Congestion Control for High Performance Networked Multimedia

Colin Perkins

http://csperkins.org/

USC INFORMATION SCIENCES INSTITUTE

UNIVERSITY *of* GLASGOW

Talk Outline

- Background
  - Real-time networked multimedia
  - Congestion control in IP networks
- Challenges deploying congestion control
  - Due to the media
  - Due to the congestion control
- Initial experiments
- Future directions
  - Research
  - Standardisation

Real-time Networked Multimedia

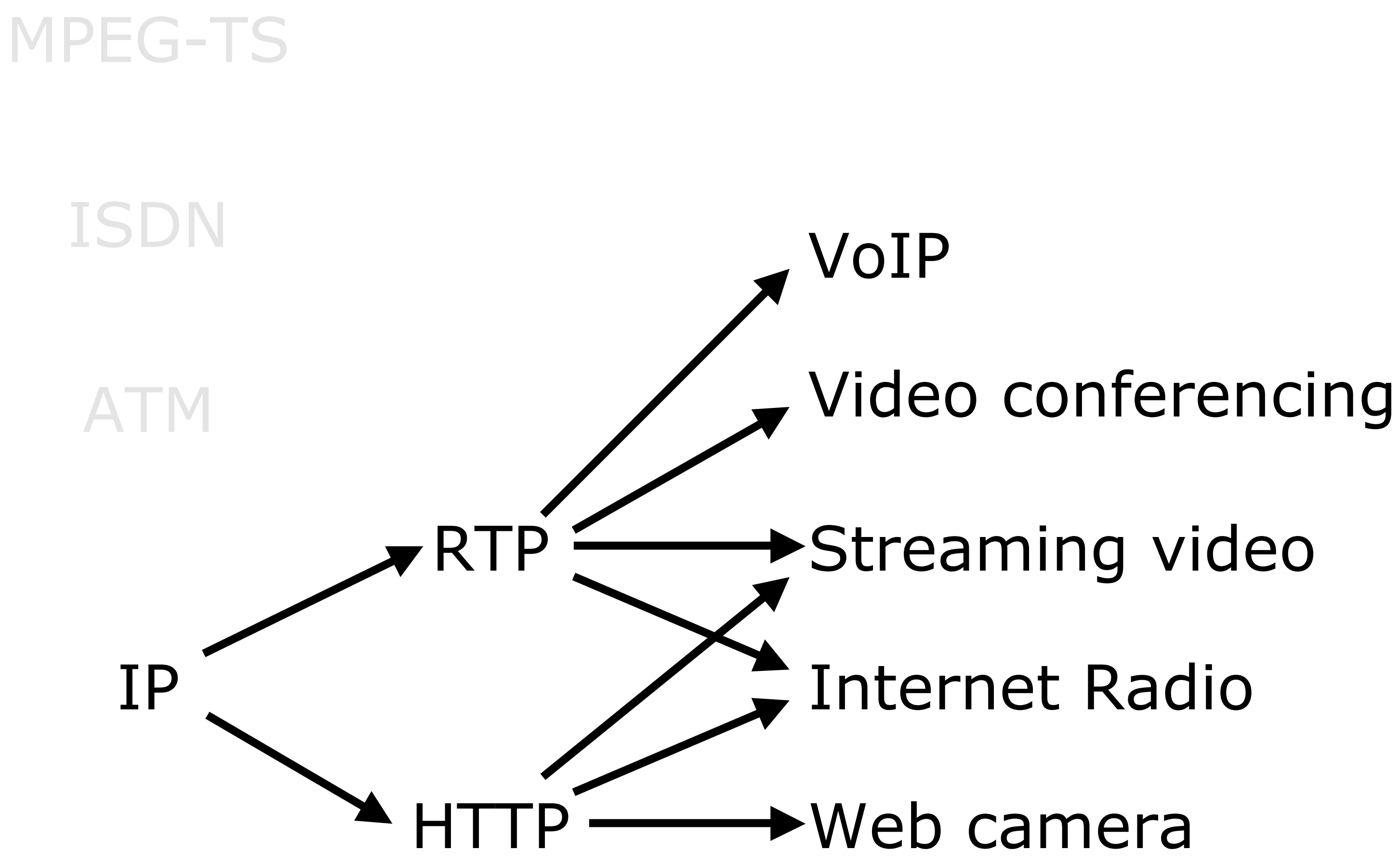
MPEG-TS

ISDN

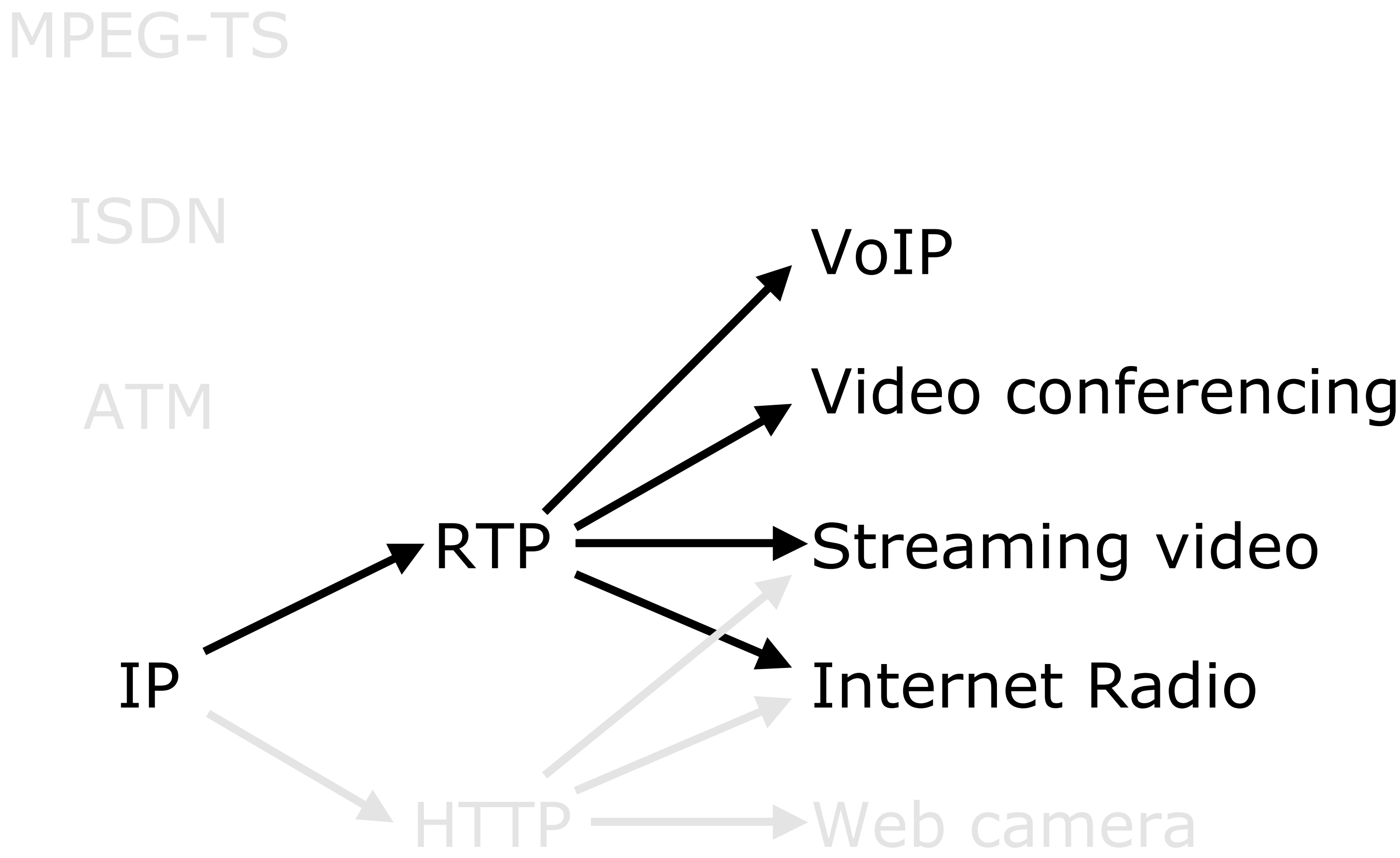
ATM

IP

Real-time Networked Multimedia



Real-time Networked Multimedia



RTP: Real-time Transport Protocol

- The standard for real-time transport over IP networks
  - Video conferencing
  - Voice over IP/telephony
  - Streaming audio and video

- Published as an IETF "draft standard" RFC
  - RFCs 3550 and 3551 in July 2003 updating earlier RFCs
  - Widespread use on conferencing
    - Adopted by ITU as part of H.323
    - Mbone tools; AccessGrid
    - Apple iChat; Windows Messenger
  - Adopted by 3GPP for next generation cellular telephony
  - Some use in streaming
    - QuickTime, Real, Microsoft
    - (competing with installed base of HTTP streaming; firewalls)

RTP: Real-time Transport Protocol

- RTP delivers a single media stream from sender to one, or more, receivers
  - Provides:
    - Participant identification
    - Reception quality statistics
    - Codec identification
    - Media transport
      - Padding, if necessary
      - Marking of significant events
    - Sequencing
    - Timing recovery
- Typically implemented in an application or as a library
  - User level, not part of the kernel
- Few assumptions about the underlying transport
  - Datagram service; not necessarily reliable or ordered
  - Usually runs over UDP/IP

Philosophy of RTP

- The challenge:
  - build a mechanism for robust, real-time media delivery above an unreliable and unpredictable transport layer
  - without changing the transport layer

Push responsibility for media delivery onto the end-points where possible	Make the system robust to network problems; media data should be loss tolerant
The end-to-end argument	Application level framing

Philosophy of RTP

- Implication: smart, network-aware, applications that are capable of reacting to problems end-to-end
  - Both sender and receiver are intelligent
  - The network is dumb and can be unreliable
- Fits well with the best-effort IP service
  - Applications handle reliability and congestion control
  - Doesn't require QoS support or congestion control in the network
- Contrast with traditional applications:
  - Telephone network is smart, end-points are dumb
  - MPEG sender is smart, receiver relatively dumb

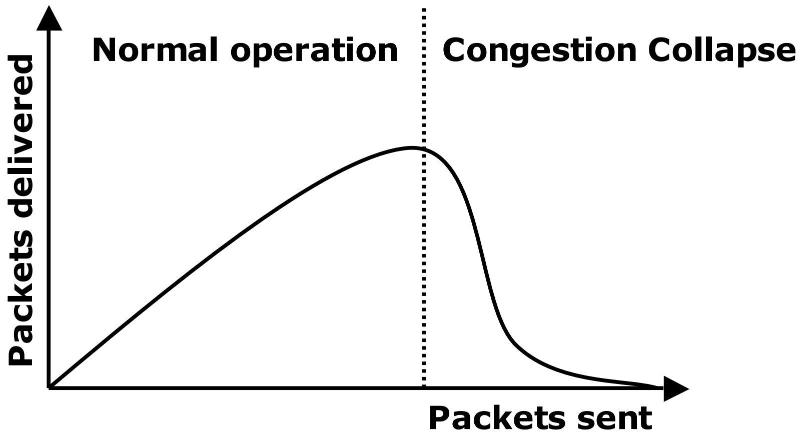
Mapping RTP onto UDP/IP

- An IP network provides:
  - Best effort packet delivery with no admission control
  - Packets are discarded at intermediate routers if the output links are congested

- Layers above IP are expected to react to packet loss:
  - As a signal to perform some loss recovery algorithm
    - Retransmission
    - Forward error correction
    - Loss tolerance
  - As a signal to reduce their sending rate
    - TCP/IP has a standard algorithm
    - Multimedia traffic, using RTP on UDP/IP, does not

Why Congestion Control?

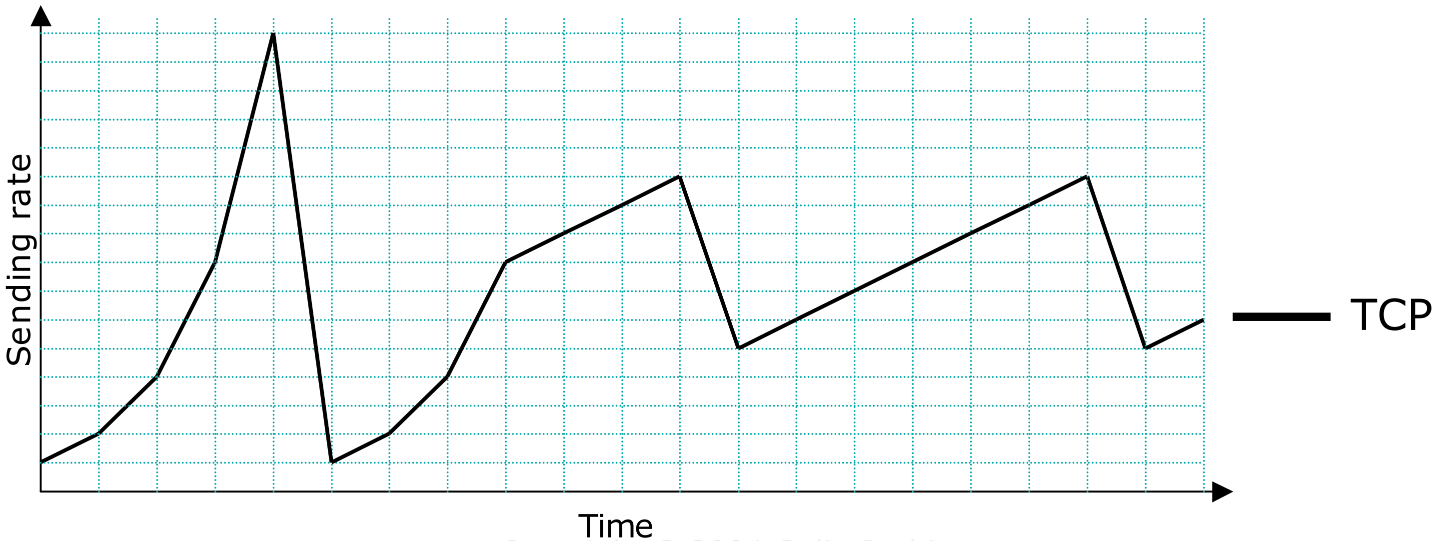
- Important to devise congestion control for multimedia:
  - For very high quality, need to fit the capacity of the pipe
    - Applications can be bandwidth hungry
  - For widespread deployment, need to ensure that the aggregate traffic is adaptive to capacity changes
  - To avoid congestion collapse



- In all cases, need to ensure that the media quality isn't affected by the adaptation

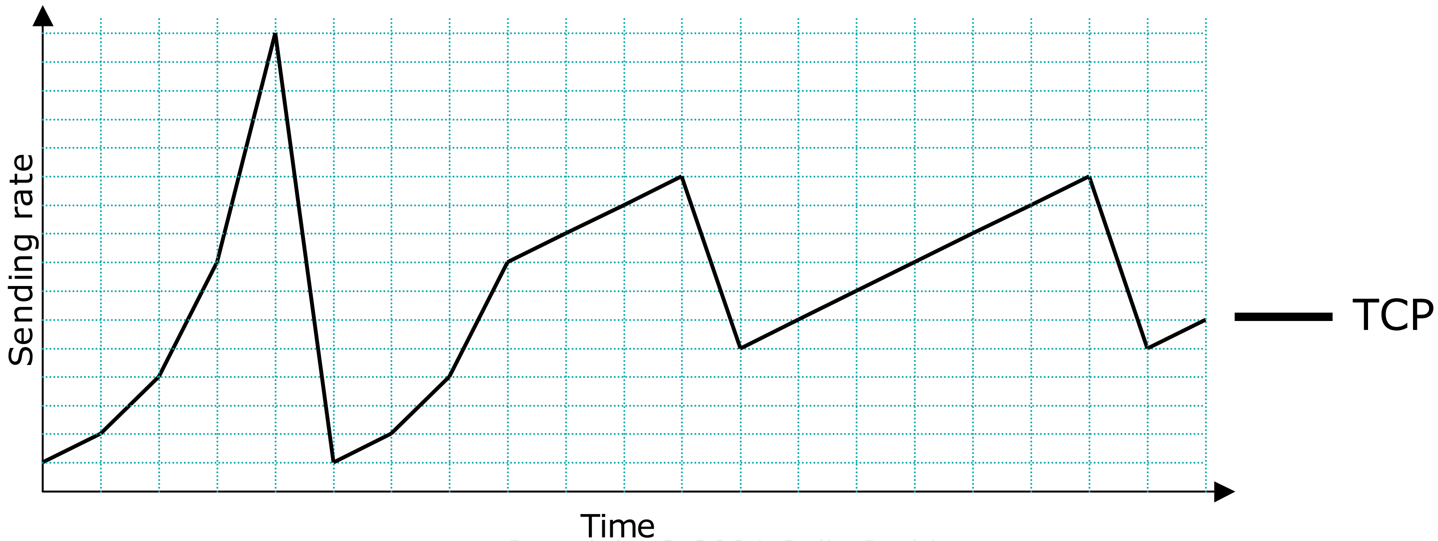
TCP Congestion Control

- Most traffic is TCP/IP
  - Additive increase/multiplicative decrease
    - Linear probe of available capacity until momentary overload
    - Multiplicative back-off to safe sending rate
  - Ensures capacity is used, avoids network overload
  - Approximately equal share of bottleneck capacity between flows



TCP Congestion Control

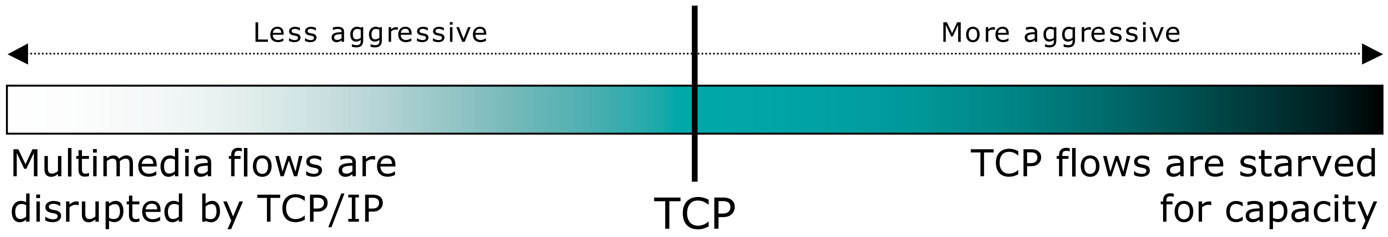
- Works well for elastic applications
  - In particular, long lived bulk transfers
- Bad for multimedia because…
  - Rate is highly variable, and profile doesn't match multimedia traffic
  - Couples congestion control with reliability
  - ⇒ Streaming audio/video over HTTP sub-optimal



Multimedia and Fairness

- Implication: multimedia flows need a different sort of congestion control

- But, the network constrains possible solutions:



- Want non-TCP traffic to be *TCP-friendly*
  - Compete fairly with TCP on average
    - But different dynamics and reliability modes
  - Assert priority with QoS mechanisms, if desired

How to be TCP-Friendly?

Derive a mathematical model of average TCP throughput

Use that to drive congestion control

Current best model due to *Padhye et al.* [SIGCOMM '98]:

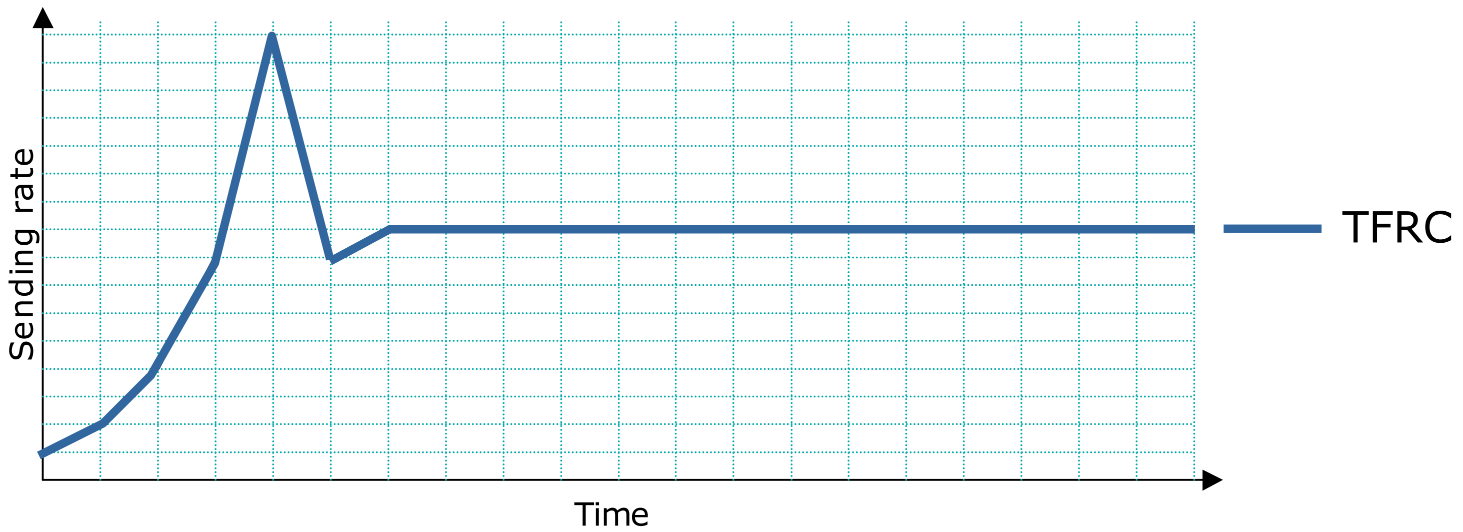
$$T = \frac{s}{R\sqrt{\frac{2p}{3}} + 3p(1+32p^2) \cdot T_{rto}\sqrt{\frac{3p}{8}}}$$

Assumptions:

- Saturated steady state TCP sender
- TCP Reno
- Packet loss correlated within sending window, uncorrelated long term
- Packet reordering rare

TFRC Protocol

- Padhye's throughput model forms the basis of a standard congestion control protocol: **TFRC** [RFC 3448]
  - Embed the throughput equation into an ACK-based feedback protocol
  - Rules for derivation of loss event rate from packet loss history
  - Slow start; damping to avoid oscillation



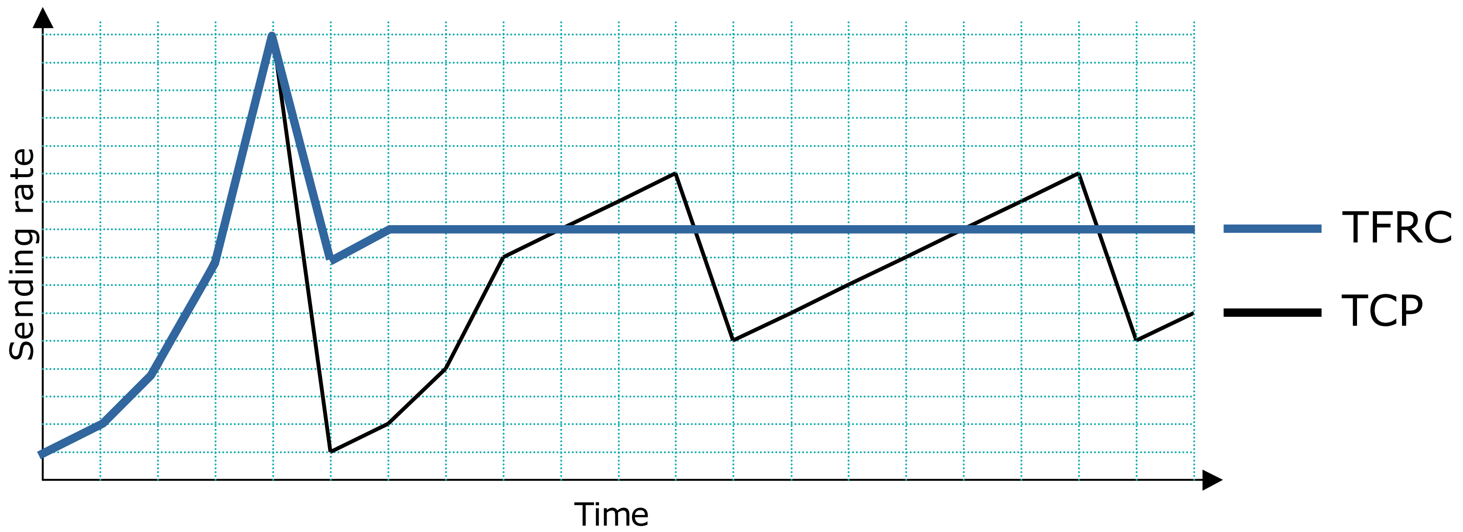
Comparison of TCP and TFRC

TFRC:

- Smoother, approximately matches average TCP rate
- Decouples congestion control from reliability

TCP:

- Faster adaptation to changes in capacity



Deployment Challenges

- Claim in IETF community is that TFRC is appropriate for multimedia flows

- Not widely tested, especially for interactive use:
  - TCP friendly algorithms are new, and evolving
  - Do the assumptions of network performance hold?
    - TCP variant, loss patterns, jitter, reordering
  - Do the assumptions match the needs of the application?
    - Interactions between codec and network are not well defined
    - Unclear how slow response, limited adaptability, impact fairness
  - Human factors aspects play a key role
    - Congestion control implies variable quality
    - Subjectively very annoying, unless the rate of change is slow
    - Can have a significant impact on congestion control
  - How can we build systems using this framework?
    - How to integrate into RTP? New protocols?

- Build a prototype, to find out if it works…

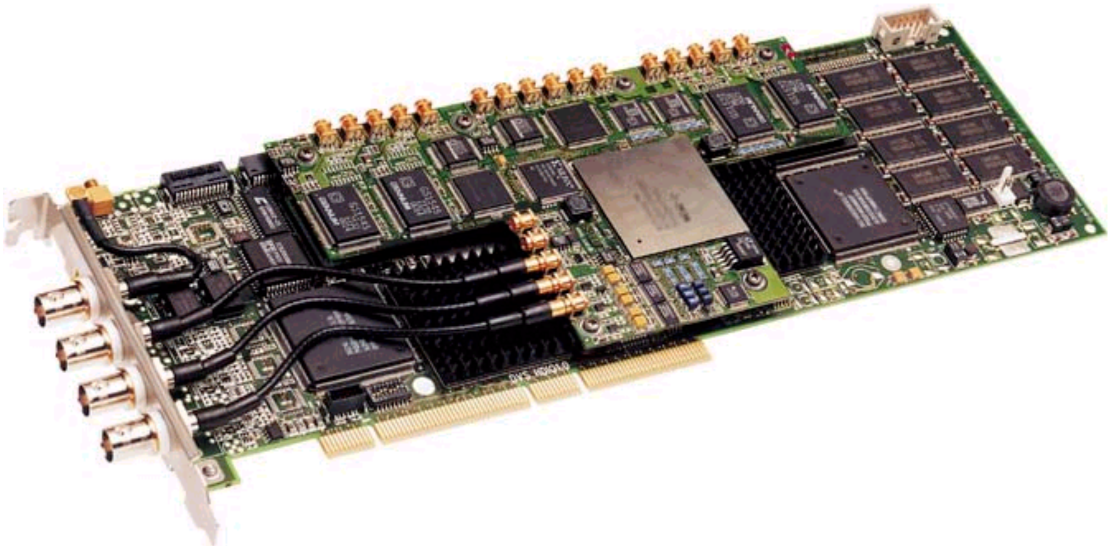
HDTV over IP Demonstrator

- Develop a very high quality teleconferencing system:
  - High Definition TV (HDTV)
    - 1280x720 @ 60Hz
  - Gigabit Ethernet
  - Wide-area OC-48 networks
  - High performance PCs

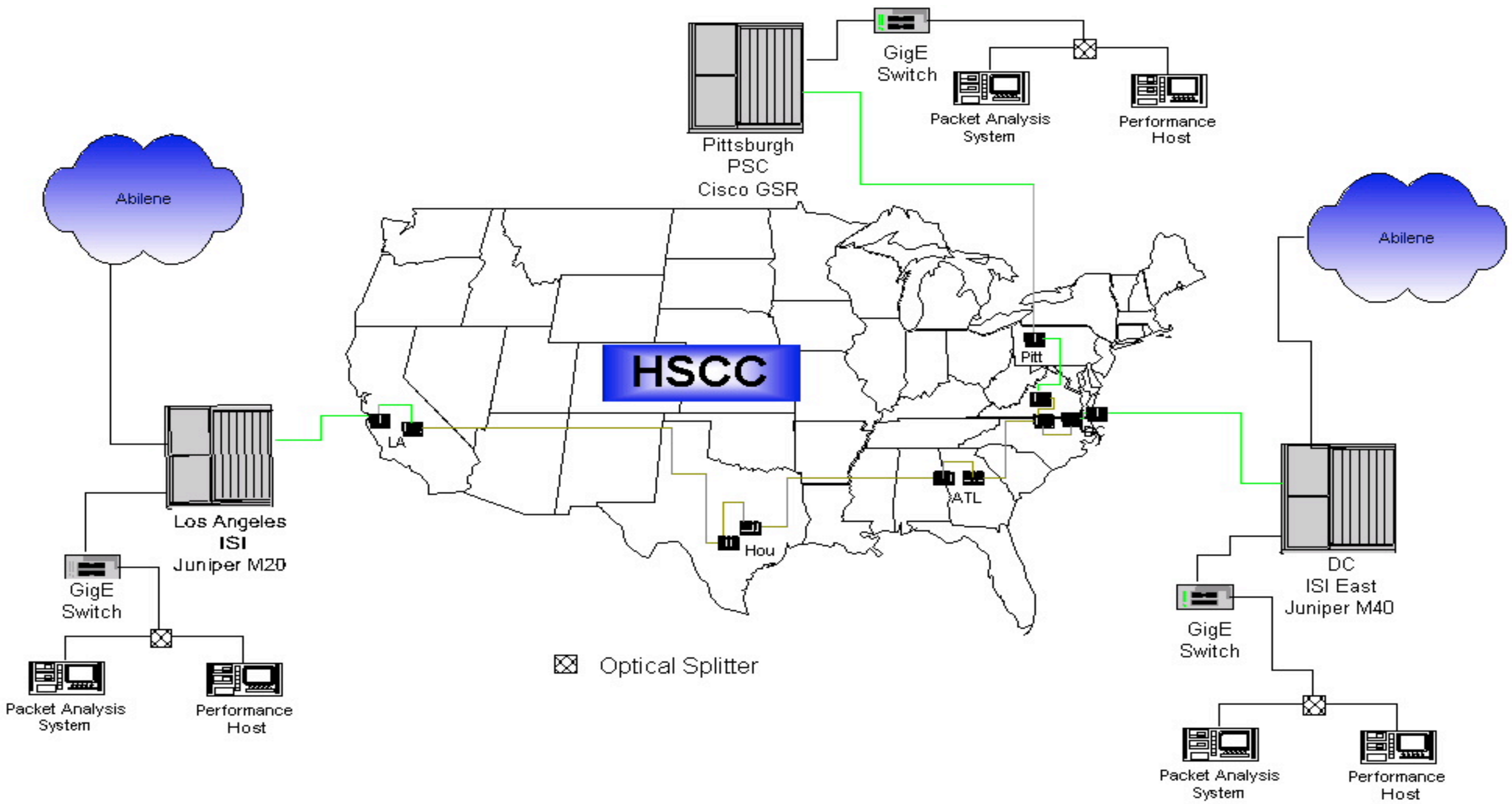
- Several aims:
  - Demonstrate high quality media over IP networks
  - Demonstrate operation of TFRC congestion control
    - Test if TFRC network performance assumptions realistic
    - Test human factors of media adaptation
  - Demonstrate scalability of the protocols

HDTV over IP Demonstrator

- Transmitter and receiver hosted on separate PCs
  - Dell PowerEdge 2500 servers
  - 1.2GHz PIII Xeon/Dual 64 bit PCI
  - Linux 2.4
- Gigabit Ethernet
  - Sub-sampled colour ⇒ 850 Mbps
- HDTV video capture card
  - DVS HDstation OEM
- Philips LDK-6000 HDTV Camera
- The combination makes HDTV grabbing and transport feasible on commodity hardware
  - Linux PC + HDTV grabber
  - Approximately $20k + HDTV camera and display
- Software available for download



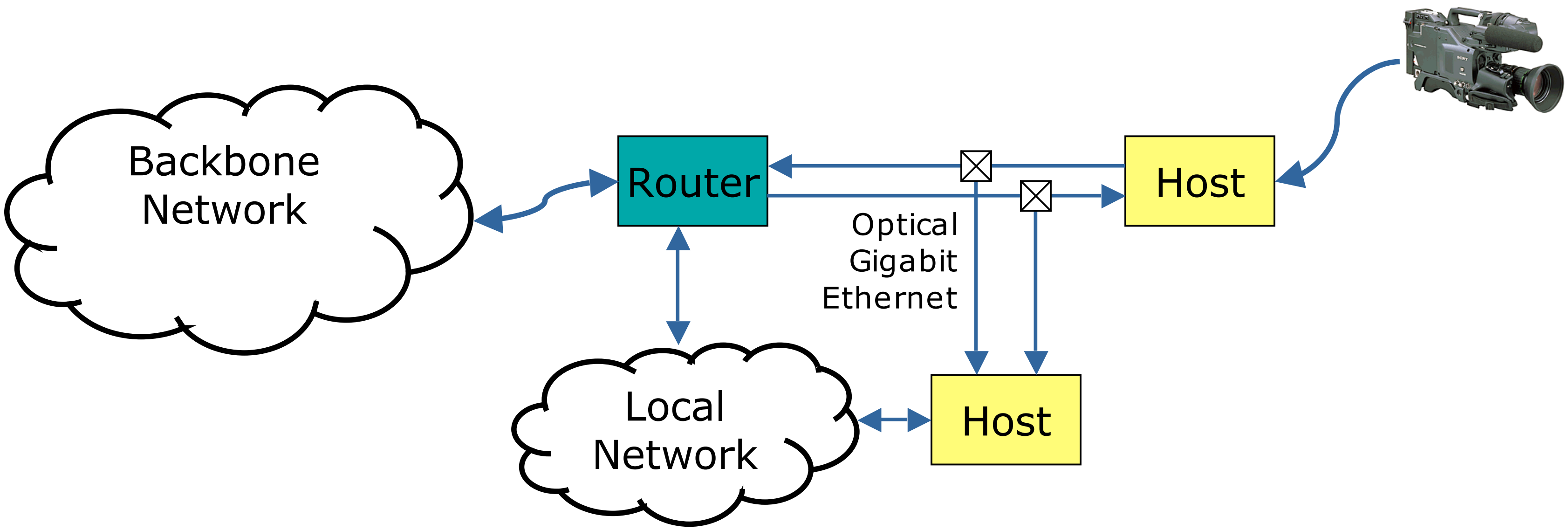
Test Environment



- System tested between ISI sites in Washington DC and Los Angeles
  - Early demonstration at SuperComputing 2002
- HSCC is a DARPA tested, routed on a commercial ISP's IP backbone
  - OC-48 shared with commercial IP traffic; no QoS support

Test Environment

- Optical splitter on gigabit Ethernet for traffic monitoring
- Monitoring host is a FreeBSD system
  - Dual gigabit Ethernet with only Rx connected
  - Capture packet headers to memory at line rate
  - Careful tuning to avoid discards



Measurements

- Capture packet traces from cross-country HDTV tests
  - Plus synthetic packet generator in some cases
  - Some fixed rate, some using TFRC
- Measure:
  - Packet loss
  - Timing variation/jitter
  - Reordering
  - Duplication
    - Linux will duplicate; wide area network doesn't seem to…

Packet Loss Rates

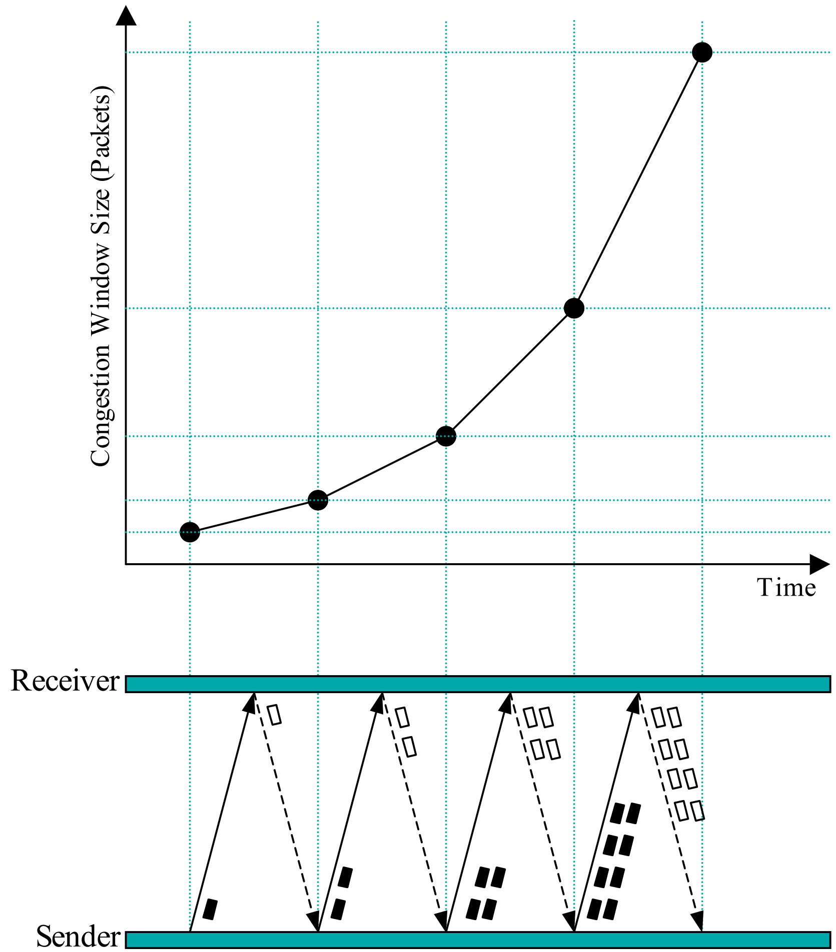
Loss event duration	Frequency	
No loss	24697400	
Single packet	85797	(0.3%)
Two consecutive packets	587	(0.002%)
Three consecutive packets	7	(0.00003%)
Four or more packets	0	

- Cross country path: DC to LA
- When the path is adequately provisioned, loss is rare
  - Data above is worst-case
  - Many hour-long traces with no loss
- We believe this is typical for major ISP backbone networks
  - Problems due to access networks/interconnects/hosts

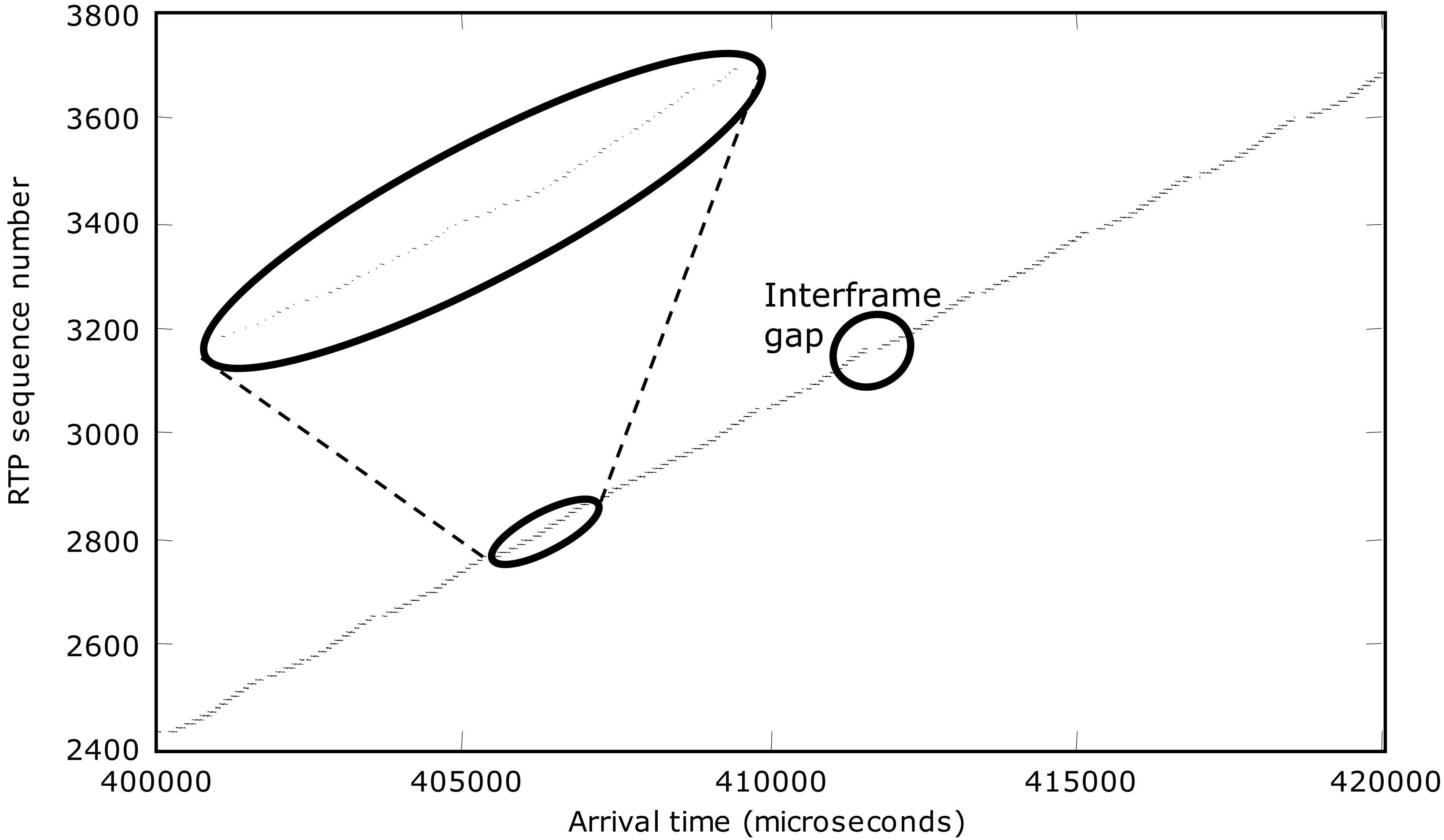
Packet Loss Rates

- Padhye equation models TCP Reno and assumes loss within window is correlated
  - Lose one, lose remainder of window
- Measurements do not show this behaviour in network
  - Losses are isolated events when window is large
  - TCP Reno is pessimistic

⇒ TFRC also pessimistic
  - Better to model SACK TCP

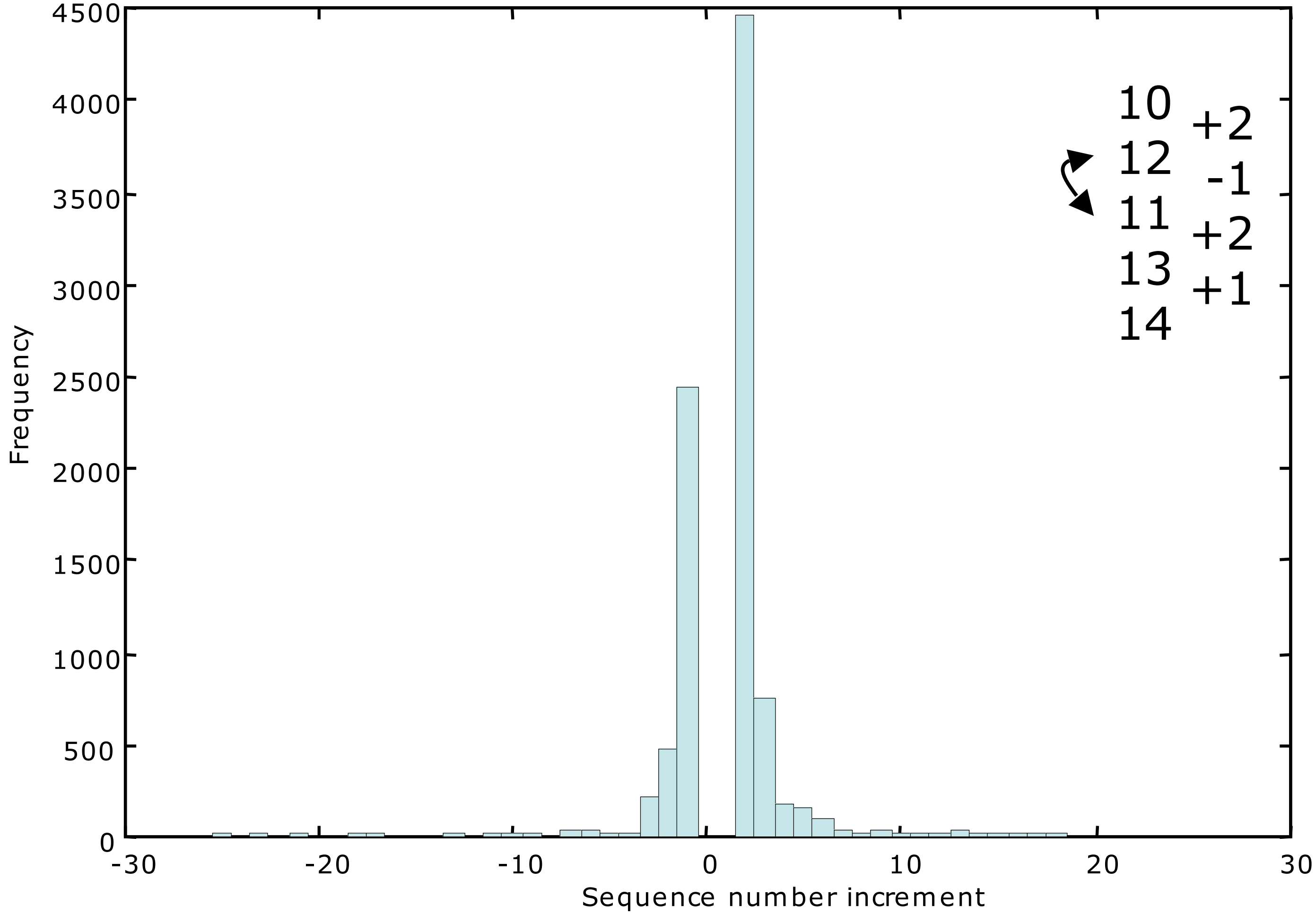


Packet Timing Variation



- Observed >99.9% of packets in order, with negligible jitter

Packet Reordering



- Packets are occasionally reordered in the network
  - (above data is from a 10 million packet trace)
- Significant effects on congestion control, even though all packets arrive

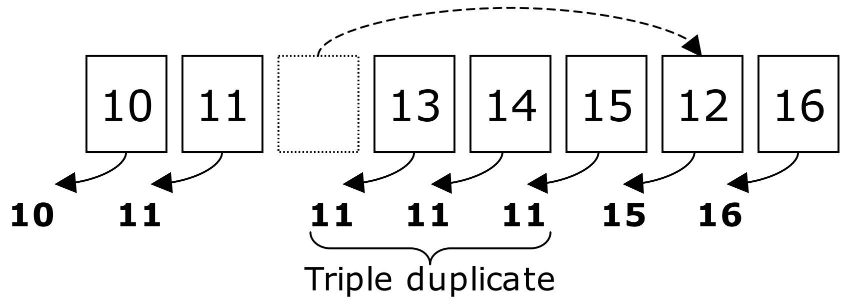
Packet Reordering

Path	Size	Metric	100Mbs	200Mbs	300Mbs	400Mbs	500Mbs	600Mbs	700Mbs	800Mbs	900Mbs
		$M1\%$	-	0.07	0.45	1.26					
	500	$M2\%$	-	0.13	0.90	2.53	N/A	N/A	N/A	N/A	N/A
		$M3$ (events)	-	0	0	0					
		Loss (packets)	-	0	0	0					
		$M1\%$	-	-	-	-	-	0.01	0.02	0.05	
LA→DC	1500	$M2\%$	-	-	-	-	-	0.02	0.04	0.10	N/A
		$M3$ (events)	-	-	-	-	-	0	0	0	
		Loss (packets)	-	-	-	-	-	0	0	0	
		$M1\%$	-	-	-	-	-	-	-	-	-
	4500	$M2\%$	-	-	-	-	-	-	-	-	-
		$M3$ (events)	-	-	-	-	-	-	-	-	-
		Loss (packets)	-	-	-	-	-	-	-	-	-
		$M1\%$	0.05	0.16	0.81						
	500	$M2\%$	0.09	0.30	1.61	N/A	N/A	N/A	N/A	N/A	N/A
		$M3$ (events)	15	191	783						
		Loss (packets)	0	0	0						
		$M1\%$	0	0.01	0.03	0.06	0.12	0.38	0.55		
DC→LA	1500	$M2\%$	0.01	0.02	0.06	0.12	0.23	0.71	1.10	N/A	N/A
		$M3$ (events)	0	0	2	21	88	3299	1049		
		Loss (packets)	0	0	0	0	0	0	0		
		$M1\%$	-	-	-	-	0	0.01	0.02	0.06	0.13
	4500	$M2\%$	-	-	-	-	0.01	0.02	0.05	0.12	0.26
		$M3$ (events)	-	-	-	-	0	0	0	4	12
		Loss (packets)	-	-	-	-	0	0	0	0	0

- Reordering strongly dependent on inter-packet spacing
- Caused by:
  - router bugs; link layer multi-path; etc.

Packet Reordering

- Reordering by >3 is treated as loss by TCP:



- TFRC follows TCP, and treats some reordering as loss
- Observed traces where *all* packets were received, but TFRC suggested throughput order-of-magnitude less than achieved
  - Due to reordering being treated as loss!
- Media decoding is generally tolerant of reordering
- Reordering robust TFRC desirable
  - But has implications for fairness

Summary of Congestion Control Results

- Packet loss is mostly isolated or short burst
  - TCP Reno under-performs ⇒ TFRC under-performs
- Timing mostly well behaved
- Small amount of reordering has disproportionate effect
  - ⇒ Reordering tolerant TFRC needed (Fairness?)

⇒ Most significant effect on TFRC observed: reordering at high rates

⇒ Network assumptions of TFRC justified at low-rate; not at high rate

⇒ (Implementation ongoing to study quality variation issues)

Future Research Directions

- Clear that TFRC brings undesirable influences from TCP
- Can evolve TCP-Friendly congestion control in several directions:
  - Evolve response to network effects
    - Tolerance to reordering at high speeds
  - Evolve response to TCP variants
    - Reno vs SACK
    - Some applications fall into the realm of HS-TCP; FAST-TCP; etc
  - Evolve awareness to needs of media
    - Dealing with bursty codecs; network queuing capacity
    - Limited codec adaptability; slow response; ad-hoc TFRC-changes
- Must consider how we affect other traffic as we do so
  - Difficult fairness questions relating to TCP traffic
- How do we scale to multicast/multiparty?

Future Standards Development

- Datagram Congestion Control Protocol
  - http://www.icir.org/kohler/dcp/
  - "Congestion controlled UDP"; implement in operating system
  - Incorporates several congestion control algorithms
    - TCP-like
    - TFRC
    - (more can be added later)
  - Possible long-term solution; difficult to deploy
- Directly incorporate TFRC into RTP
  - RTCP extensions to provide feedback
  - Short-term solution; implement in applications/libraries to allow experimentation; simple to deploy
- TFRC itself is controversial
  - Disconnect between codec designers and protocol designers

Summary

- Background
  - Real-time networked multimedia
  - Congestion control in IP networks
- Challenges deploying congestion control
  - Due to the media
  - Due to the congestion control
- Initial experiments
- Future directions
  - Research
  - Standardisation

Questions?

UNIVERSITY
of
GLASGOW