cosine Transform) per la compressione (es. MPEG)
  - Amplitude Scaling: riduzione della profondità dei colori
  - Color Space Scaling: riduzione del numero di colori, esempio: passaggio da colori a scala di grigi

Caso di Studio (CS2)

Adaptive MPEG4 Video Streaming using Bandwidth Estimation

Mario Gerla, Alex Balk, Medy Sanadidi, Dario Maggiorini

Outline **(CS2)**

- Problem Statement
- Current Streaming Research Directions
- Unique Feature of VTP: Bandwidth estimation
- Testbed Evaluation

Problem Background **(CS2)**

- Internet multimedia streaming on the rise
- Most real-time video is still UDP (no end-to-end congestion control)
- UDP approach could potentially lead to congestion collapse
- Active research on congestion-controlled streaming protocols

Controlled Stream Approaches ***(CS2)***

- RAP (Rate Adaptation Protocol)

  Mimics TCP (i.e., AIMD); multilayer video adaptation

- SR-RTP (Selective Retransmission-RTP)

  “Binomial” rate control; selectively retransmits only certain packets that carry “key” video data.

- SCTP (Stream Control Transmission Protocol)

  Mimics TCP; multistream;

- TFRC (TCP-Friendly Rate Control)

  Mimics TCP via equation;

Limitations:

AIMD algorithm leads to rate oscillations

poor link utilization in random loss environments

VTP: Video Transport Protocol ***(CS2)***

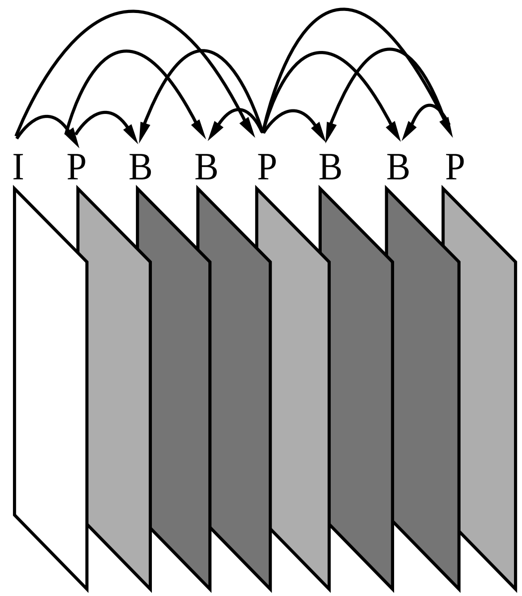
Key features:

- **bandwidth estimation** to adapt/reduce video stream
- MPEG specific adaptive **quantization levels** (while video frame rate is kept fixed to preserve perceived video quality).
- Inter-Protocol **fairness with TCP**.

Comments on MPEG-4

(CS2)

- **Intra-coded frames** (I-frames) are encoded independently, can be considered reference frames.
- **Predicted frames** (P-frames) depend on preceding I or P-frames, contain predicted motion data and error information.
- **Bi-directionally predicted frames** (B-frames) depend on both previous and next frames.



- VTP takes advantage of the wide **range of compression ratios** available in MPEG-4 to select appropriate video quality for streaming.

Bandwidth Estimation **(CS2)**

- Receiver estimates available bandwidth: unique feature of VTP.
- Bandwidth estimation technique (inspired to TCP Westwood):

$$B_i=(\alpha)B_{i-1}+(1-\alpha)(b_i+b_{i-1})/2$$

B_i: bandwidth estimate
b_i: bandwidth sample (i.e., packet bits in packet/interpkt interval)
α: tunable coefficient

- Receiver sends back to source bandwidth estimates periodically (at least, each RTT)

Digression: TCP Westwood ***(CS2)***

- Congestion window control via Achieved Rate Estimate (*RE*)
- Sender estimates currently Achieved Rate RE by extracting/filtering *rate samples* from ACKs
- After packet loss (ie, 3 DUPACKs, or Timeout), RE estimate is used by sender to cut back ***cwnd*** as follows:

  cwnd = RE x RTTmin

Note: cwnd = RE x RTTmin is the min window required to achieve rate RE without causing congestion

Additive Increase "Fair" Decrease (AIFD)

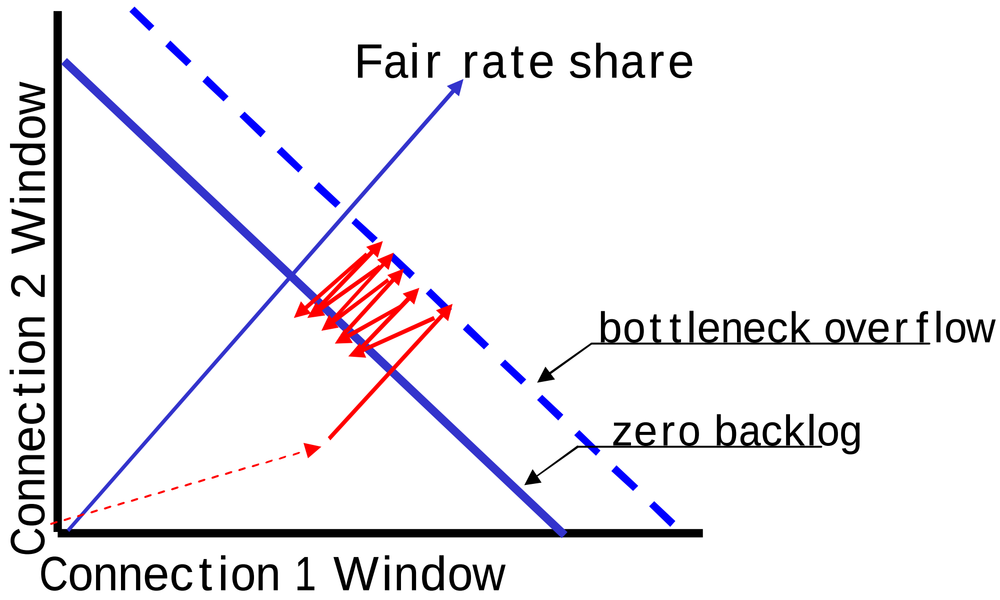
TCP Westwood: the control algorithm ***(CS2)***

- When three duplicate ACKs are detected:
  - ✓ set `cwin=RE*RTTmin` (instead of `cwin=cwin/2` as in Reno) and `ssthresh=RE*RTTmin` (instead of `ssthresh =cwin/2`)
  - ✓ `Go to` **`congestion avoidance`**
- When a TIMEOUT expires:
  - ✓ set `ssthresh=RE*RTTmin` (instead of `ssthresh=cwnd/2` as in Reno) and `cwin=1`
  - ✓ `Go to` **`slow start`**

Note: RTTmin = min round trip delay experienced by the connection

TCP W converges to Fair equilibrium

(CS2)



VTP algorithm

(CS2)

- **Multiple copies** of the video stream with different quantization levels are pre-computed and stored in the server (in the future, on line adaptive quantization will be explored)

- On the sender side: If estimated bandwidth feedback from receiver is **equal or larger to sent rate**, gradually increase sending rate by one packet per RTT (probing phase)

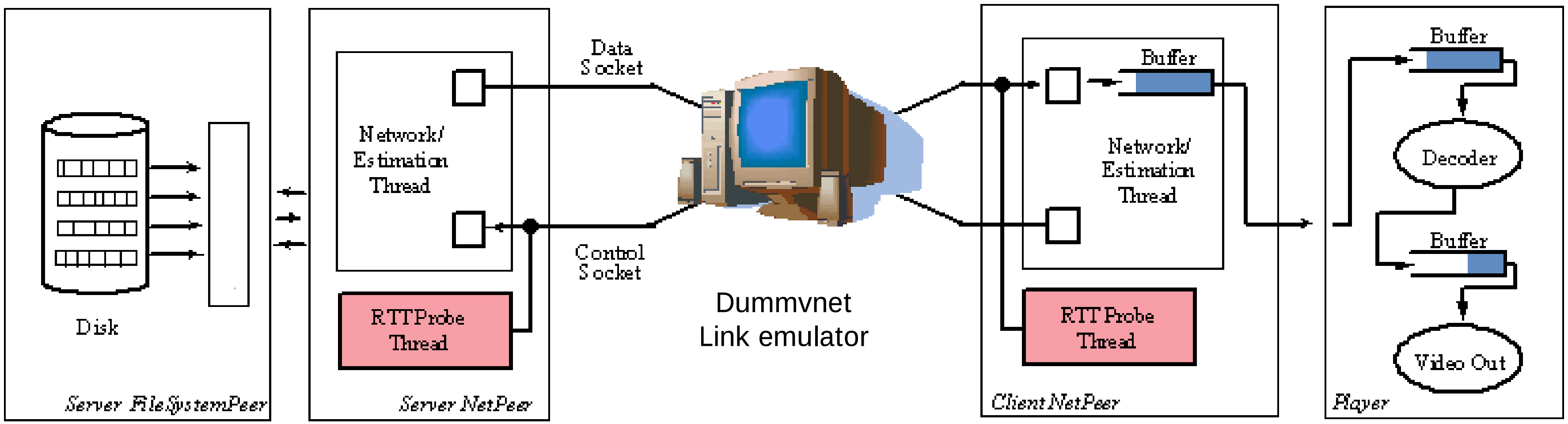
- When bandwidth **estimate can support next quantization level**, switch to higher quality stream and higher bitrate.

- If bandwidth estimate falls below current sending rate, **switch** to **lower quantization** level.

VTP: Linux testbed

(CS2)

- VTP was implemented and evaluated on the Linux operating system.



Software Architecture

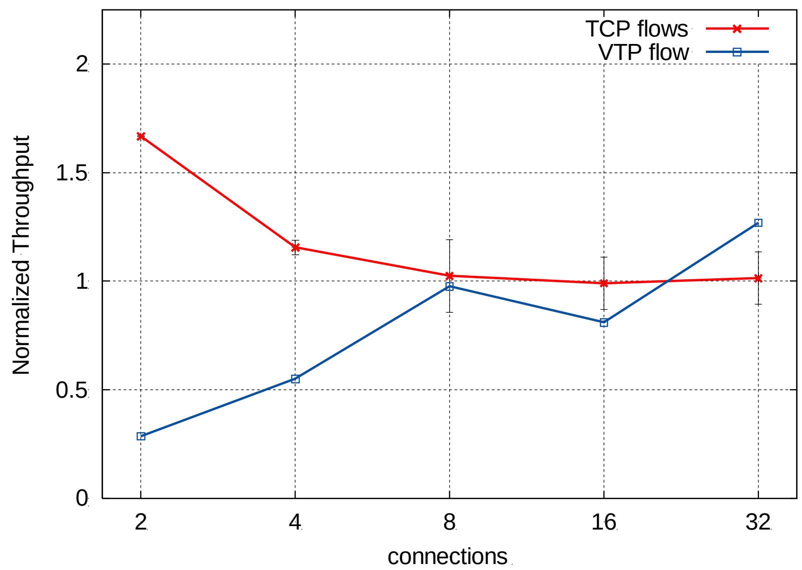
(CS2)

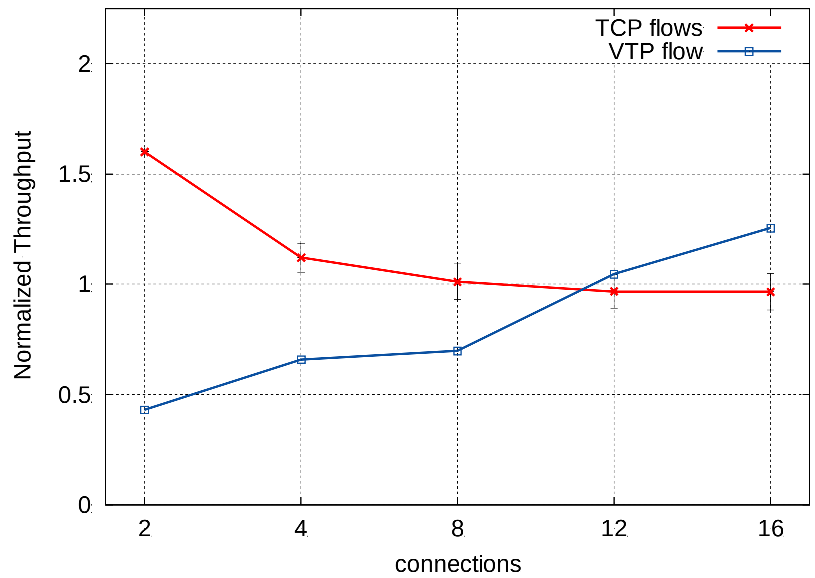
- VTP uses UDP to send both video packets and controls;
- Stream control and adaptation are done exclusively at application level;
- RTP and RTCP are not required, but can be integrated in the future (RTCP can be used for feedback)

Performance measures ***(CS2)***

- Fairness
- Stability
- Adaptive compression (QP)
- Robustness to random errors/loss

Fairness (1 VTP + N TCP Reno's) ***(CS2)***

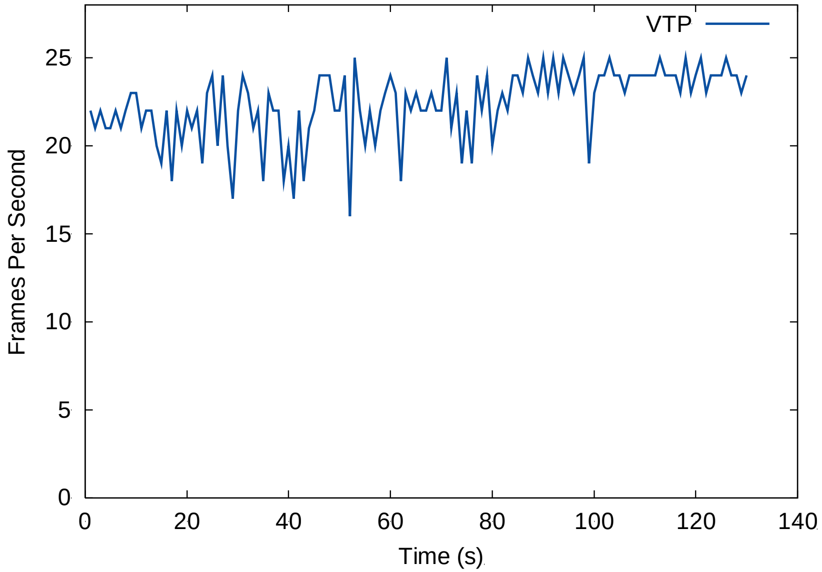


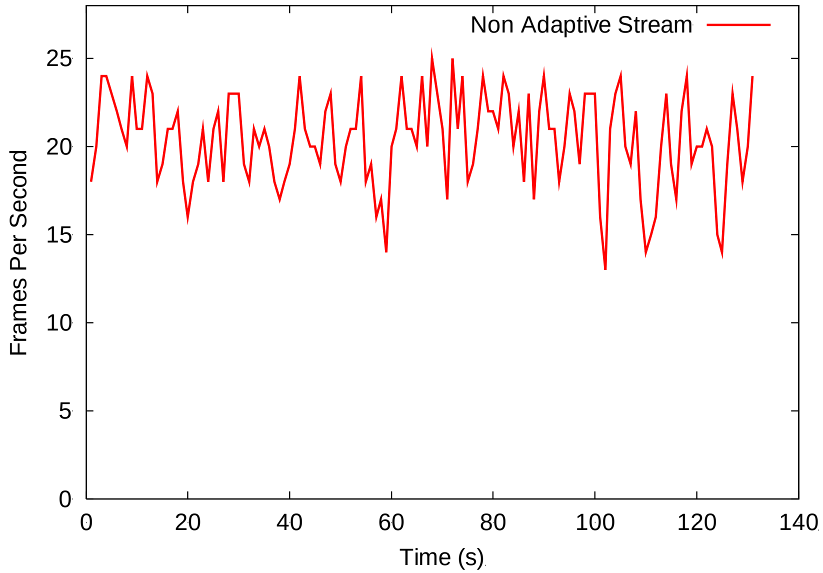


- VTP streams a segment of "Atlantis" encoded with the FFMPEG codec over a 3 Mbps, 10 ms RTT link.
- Normalized throughput is shown, ideal fairness occurs when both protocols achieve "1"(note: single VTP needs .45 Mbps)

- VTP streams a segment of "TRON" encoded with the DivX codec over a 5 Mbps, 10 ms RTT link.
- VTP uses its fair share of bandwidth or less in most cases.

VTP Rate Stability (1 video +11 TCP Reno's) **(CS2)**





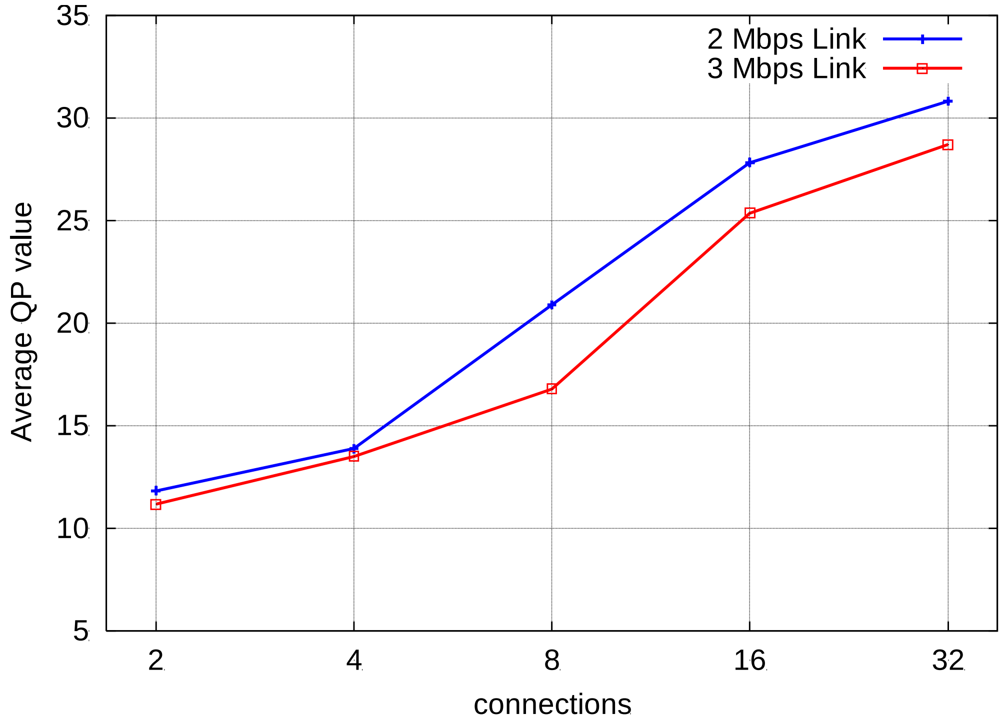
- Resulting frame rate of 1 monitored flow, either VTP or Non Adaptive Streaming, competing with 11 TCP connections.
- VTP frame rate stabilizes with time, as VTP discovers the available bandwidth. Non Adaptive Stream oscillates throughout the duration of the connection.

Adaptive Compression

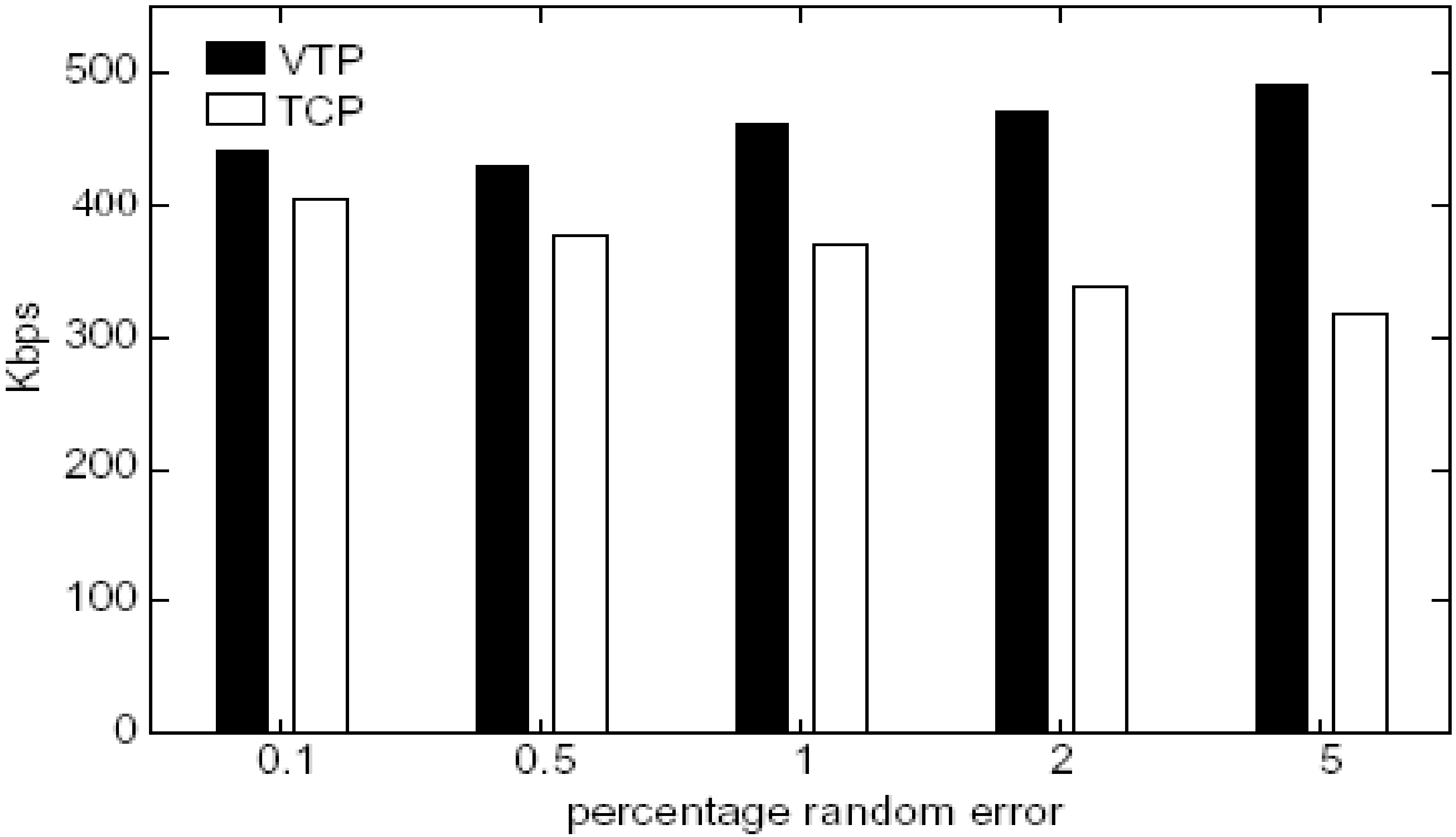
(CS2)

- "QP": quantization parameters (higher QPs imply more compression and lower video quality)

- As the number of connections increases, VTP transmits more compressed video streams using less bandwidth.

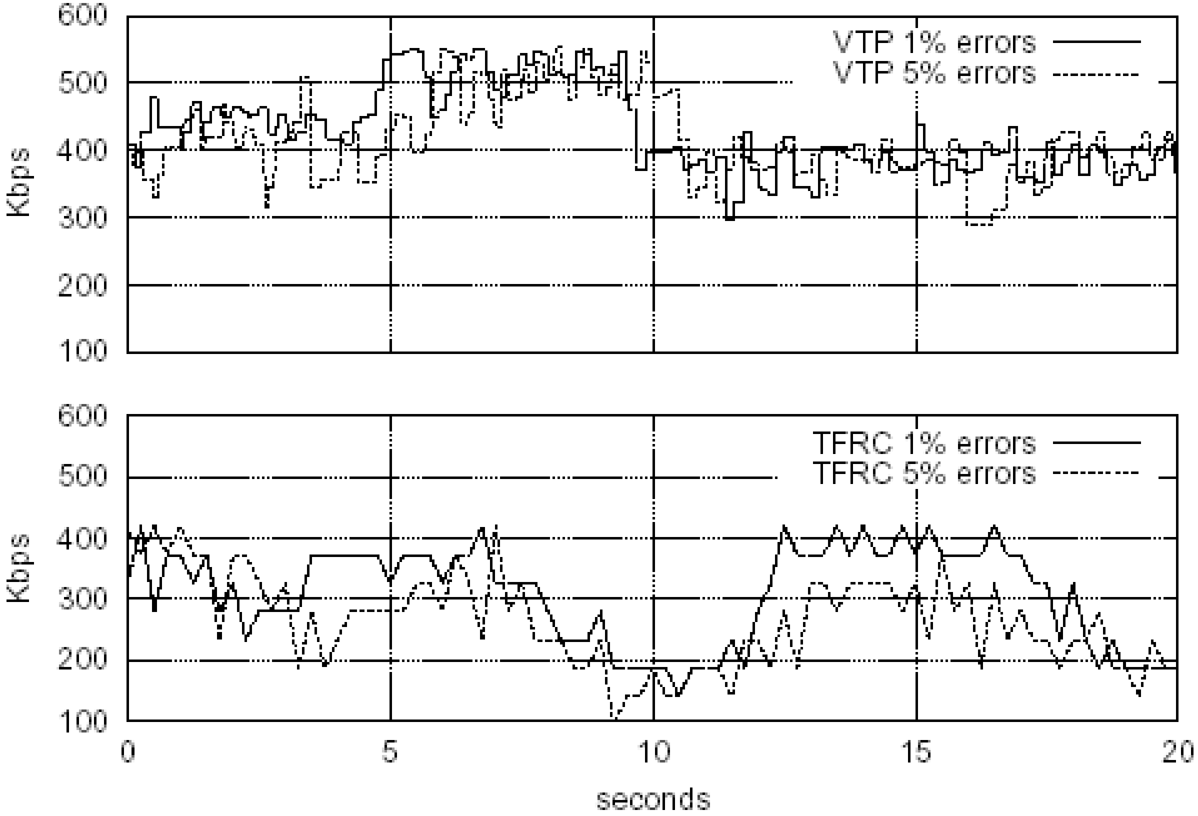


VTP & TCP Reno: random loss channel (CS2)



VTP and TCP sharing a random loss channel

VTP vs TFRC in random loss ***(CS2)***



VTP vs TFRC behavior in random loss – same video trace for both

Conclusions **(CS2)**

- VTP streams yield more stable frame rates than non-adaptive streaming.
- VTP fair to TCP
- VTP robust to random loss
- On going work:
  - Adaptive adjustment of receiver feedback reports
  - Investigation of other bandwidth estimation filters (e.g., AB probe)
  - Object oriented adaptive encoding

FINE Caso di Studio (CS2)