IEEE International Conference on Multimedia Computing and Systems

Issues in Multimedia Delivery Over Today's Internet

Kevin Jeffay
Department of Computer Science
University of North Carolina at Chapel Hill
jeffay@cs.unc.edu
June 28, 1998

http://www.cs.unc.edu/~jeffay

Issues in Multimedia Delivery on Today's Internet

Outline

- Domain of discourse
  - » Definitions, concepts, and objectives
- Performance of "naive" applications today
  - » What's "broken"?
- Media adaptations for best-effort multimedia delivery
  - » Can we fix "it"?
- Performance of best-effort applications today
  - » Fundamental challenges for tomorrow's Internet

Multimedia Delivery on Today's Internet
Domain of discourse

- A prototypical videoconferencing system
  - » Architecture
  - » Quality-of-service requirements
- Performance metrics
  - » End-to-end latency
  - » Packet loss
- Some typical experimental results

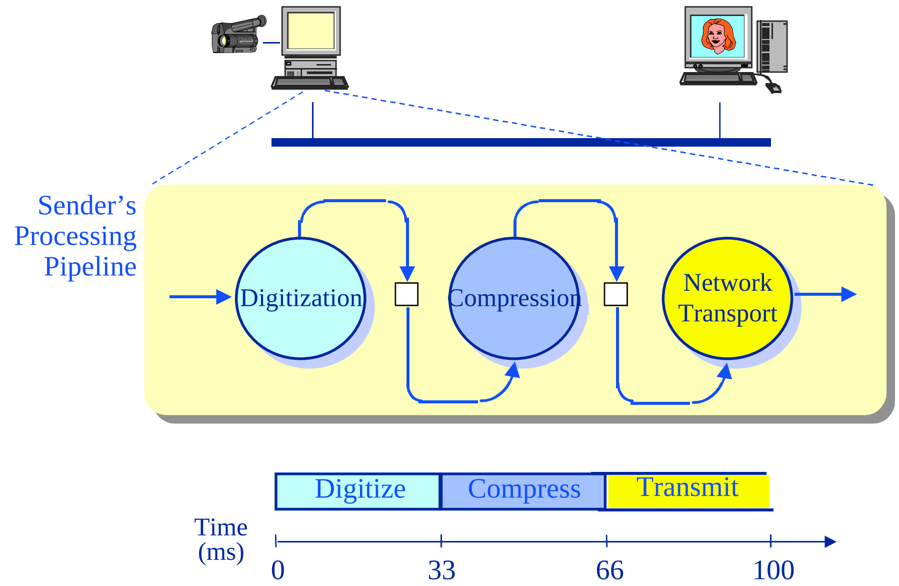
Interactive Multimedia Applications
Performance requirements

- Latency — the duration between acquisition of a signal and its display

- Videoconferencing latency requirements
  - » telephony literature — 100 ms roundtrip
  - » multimedia networking literature — 250 ms one-way
  - » CSCW literature — tolerance of latency as high as 400 ms

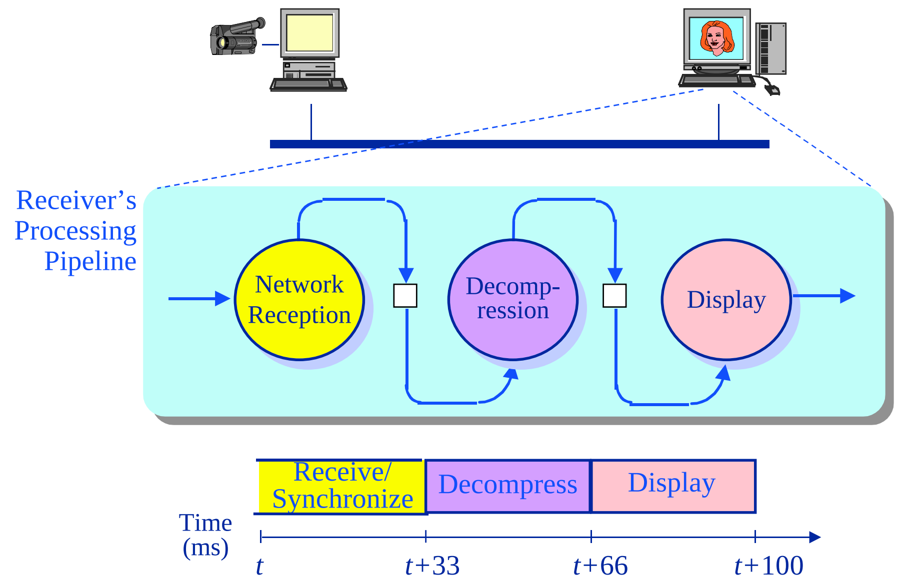
Latency in Computer-Based Video Systems

Canonical application structures



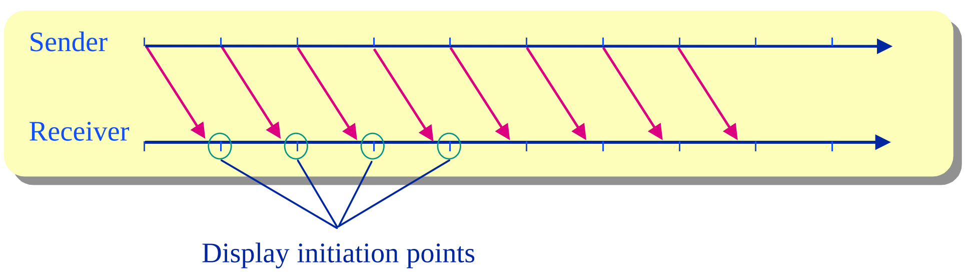
Latency in Computer-Based Video Systems

Canonical application structures



Latency in Computer-Based Video Systems

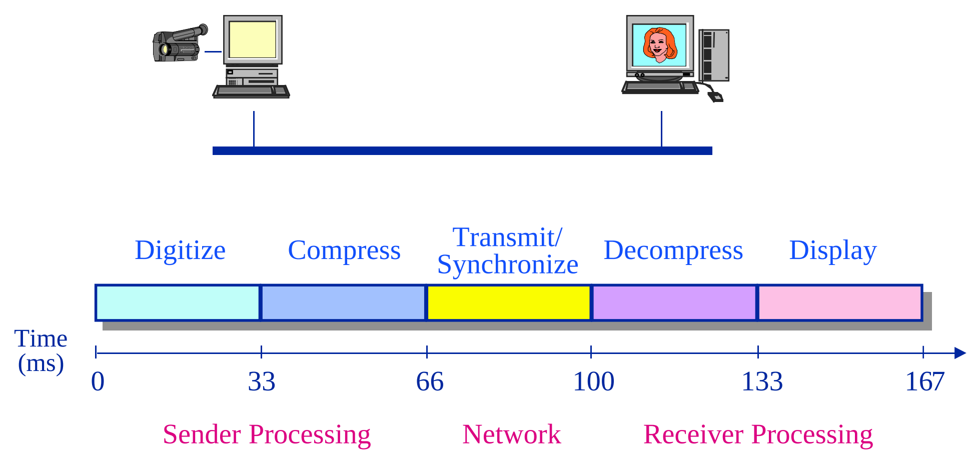
Recevier synchronization



- In general, acquisition and playout clocks are not synchronized
- Therefore a buffer must be present at the receiver to adjust for phase-shift in sender's & receiver's media clocks

Latency in Computer-Based Video Systems

Best case end-to-end latency

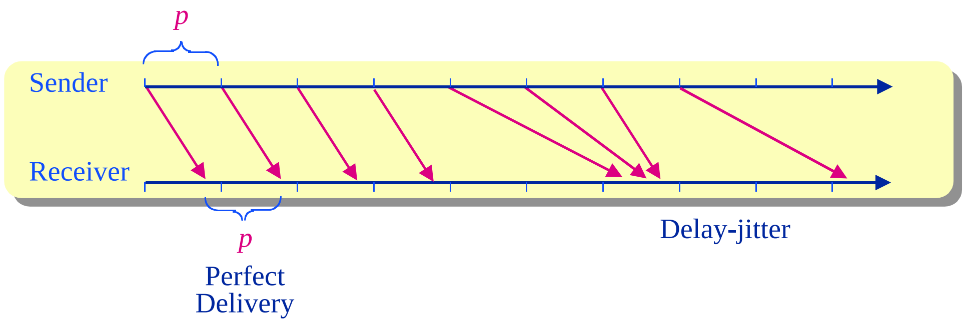


Latency in Computer-Based Audio Systems
How bad can audio latency be?

- Just as bad as video if lip-synchronization is required
- Otherwise, it depends on how one manages the network interface
  - » Video frames are typically too large to fit into a single network packet
  - » Multiple audio samples can be transmitted together
- Example: An audio codec generating 1 byte of data every 125 μs
  - » Building 500 byte packets requires 62.5 ms/packet
  - » Building 1,500 byte packets requires 187.5 ms/packet

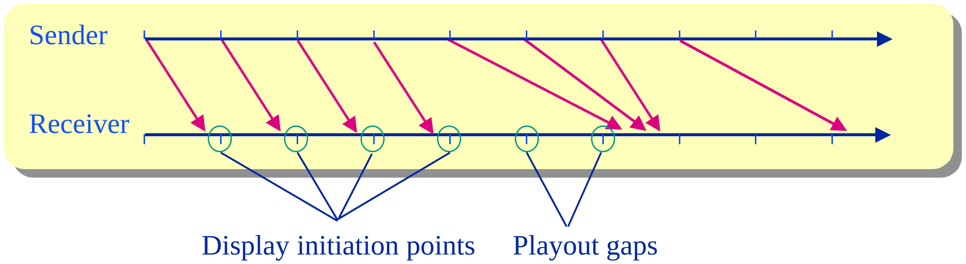
Performance Requirements
Delay-jitter

- Latency
  - » 250 ms one-way
- Delay-jitter — Variation in end-to-end latency



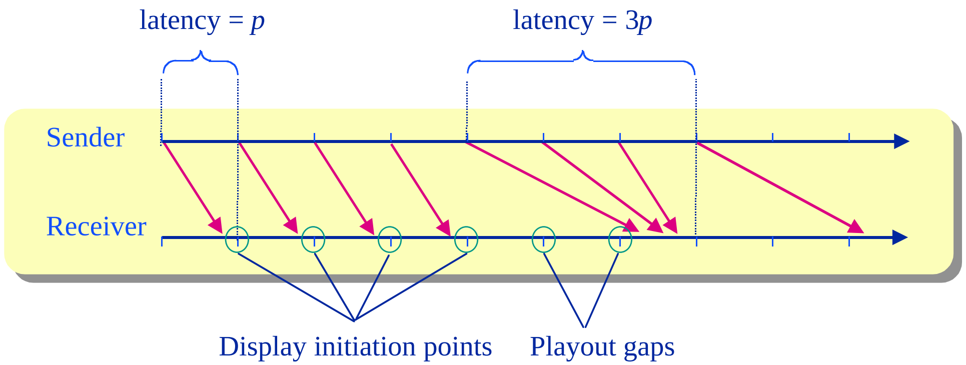
Performance requirements
The impact of delay-jitter

- Delay-jitter leads to “gaps” in the playout of media and increases playout latency



Performance requirements
The impact of delay-jitter

- Delay-jitter increases playout latency

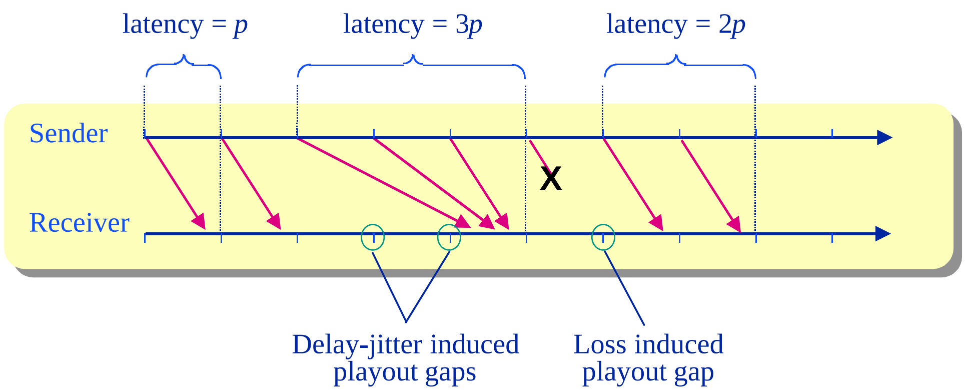


Performance Requirements

- Latency
  - » 250 ms one-way
- Delay-jitter
- Throughput —the effective delivered frame or sample rate
  - » For video the issue is *motion perception*
  - » For audio the issue is comprehension
- Loss — the complement of throughput

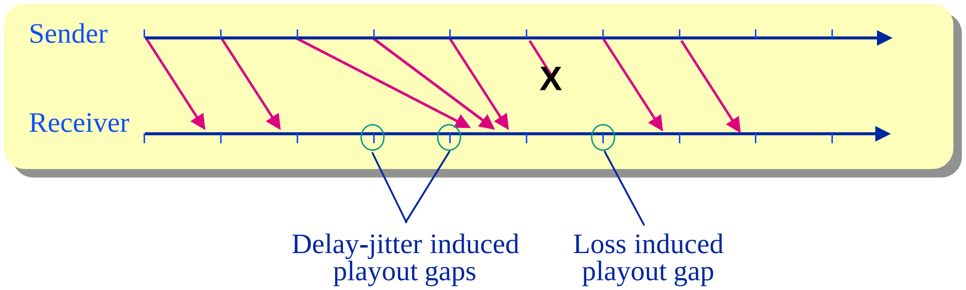
Performance requirements
Loss

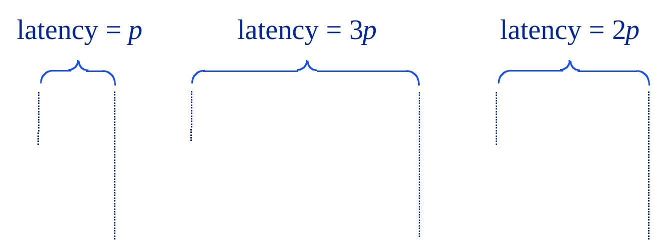
- Loss has the same effect as delay-jitter: *gaps*
  - » With a potentially beneficial effect of potentially lower latency



Performance requirements
Loss

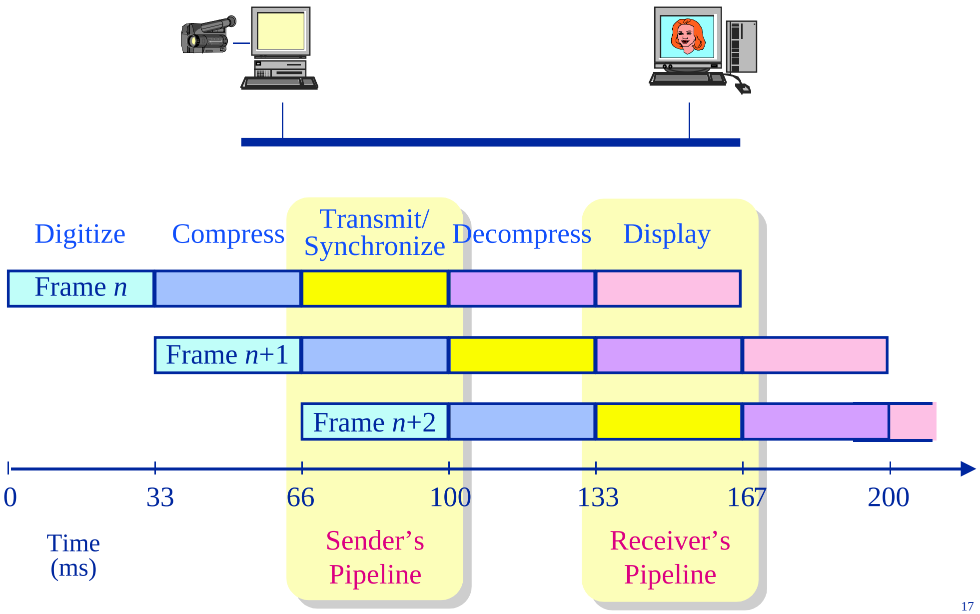
- Loss has the same effect as delay-jitter: *gaps*
  - » With a potentially beneficial effect of potentially lower latency





Avoiding Loss in the End System

Real-time management of a processing pipeline



Performance requirements

Loss requirements

- Audio — 1-2% sample loss
  - » individual sample losses (depending on sample size) are noticeable
  - » 5-10 lost samples per minute are tolerable (the distribution of loss is critical)
- Video — 10-15 frames/s required for minimal motion perception
  - » highly application dependent
  - » video loss raise issues of "network citizenship"

Performance Requirements

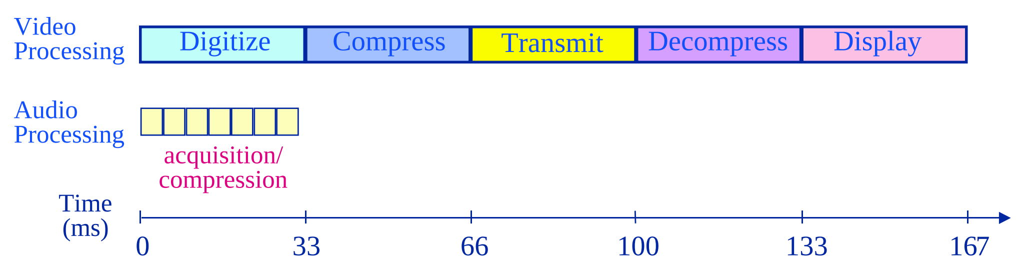
- Latency
  - » 250 ms one-way
- Delay-jitter
- Throughput —the effective delivered frame or sample rate
  - » For video the issue is *motion perception*
  - » For audio the issue is comprehension
- Loss

- Lip synchronization
  - » The temporal relationship between an audio and video stream representing a human speaking

Performance Requirements

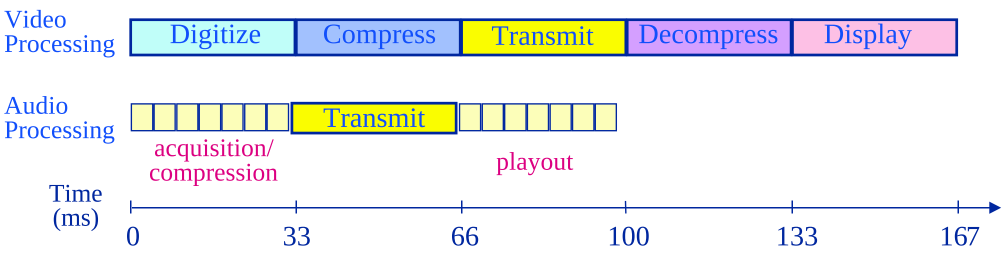
Lip synchronization

- Perfect lip synchronization requires audio playout at time _____



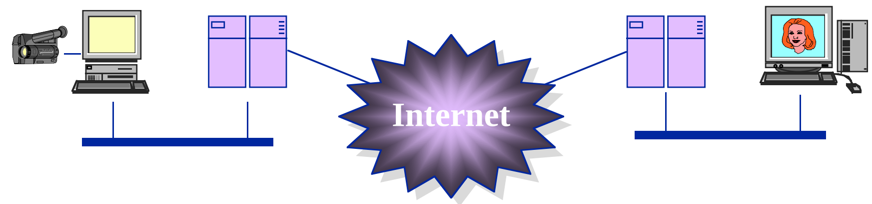
Performance Requirements
Lip synchronization

- Varying lip sync can be an effective technique in mitigating high video latency



- But... this is fundamentally unnatural!

Interactive Multimedia Applications
Performance requirements



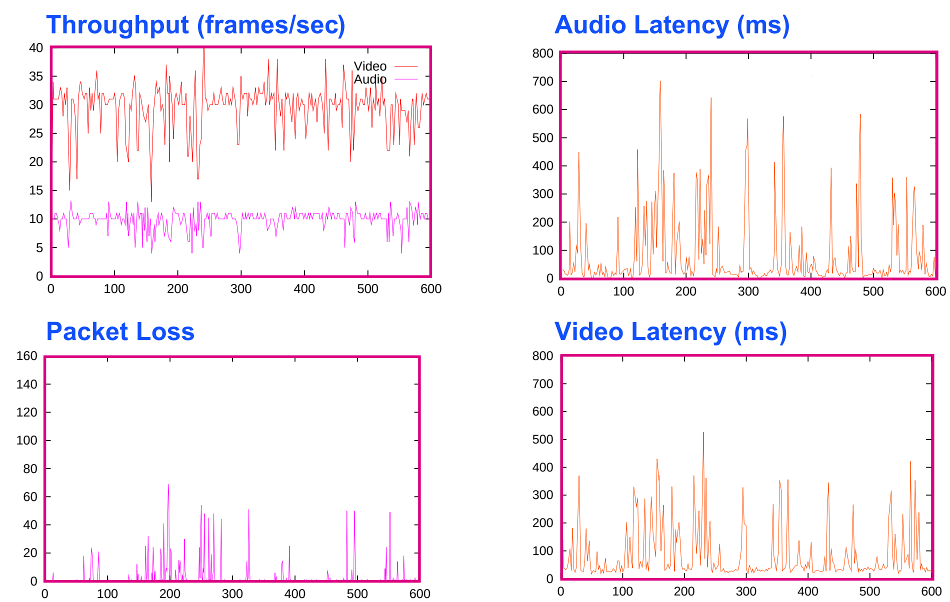
- No more than 250 ms end-to-end, one-way latency
- Continuous audio
- Minimum of 10 frames per second video throughput
- “Loosely synchronized” playout — ± 80 ms skew

Issues in Multimedia Delivery on Today’s Internet

Outline

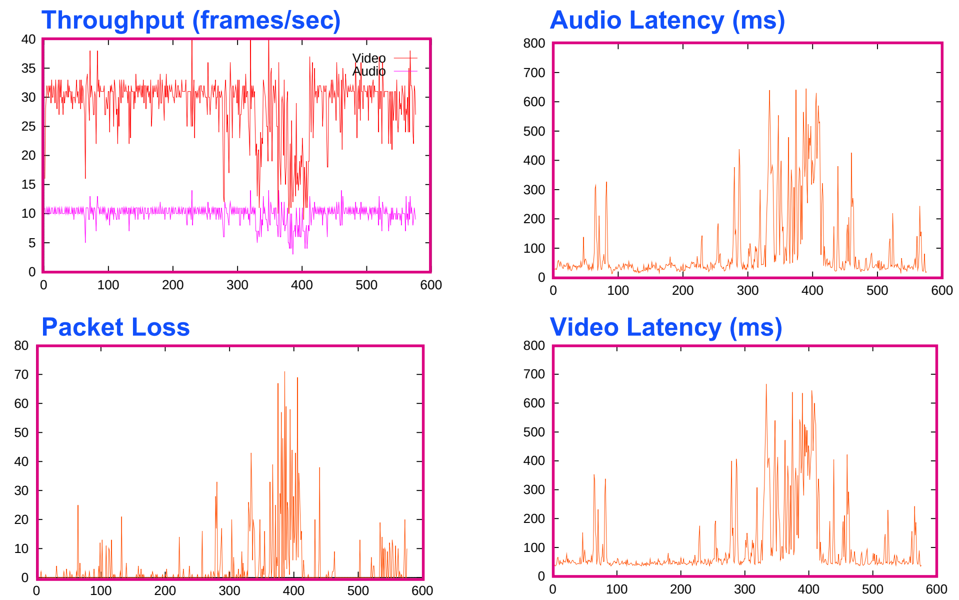
- Domain of discourse
  » Definitions, concepts, and objectives
- Performance of “naive” applications today
  » What’s “broken”?
- Media adaptations for best-effort multimedia delivery
  » Can we fix “it”?
- Performance of best-effort applications today
  » Fundamental challenges for tomorrow’s Internet

Videoconferencing on the Internet Today
ProShare™ performance on the Internet



Videoconferencing on the Internet Today

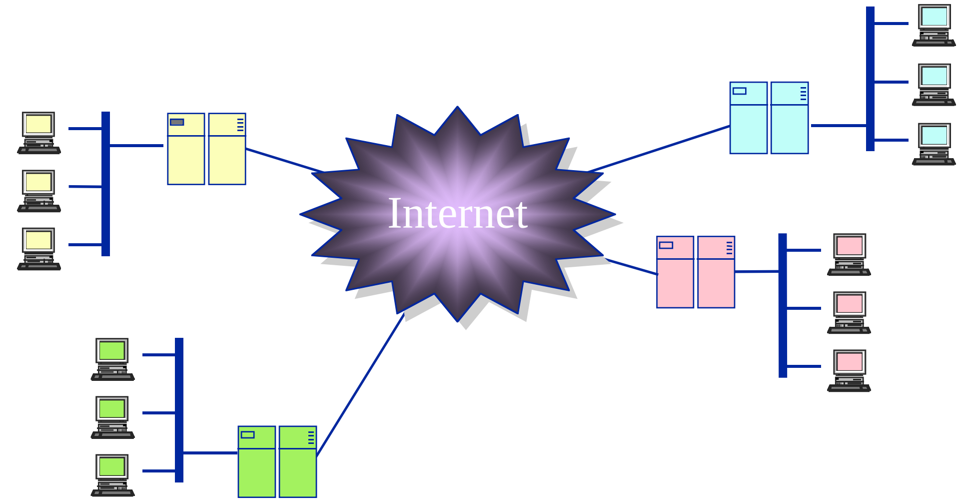
ProShare™ performance on the Internet



Videoconferencing on the Internet Today

What's the problem?

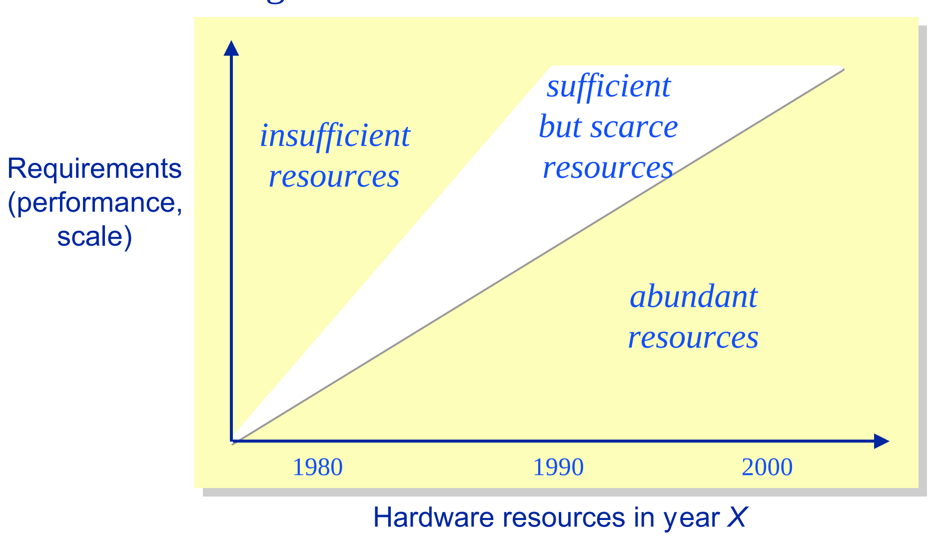
- Where is data being delayed and lost?



Videoconferencing on the Internet Today

What's the problem?

- Do we need more bandwidth or just better management of the existing bandwidth?



Where do we go from here?

Two fundamental approaches

- Provide true quality-of-service through reservation of resources in the network
  - » Requires coordination amongst all parties
    - admission control
    - policing
    - ...
- Provide "best-effort" service by adapting media streams
  - » Monitor & provide feedback on performance
  - » Bias transmission and processing of media to ameliorate the effects of congestion

Issues in Multimedia Delivery on Today's Internet

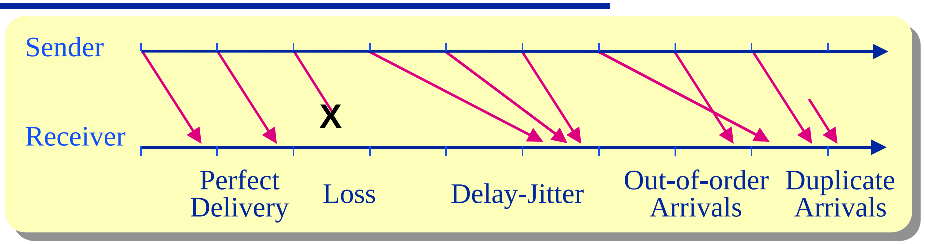
Outline

- Domain of discourse
  - » Definitions, concepts, and objectives
- Performance of "naive" applications today
  - » What's "broken"?
- Media adaptations for best-effort multimedia delivery
  - » Can we fix "it"?
- Performance of best-effort applications today
  - » Fundamental challenges for tomorrow's Internet

Best-Effort Multimedia Networking
Outline

- IP message delivery semantics
  - » The four common Internet pathologies
- Ameliorating the effects of delay-jitter
  - » "60 ways to queue & play your media samples"
- Ameliorating the effects of packet loss
  - » Recovery of lost samples through retransmission
  - » Recovery of lost samples through the addition of redundant information
- Congestion control
  - » Adaptive media scaling and packaging

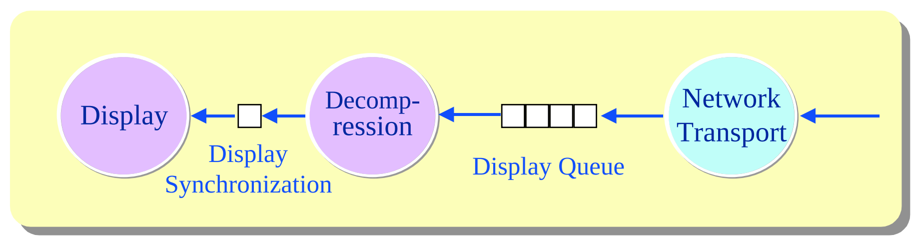
Best-Effort Multimedia Networking
The four Internet pathologies



- Delay-jitter
  - » Managing a trade-off between end-to-end latency and continuous playout
- Loss
  - » Proactively control through forward error correction
  - » Reactively control through retransmission
- Out-of-order arrivals
  - » Assume out-of-order sample is lost
  - » Assume sample is late
- Duplicate arrivals
  - » Do we care?

Ameliorating the Effects of Delay-Jitter
Trading-off end-to-end latency for continuous playout

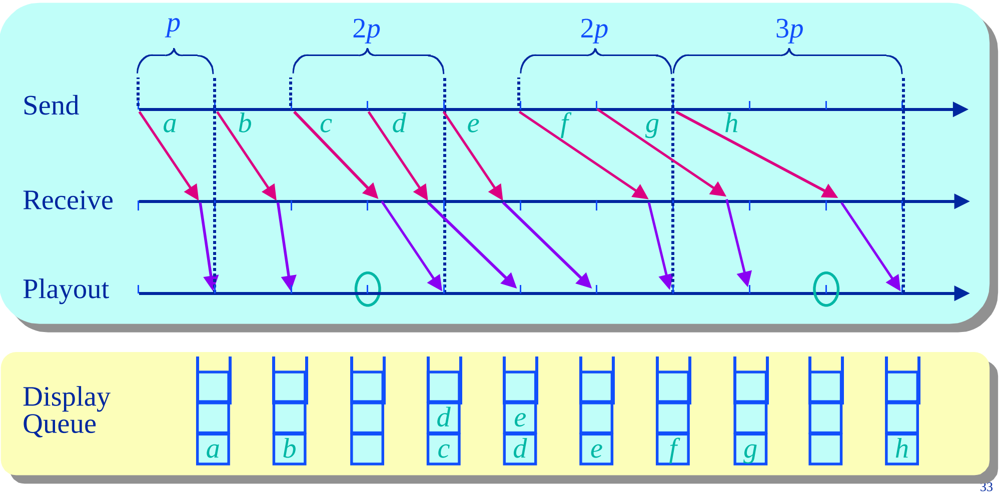
- When the first media sample arrives, should it be played or enqueued?



Receiver's processing pipeline

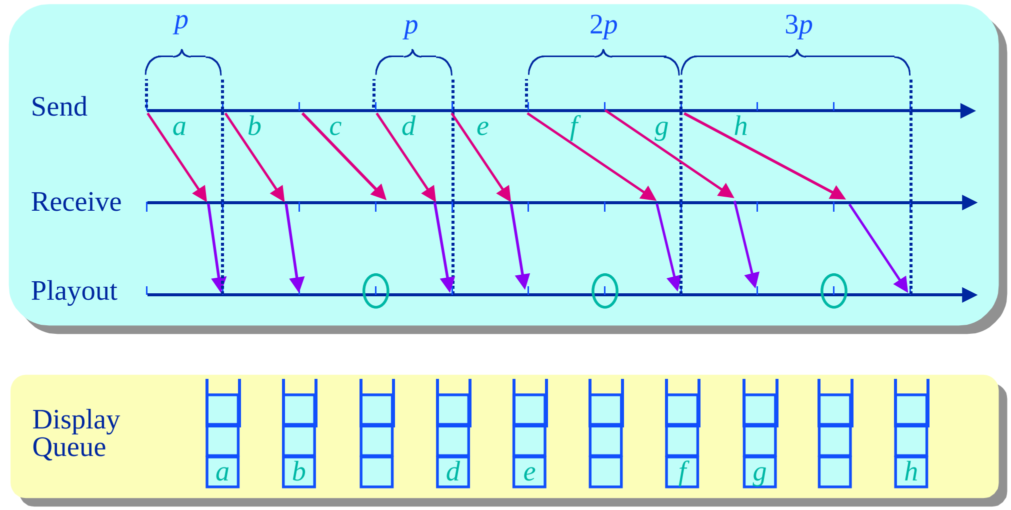
Ameliorating the Effects of Delay-Jitter
Trading-off end-to-end latency for continuous playout

- When the first media sample arrives, should it be played or enqueued?
  - » playing the sample ensures minimal end-to-end latency...



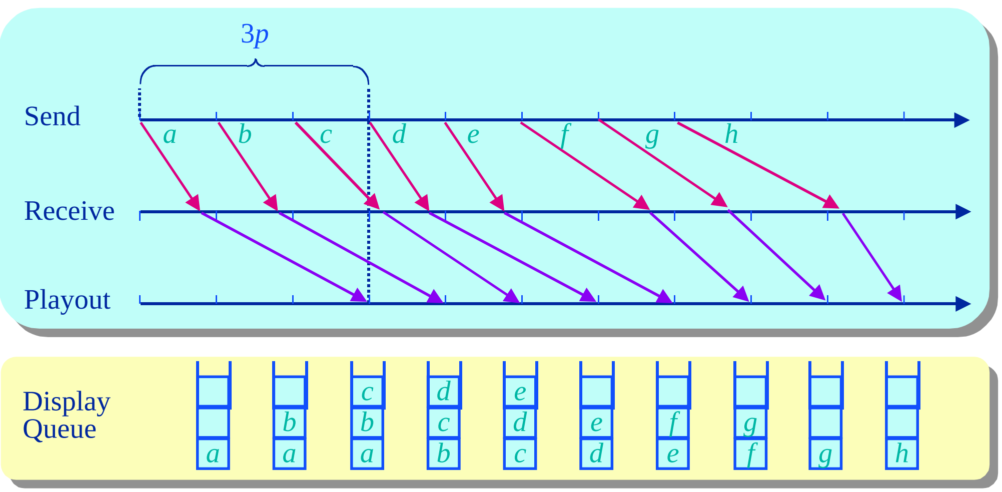
Ameliorating the Effects of Delay-Jitter
Trading-off end-to-end latency for continuous playout

- Samples that arrive "too late" may be discarded



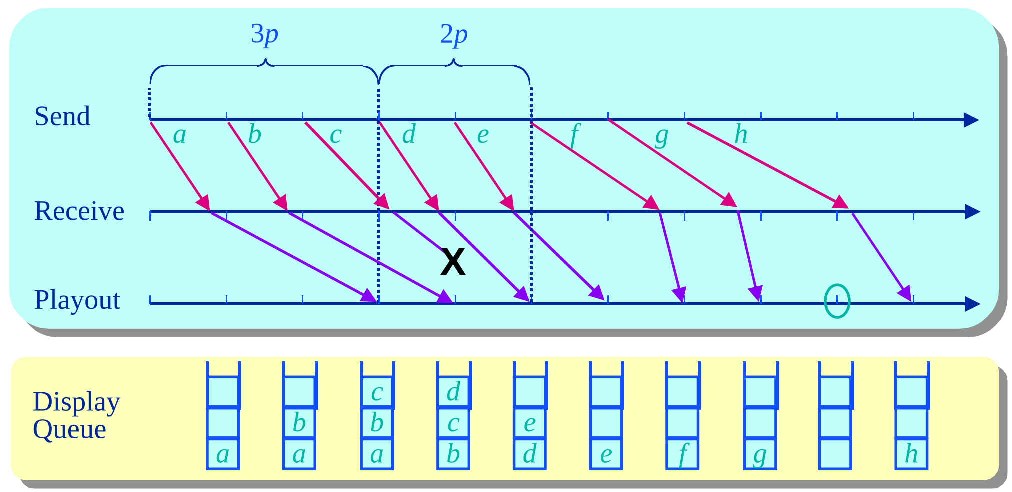
Ameliorating the Effects of Delay-Jitter
Trading-off end-to-end latency for continuous playout

- Enqueueing the sample ensures continuous playout...



Principles of Delay-Jitter Buffering
Sample discarding

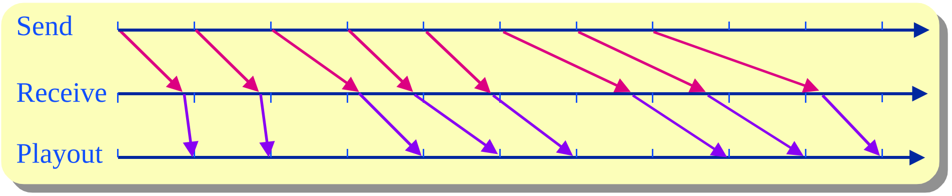
- Purposefully throwing away samples reduces latency



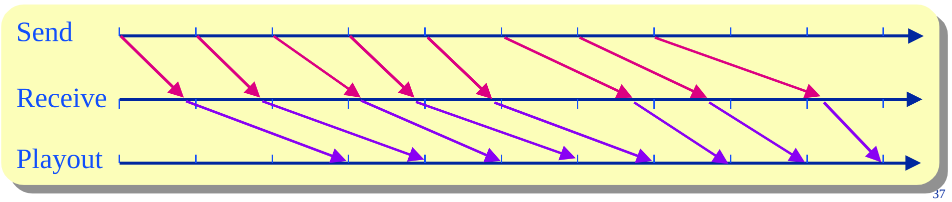
Principles of Delay-Jitter Buffering

Two fundamental initial playout strategies

- Let naturally occurring network delays determine playout latency



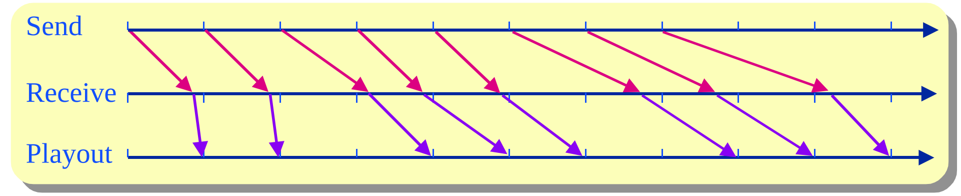
- Avoid initial sequence of playout gaps by estimating network delay and setting playout delay accordingly



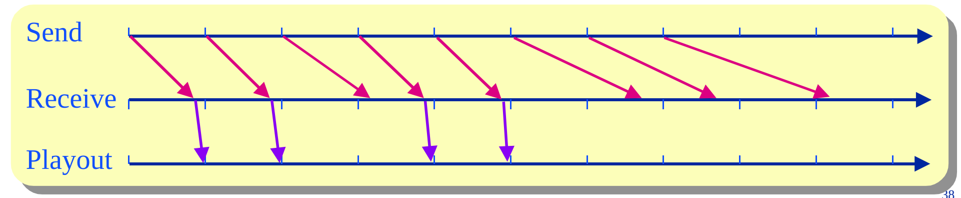
Principles of Delay-Jitter Buffering

Two fundamental late arrival strategies

- Play media samples as they arrive
  - » Latency increases as delay-jitter increases



- Discard "late" samples
  - » Playout media with constant latency

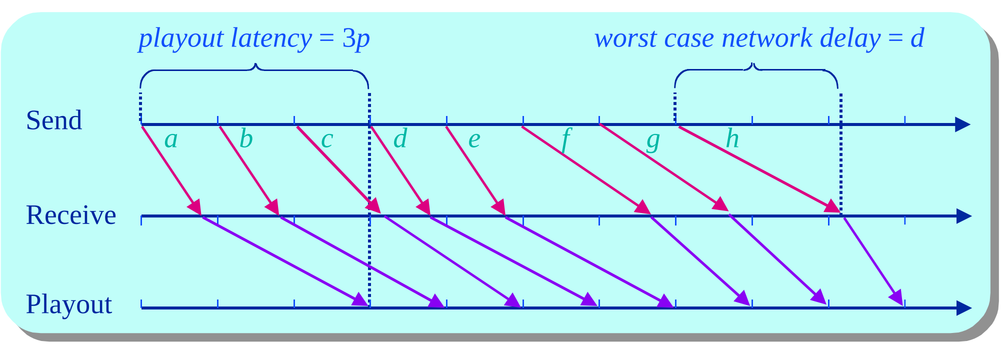


Principles of Delay-Jitter Buffering

Estimating network delay

- If network delay is bounded by a constant *d*:
  - » timestamp each packet at sender
  - » when a packet arrives arrives, enqueue the packet and dequeue at time:

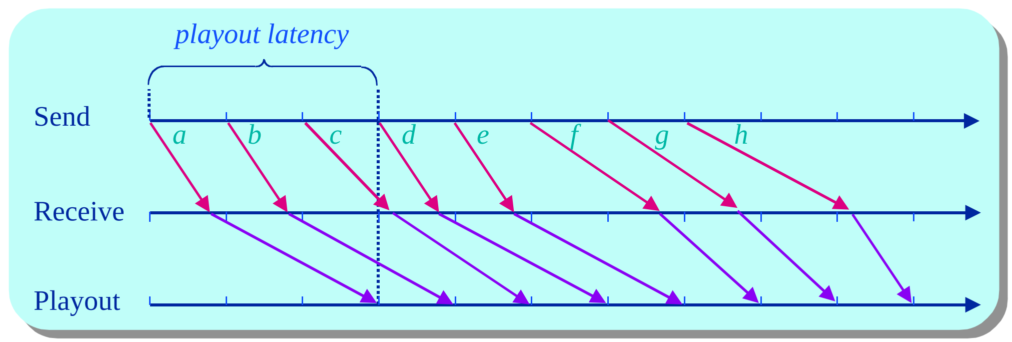
    sender's_transmission_time + *d*



Principles of Delay-Jitter Buffering

Estimating network delay

- Basic algorithm: *playout latency* = *d* + (*k* x *v*) where
  - » *d* is the average estimated network delay
  - » *v* is the estimated variation deviation
  - » *k* is a "congestion estimator"



Principles of Delay-Jitter Buffering
Estimating network delay

$$\boxed{\textit{playout latency} = d + (k \times v)}$$

- The average network delay and variation can be estimated by:

$$d_{new\ estimate} = d_{old\ estimate} + \alpha \times (d_{observed} - d_{old\ estimate})$$
$$v_{new\ estimate} = v_{old\ estimate} + \beta \times (|d_{observed} - d_{old\ estimate}| - v_{old\ estimate})$$

or

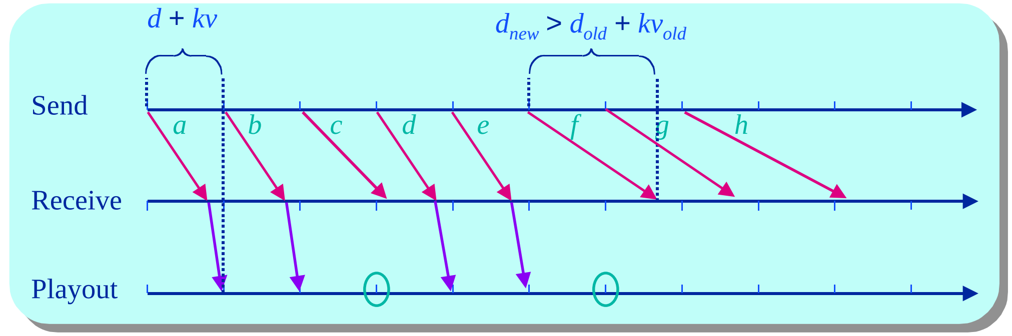
$$d_{new\ estimate} = \alpha\, d_{old\ estimate} + (1 - \alpha)\, d_{observed}$$
$$v_{new\ estimate} = \beta\, v_{old\ estimate} + (1 - \beta) \times (|d_{observed} - d_{old\ estimate}|)$$

or

$$d_{new\ estimate} = \mathrm{MIN}(d_{observed}\ \textit{in the recent past})$$

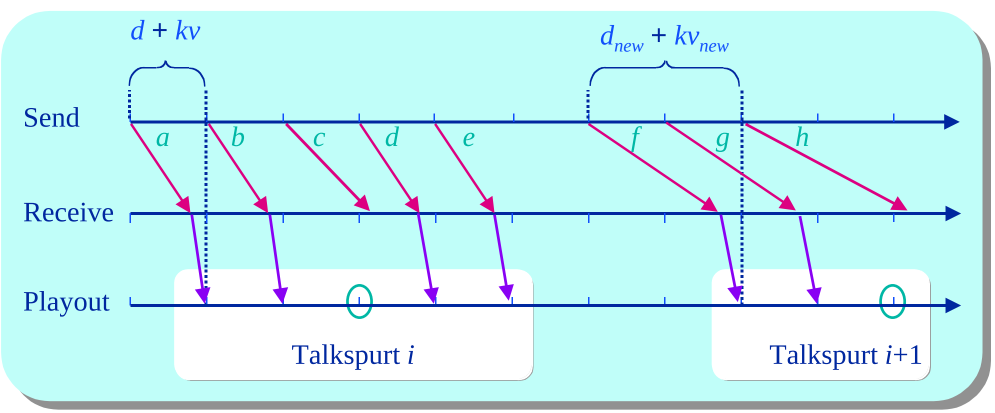
Principles of Delay-Jitter Buffering
Estimating network delay

- All samples are scheduled for playout at time
  playout latency = $d + (k \times v)$
- But when should playout latency be changed?



Principles of Delay-Jitter Buffering
Voice transmission

- For voice transmission we can dynamically adapt playout times of audio samples using *silent periods* to "resync" the stream

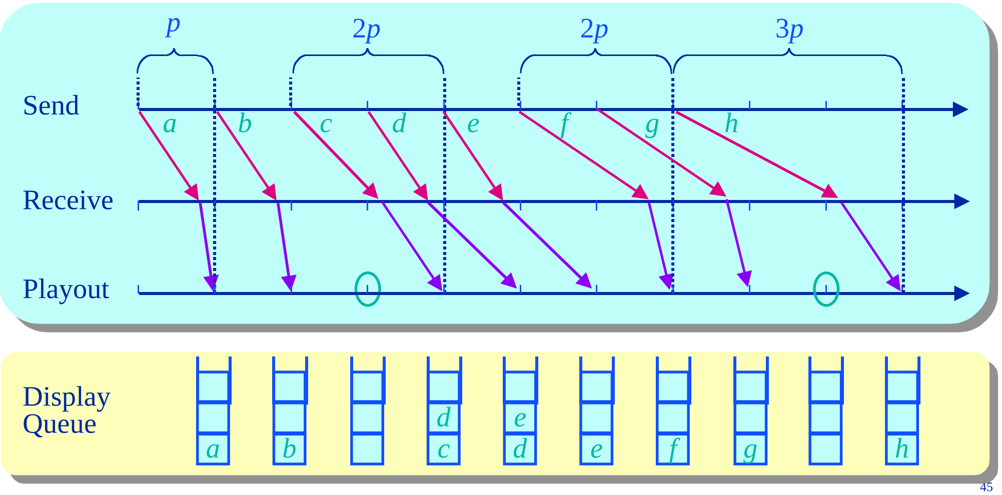


Principles of Delay-Jitter Buffering
Continuous audio transmission

- Many forms of audio (and other media) must be transmitted continuously
  - » Music
  - » "Noisy" voice
  - » Mixed audio streams
  - » Video?
- Scheduling individual samples for playout based on estimates of network delay gives poor results

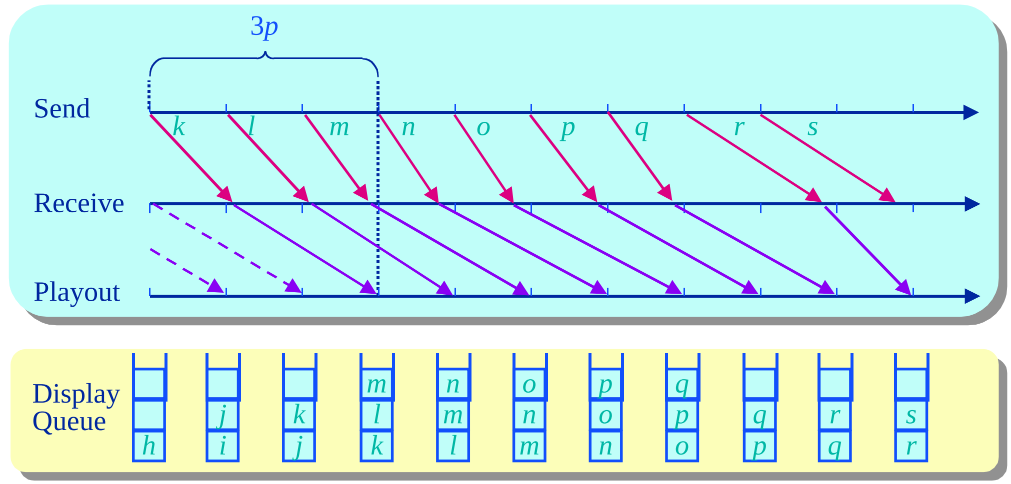
Principles of Delay-Jitter Buffering
Continuous audio transmission

- Let naturally occurring network delays determine playout latency



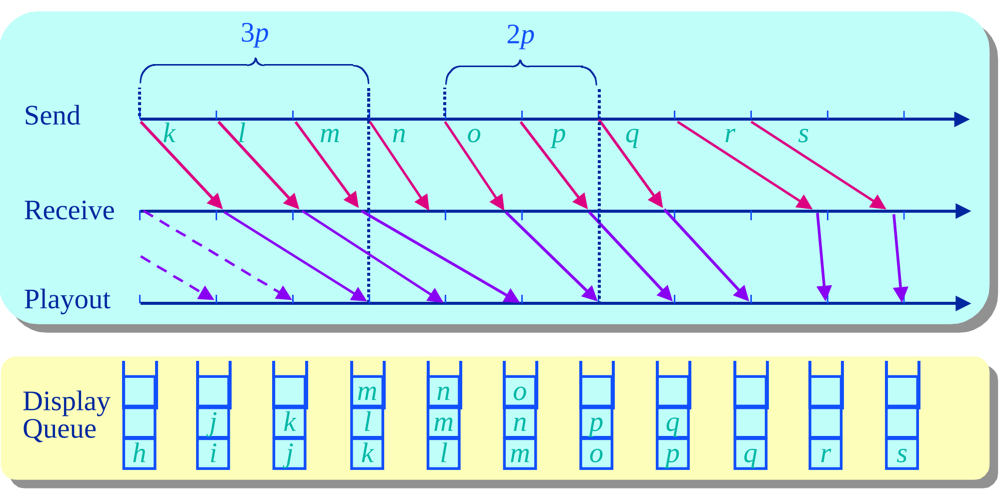
Principles of Delay-Jitter Buffering
Continuous audio transmission

- How do we determine if our delay-jitter buffer is too large?



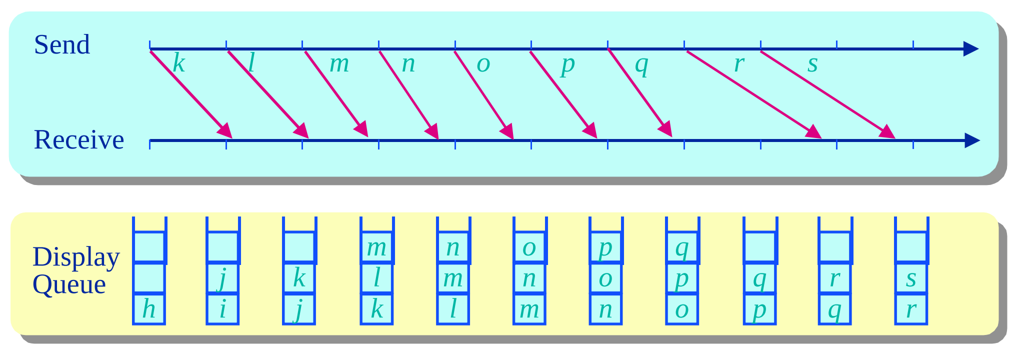
Principles of Delay-Jitter Buffering
Continuous audio transmission

- Simulating packet loss by discarding samples at the receiver will reduce playout latency

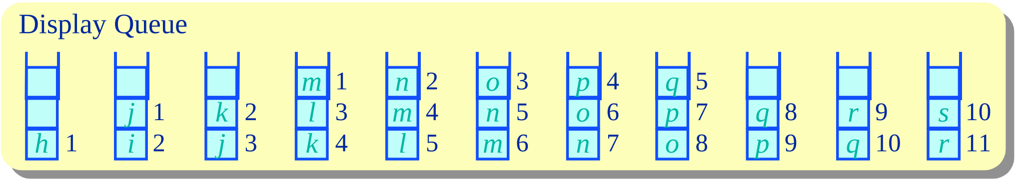


Continuous Audio Transmission
Queue monitoring

- Rather than compute network delay, infer it from the length of the display queue
  - » If queue length grows, network delay is decreasing
  - » If queue length shrinks, network delay is increasing
  - » If queue length remains constant, network delay is stable



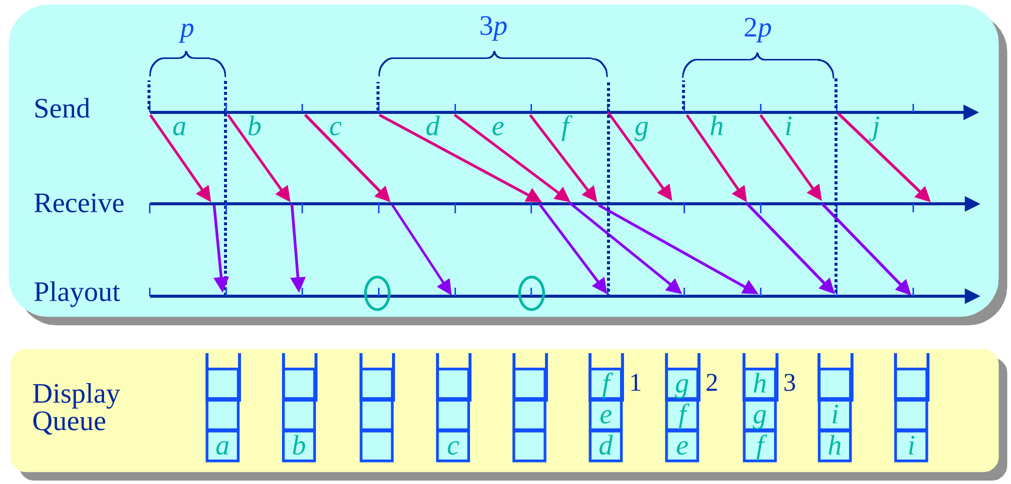
Continuous Audio Transmission
Queue monitoring



- Keep count of the number of consecutive display initiation times at which the display queue contained *n* items
- When the count exceeds a threshold, the oldest sample in the queue is discarded
  - » queue locations near the head of the queue have large thresholds
  - » queue locations near the tail of the queue have small thresholds

Continuous Audio Transmission
Queue monitoring

- Example: Queue monitoring with thresholds = 3, 10
  - » sample *g* discarded at playout time 10



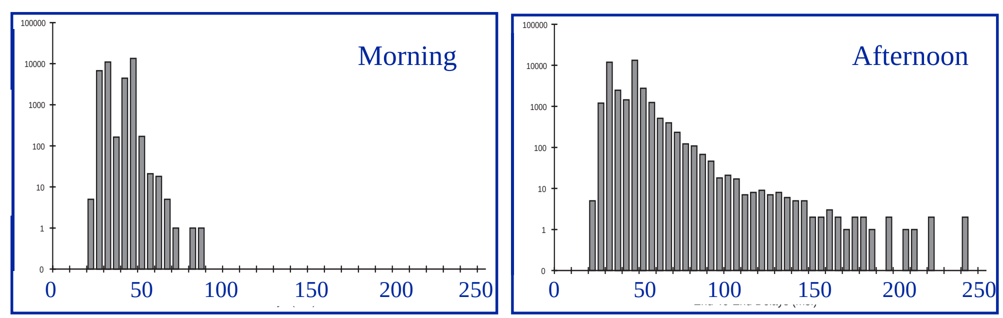
Queue Monitoring
Performance on a campus-area network

- How much delay-jitter can be accommodated in practice?
  - » What ranges of delay-jitter are observed?
  - » How well do these buffing schemes work in practice?
- Stone's delay-jitter study in the UNC CS department:
  - » A comparison of the effectiveness of three delay-jittter management policies:
    - I-Policy — playout media with fixed latency
    - E-Policy— playout media samples as they arrive
    - Queue Monitoring — adaptively set the playout delay

    on the playout of audio/video in a videoconferencing system

Queue Monitoring
Performance on a campus-area network

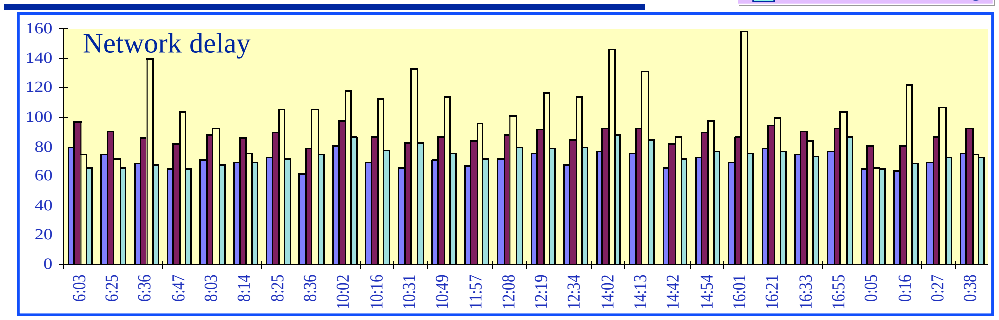
- What ranges of delay-jitter are observed?
  - » Stone measured the performance of 28, 5 minute conferences during the course of a "typical" day

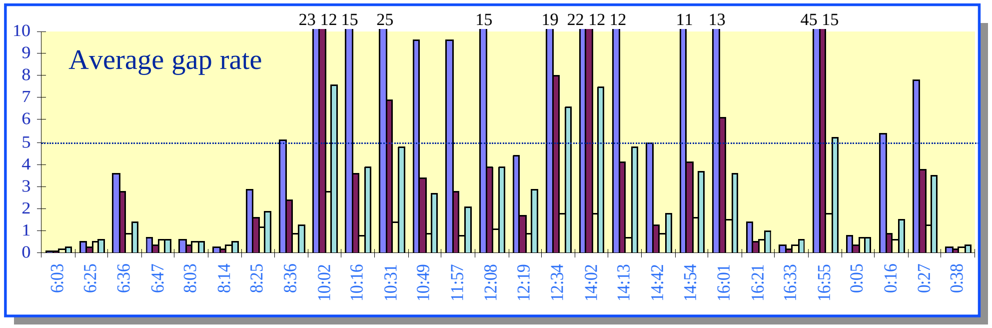


Audio end-to-end delay distribution (in ms)

Queue Monitoring

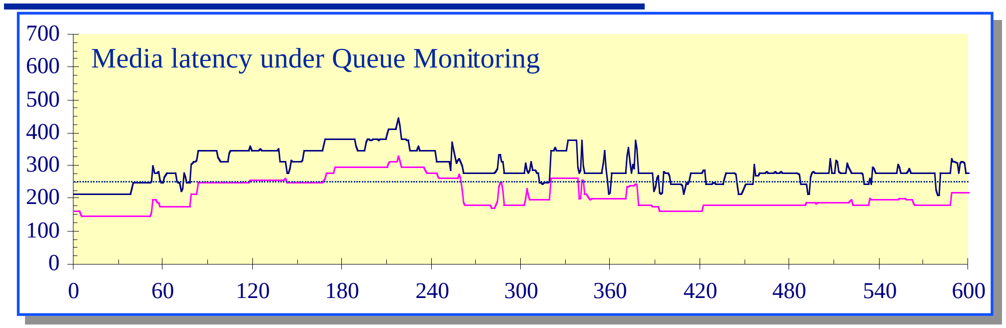
Performance on a campus-area network

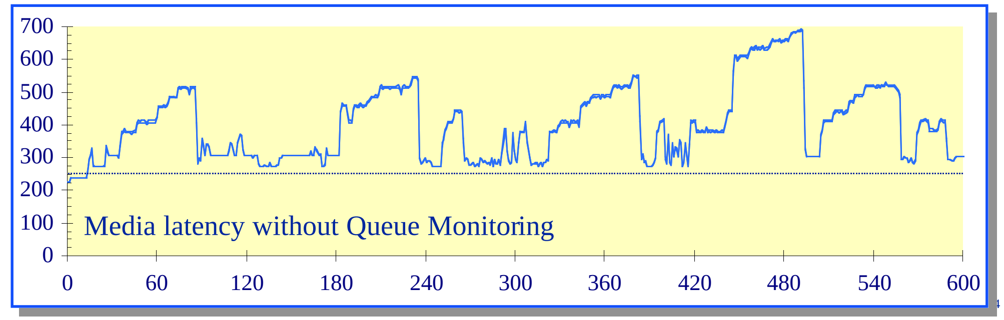




Queue Monitoring

Performance on a campus-area network





Principles of Delay-Jitter Buffering

Non-real-time media transmission

- If the communication is non-real-time, doesn’t simple static buffering solve the problem?
  - » Yes, but...

Adaptive, best-effort, multimedia networking

Outline

- IP message delivery semantics
  - » The four common Internet pathologies
- Ameliorating the effects of delay-jitter
  - » “60 ways to queue & play your media samples”
- Ameliorating the effects of packet loss
  - » Recovery of lost samples through retransmission
  - » Recovery of lost samples through the addition of redundant information
- Congestion control
  - » Adaptive media scaling and packaging

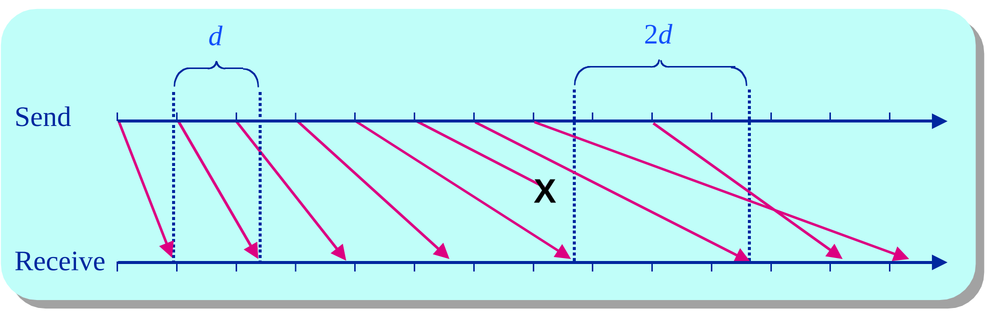
Dealing With Packet Loss
Application requirements

- Audio — 1-2% sample loss
  - » individual sample losses are noticeable (depending on the sample size)
  - » 5-10 lost samples per minute are tolerable (the distribution of loss is critical)
- Video — 10-15 frames/s required for minimal motion perception
  - » highly application dependent
  - » video loss raise issues of “network citizenship”

Dealing With Packet Loss
Two basic approaches

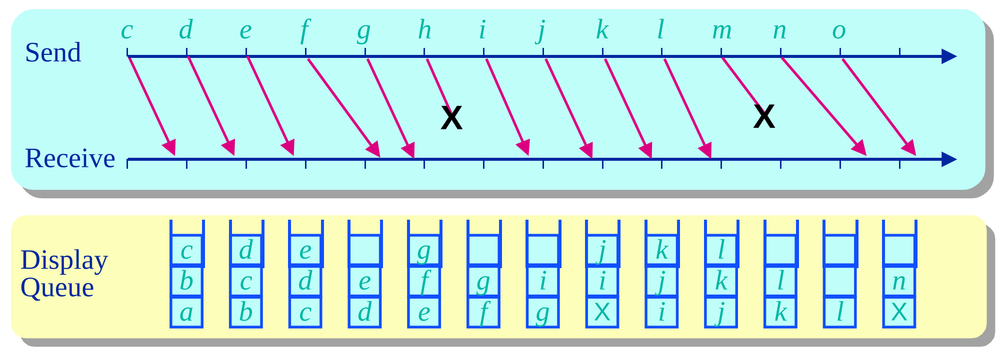
- Traditional “reactive” approach
  - » Acknowledge transmissions and resend lost packets
    - “Automatic Repeat Request” (ARQ)
- Two proactive approaches
  - » Introduce redundancy into streams to enable reconstruction of lost media samples
    - “Forward error correction” (FEC)
  - » Dynamically adapt streams to the bandwidth perceived to be available at the current time
    - Media scaling & packaging

Retransmission-Based Error Correction
Conventional wisdom



- Retransmission is silly...
  - » By the time you realize something is lost, it’s too late to resend it
  - » Traditional sender-oriented retransmission techniques do not scale to multicast environments

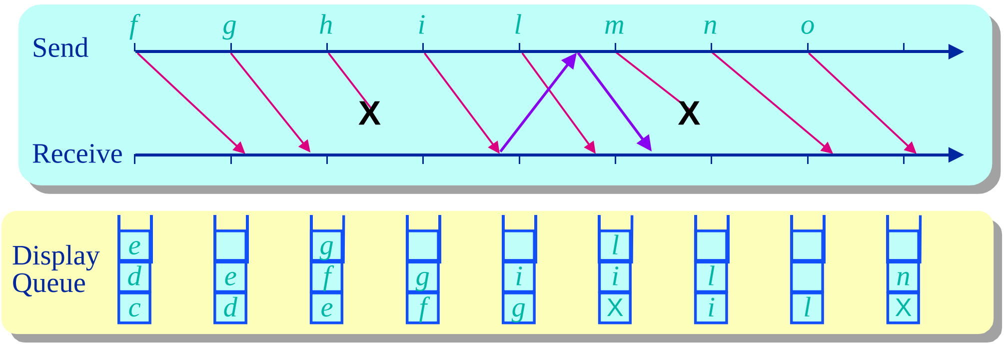
Retransmission-Based Error Correction
The reality



- Retransmission is potentially beneficial...
  - » Since data is buffered at the receiver to ameliorate the effects of jitter, provide intermedia synchronization, *etc.*, retransmission may work!

Retransmission-Based Error Correction

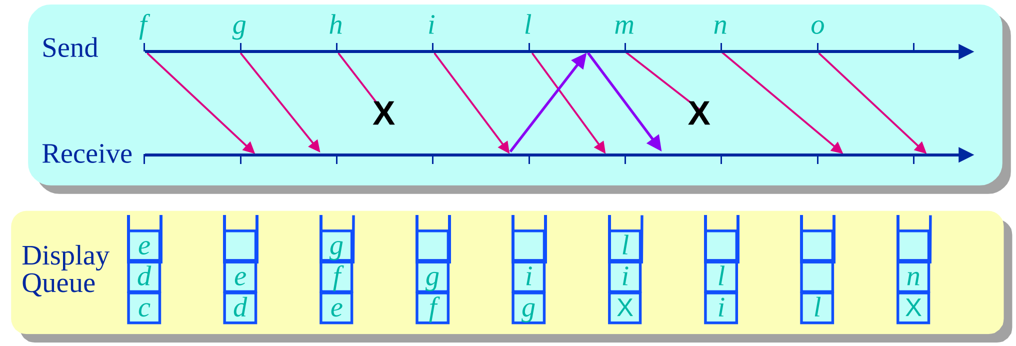
The retransmission process



1. Loss is detected
2. A retransmission request is issued
3. The requested packet is retransmitted

Retransmission-Based Error Correction

The retransmission "budget"



- If:

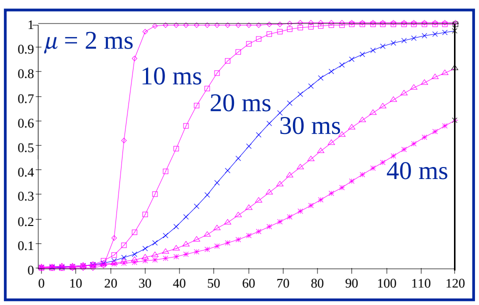
$$\textit{gap length} + (3 \times \textit{one-way transmission time}) < \textit{playout latency}$$

then retransmission is a possibility

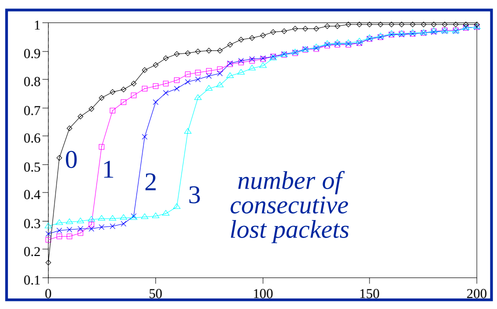
Is there likely to be enough time to retransmit?

The Dempsey *et al.* study

Retransmission effectiveness for different average network delays



Retransmission effectiveness for different loss patterns

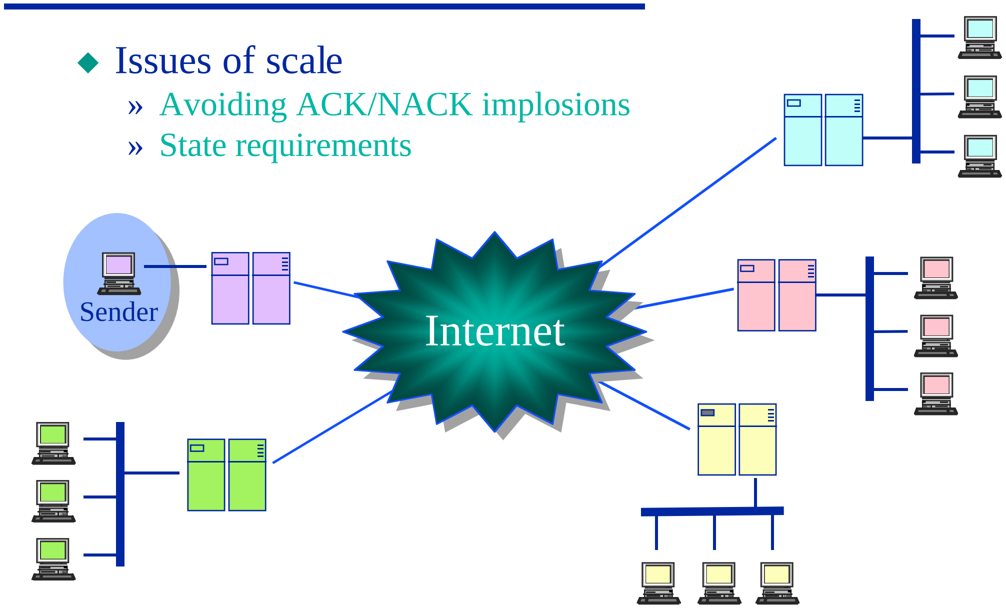


probability of continuous playout v. receiver buffering delay (in ms)

Retransmission-Based Error Correction

How can retransmission work in a multicast environment?

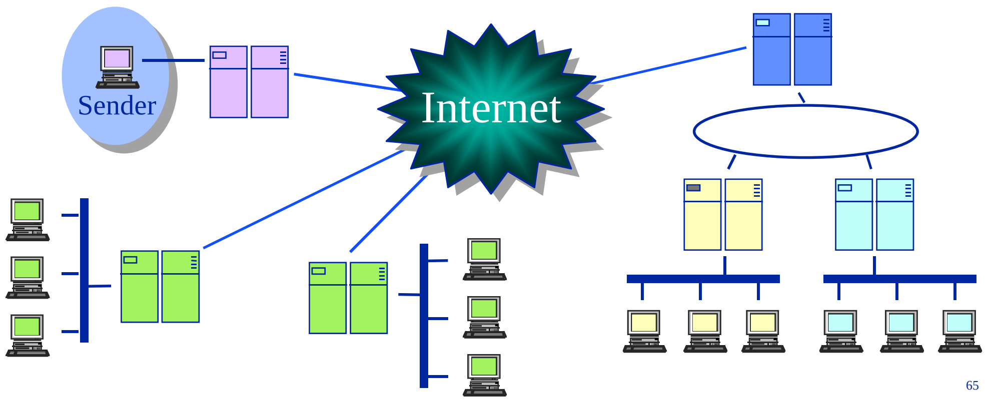
- Issues of scale
  - » Avoiding ACK/NACK implosions
  - » State requirements



Scalable Reliable Multicast

Principles of operation

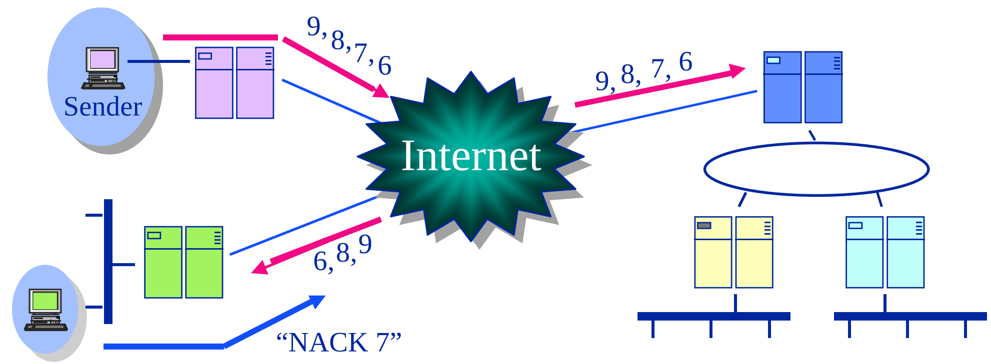
- Receivers are responsible for ensuring they receive the data they care about
  - » Repair requests are multicast to the group
- Any receiver is capable of acting as a sender and sending a repair response



Scalable Reliable Multicast

Avoiding repair and repair response implosions

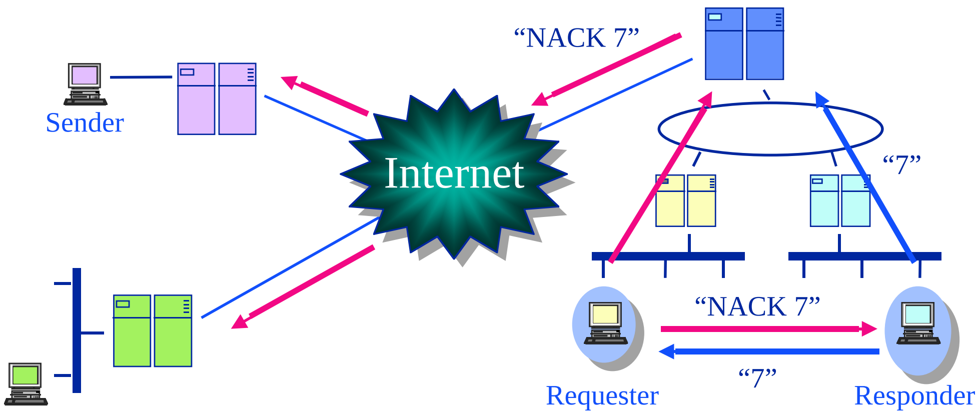
- Hosts continually measure the distance to each other
  - » Hosts periodically emit control messages as in RTCP
- When a receiver detects a loss, it sets a timer for emitting its repair request based on its estimate distance to the sender
  - » Send repair requests quickly to nearby senders



Scalable Reliable Multicast

Avoiding repair and repair response implosions

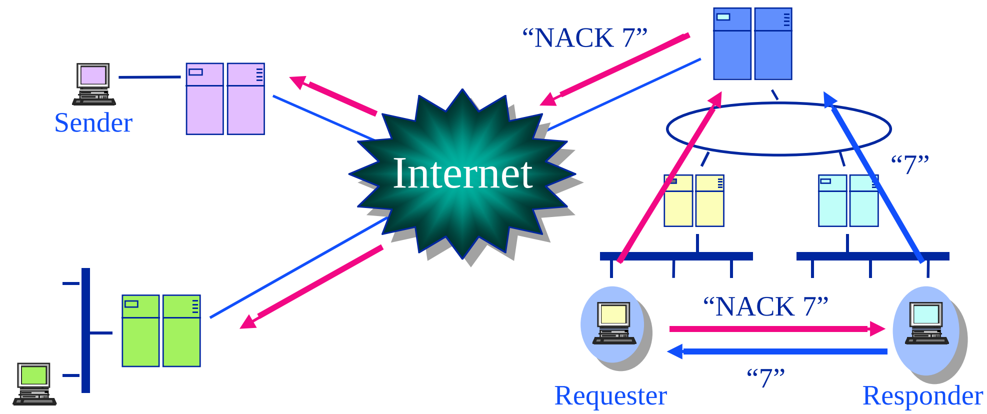
- If a host receives a repair request and it has the request packet, it similarly sets a timer for emitting its response based on its estimated distance to the receiver



Scalable Reliable Multicast

Avoiding repair and repair response implosions

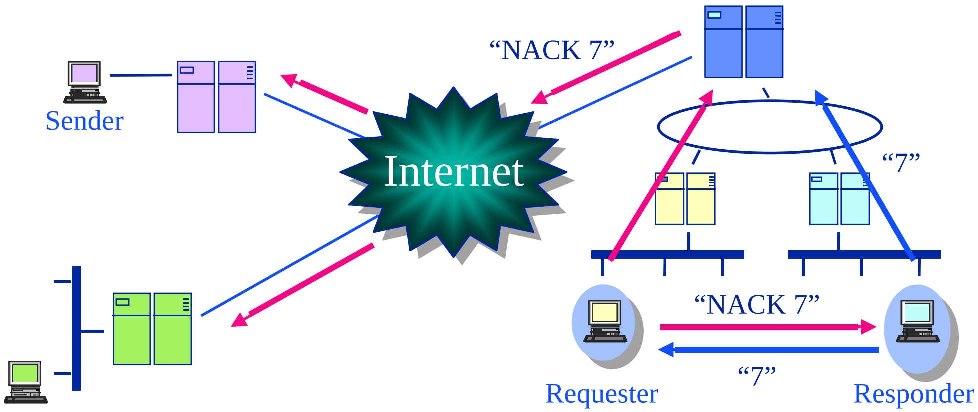
- Ideally a lost packet triggers only 1 repair request from a host just downstream from the point of failure & a single repair response from a host just upstream of the failure



Scalable Reliable Multicast

Performance issues

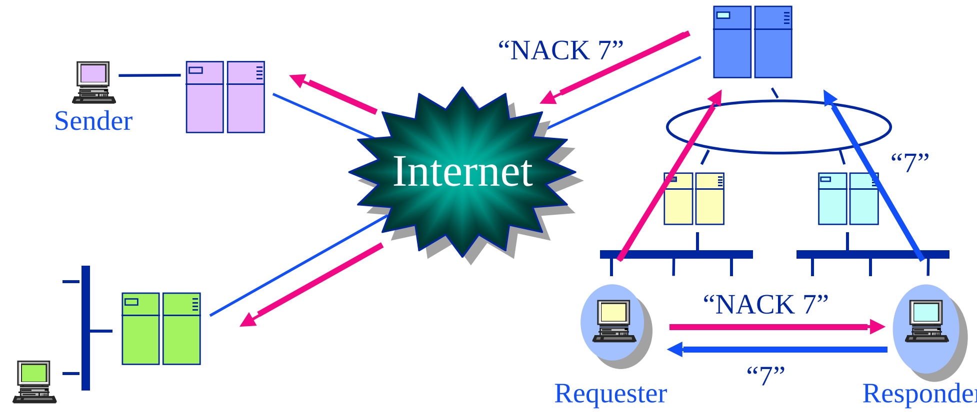
- If losses are infrequent and correlated, then few repair/response messages are sent
  - » But every host will receive each message
- Otherwise, in the worst case the data traffic can double



Scalable Reliable Multicast

Performance issues

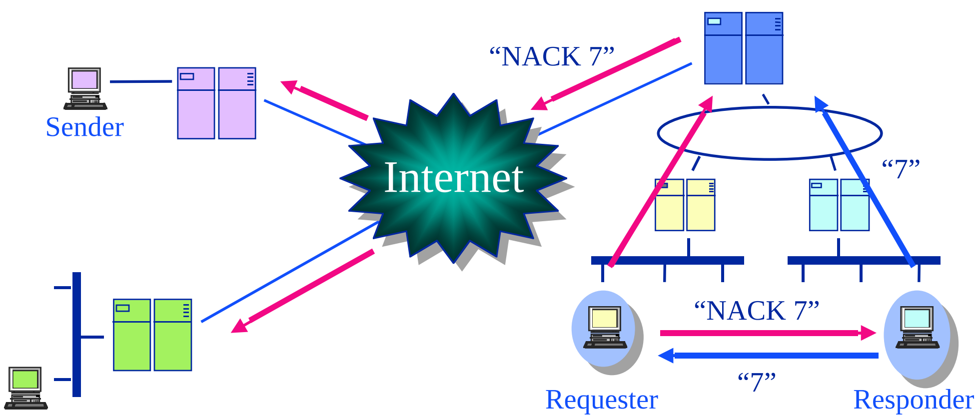
- What is the impact of having both the repair requester & responder delay before issuing their message?
  - » What is the likelihood that the resulting retransmission will be on time?



Scalable Reliable Multicast

Open issues

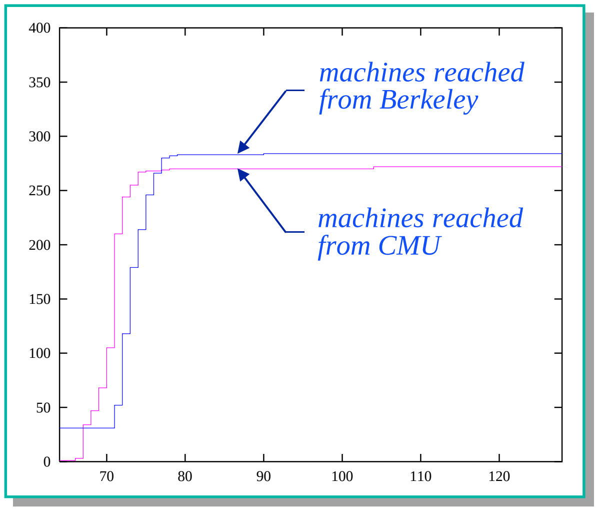
- How to limit the scope of repair/repair response messages?
- Managing the trade-off between keeping silent to avoid implosions and sending quickly to maximize (individual) performance



Scalable Reliable Multicast

Using TTL to limit the scope of repair/response messages

- TTL is not a good measure of locality
  - » Number of hosts reachable is not linear in TTL
- TTLs between two hosts are not symmetric

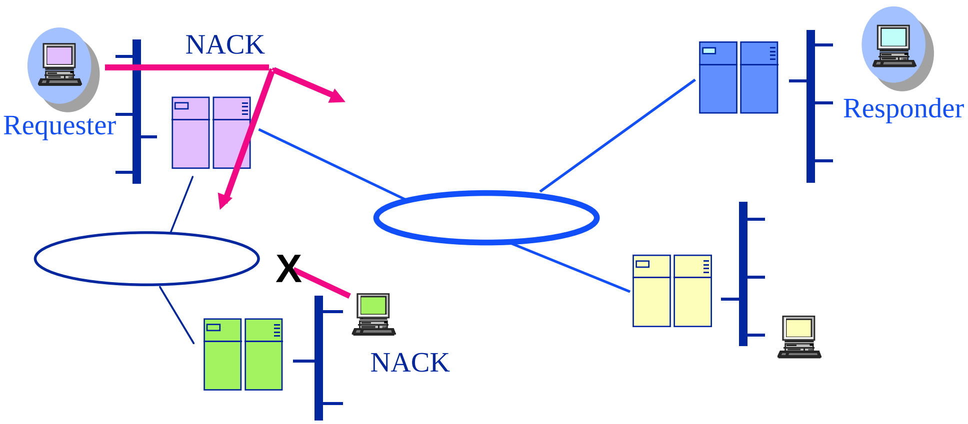


number of hosts reachable by a given TTL v TTL

Scalable Reliable Multicast

Using TTL to limit the scope of repair/response messages

- How can a repair responder ensure its reply reaches:
  - » the original requestor
  - » all would-be requestors who suppressed their repair request



Retransmission-Based Error Correction

Summary

- Retransmission will be effective means of dealing with packet loss if...
  - » we can detect losses quickly
  - » *average receiver buffering delay* ≥ (1.5 x RTT) + *gap length*
- Retransmission can be made to scale if...
  - » we can avoid repair request and response implosions
  - » repairs can be performed locally

Dealing With Packet Loss

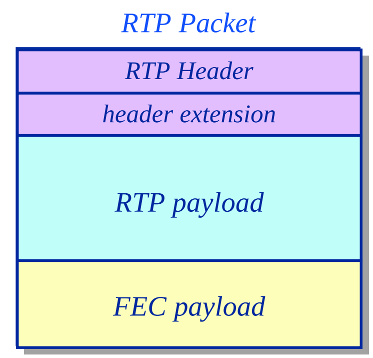
Two basic approaches

- Traditional “reactive” approach
  - » Acknowledge transmissions and resend lost packets
    - ❖ “Automatic Repeat Request” (ARQ)
- Two proactive approaches
  - » Introduce redundancy into streams to enable reconstruction of lost media samples
    - ❖ “Forward error correction” (FEC)
  - » Dynamically adapt streams to the bandwidth perceived to be available at the current time
    - ❖ Media scaling & packaging

Forward Error Correction

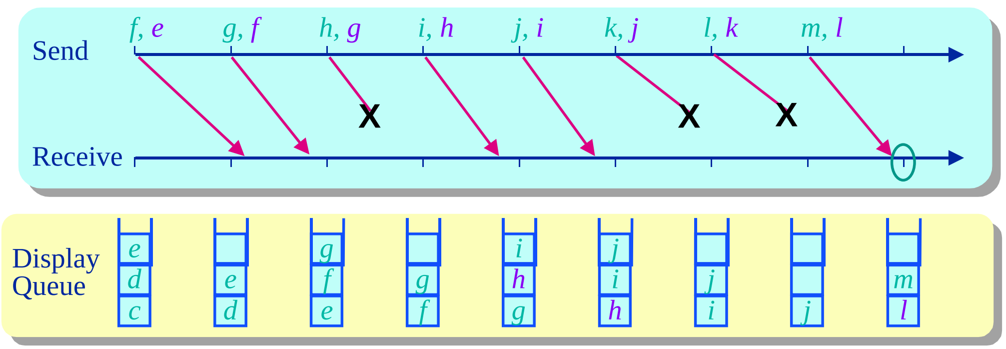
Basic concepts

- We introduce redundancy into the stream to enable the receiver to recover from errors due to loss
- Forms of redundancy
  - » Simple replication and re-transmission of original data
  - » *k*-way XOR
  - » Replication, recoding, and re-transmission of original data



Forward Error Correction

Simple replication and retransmission example

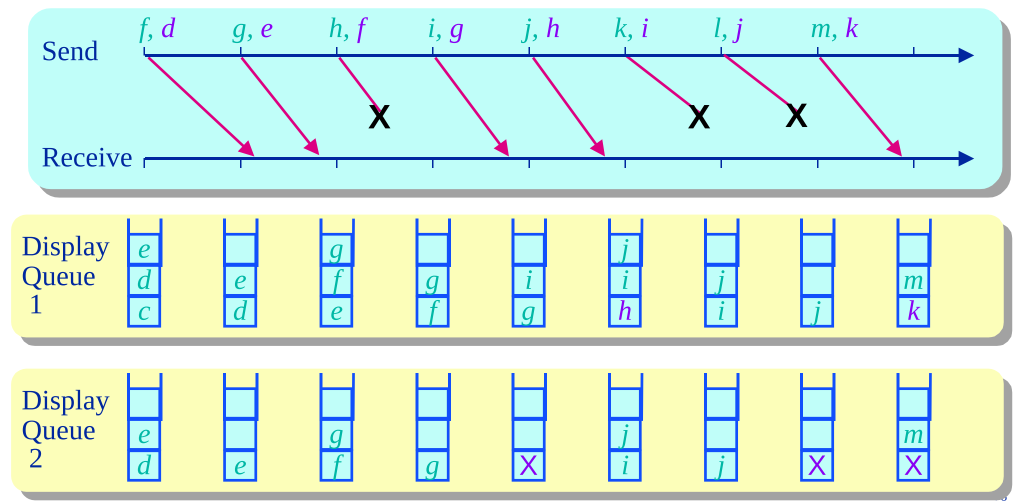


- Key issue: If a sample is lost, how do we ensure that the redundant information necessary for the repair arrives?
  - » How much bandwidth should we dedicate to FEC?
  - » Where should we place the redundant information in the stream?

Forward Error Correction

Staggering original & redundant samples by two samples

- As before, the length of receiver's buffering delay is a critical performance parameter



Forward Error Correction

k-way XOR

- Assume consecutive packet losses are rare and transmit the word-by-word XOR of groups of *k* samples
- <u>Example</u>: 3-way XOR

1010001101011111	1010001101010000	0000001111111111	0000001111110000
1010001101010011	1010001101000000	0000001111000001	0000001111010010
1010001100001111	1010111100000011	0000001100001010	0000111100000110
⋮	⋮	⋮	⋮
1010001000111010	1010111000101010	0000001000000000	0000111000010000
Sample 1 ⊗	*Sample 2* ⊗	*Sample 3* =	*Repair Sample*

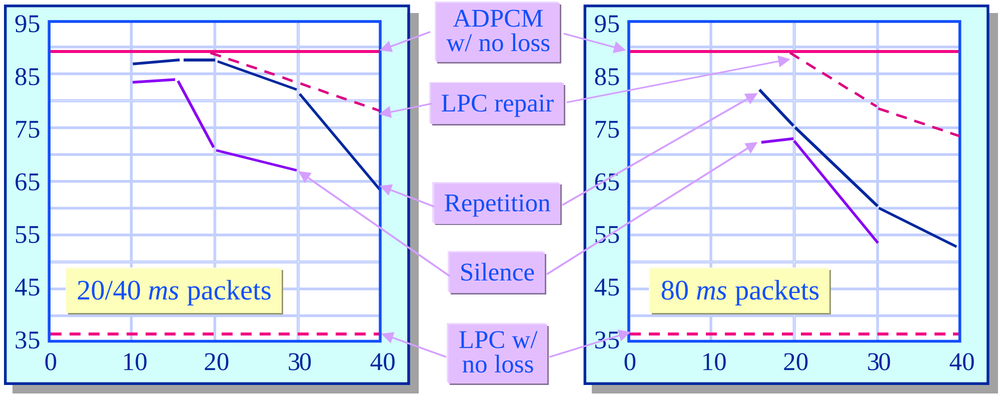
Forward Error Correction

Recoding/transcoding of original sample

- If losses are infrequent, perhaps we can get by with lower quality repairs
- <u>Example</u>: UCL's *Robust Audio Tool* (RAT) recodes the stream using an LPC codec for error recovery
  - » Normal samples are generated by an ADPCM codec
  - » LPC codec generates a 4.8 kbps stream (12 bytes/20 *ms* sample)
  - » Redundant samples separated from originals by 1 sample

RAT LPC Redundancy Experiments

Intelligibility *v.* percentage of packet loss



- Conclusion: LPC redundancy is likely not warranted with small packets; it is worthwile for large packets
  - » (This is due in large part to quality of LPC coded speech)

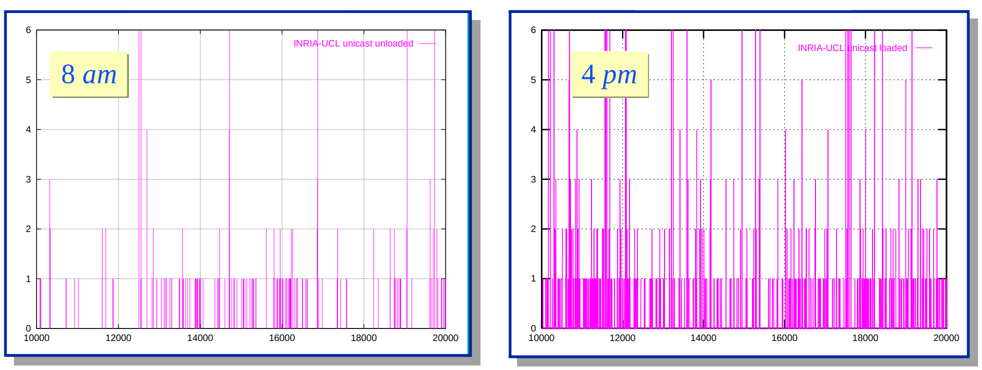
Foward Error Correction

Summary

- FEC will be effective means of dealing with packet loss if...
  - » we can tolerate the overhead
  - » consecutive packet losses are rare or
    we can tolerate higher playout delays

The Incidence of Consecutive Packet Loss

The INRIA unicast IVS experiments

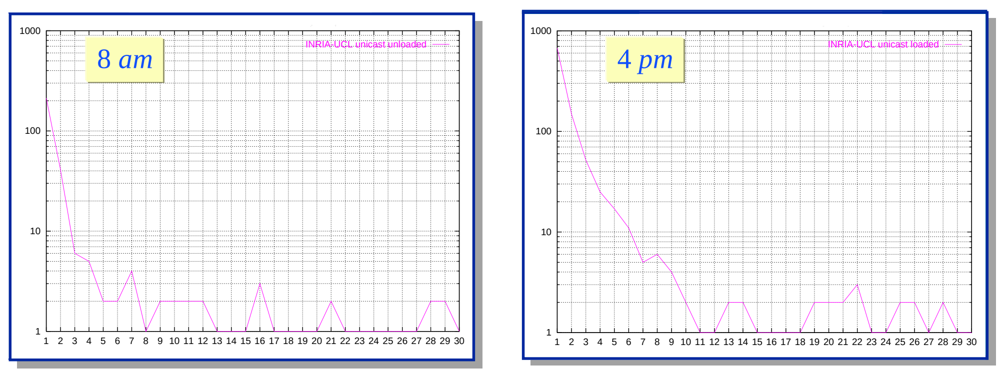


number of consecutive losses v. sequence number

- Packet loss from INRIA to UCL

The Incidence of Consecutive Packet Loss

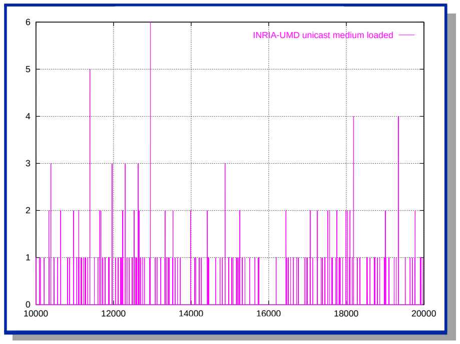
The INRIA unicast IVS experiments

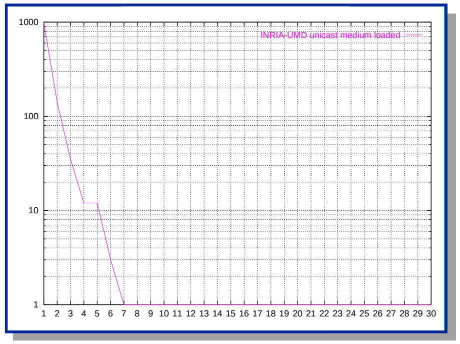


number of occurences of n consecutively lost packets v. n

- Frequency distribution of consecutive packet losses from INRIA to UCL

The Incidence of Consecutive Packet Loss
The INRIA unicast IVS experiments



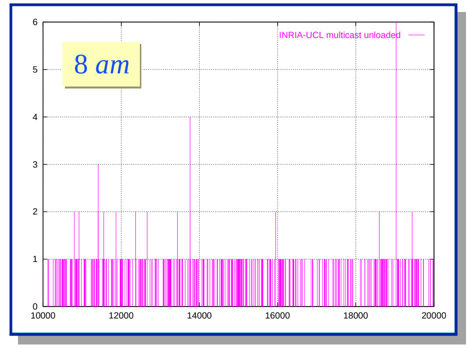


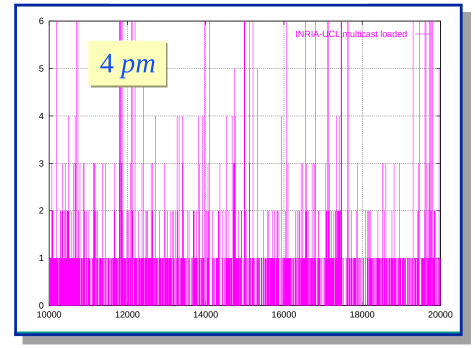
number of consecutive losses v. sequence number

number of occurences of n consecutively lost packet v. n

- Packet loss from INRIA to Maryland at 3 *pm* (9 *am* EST)

The Incidence of Consecutive Packet Loss
The INRIA multicast IVS experiments

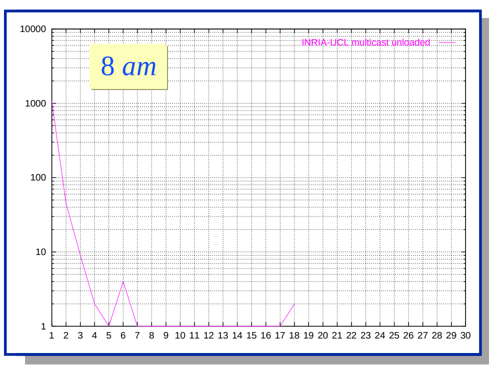


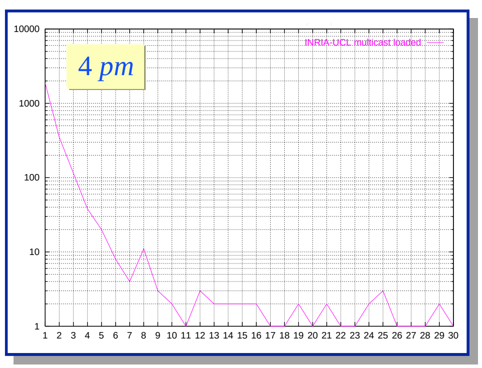


number of occurences of n consecutively lost packets v. n

- Packet loss from INRIA to UCL

The Incidence of Consecutive Packet Loss
The INRIA multicast IVS experiments



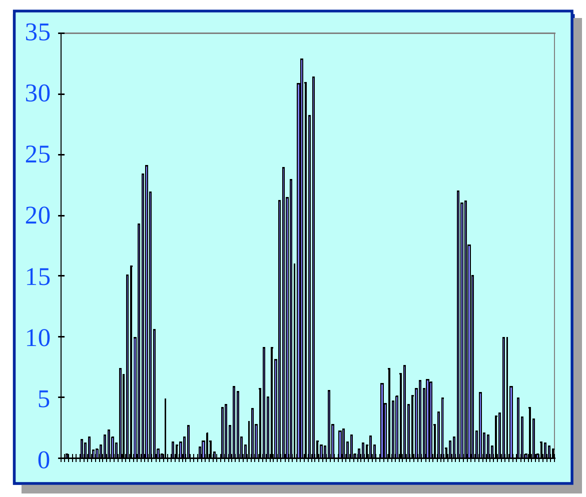


number of occurences of n consecutively lost packets v. n

- Frequency distribution of consecutive packet losses from INRIA to UCL

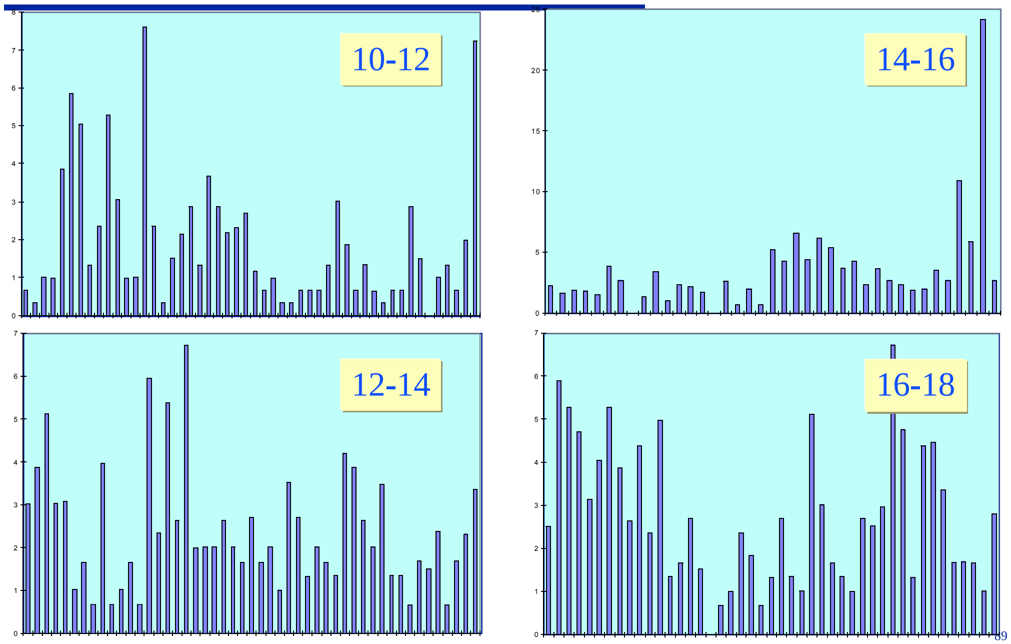
Packet Loss on the Internet Today
Audio packet loss for UNC-UW-UNC ProShare *x*mission

- Percentage of audio packet loss during a 5 minute interval
  - » Each line represents a 5 second average



Packet Loss on the Internet Today

Audio packet loss for UNC-UVa-UNC ProShare *x*mission



Adaptive, best-effort, multimedia networking

Outline

- IP message delivery semantics
  - » The four common Internet pathologies
- Ameliorating the effects of delay-jitter
  - » "60 ways to queue & play your media samples"
- Ameliorating the effects of packet loss
  - » Recovery of lost samples through retransmission
  - » Recovery of lost samples through the addition of redundant information
- Congestion control
  - » Adaptive media scaling and packaging

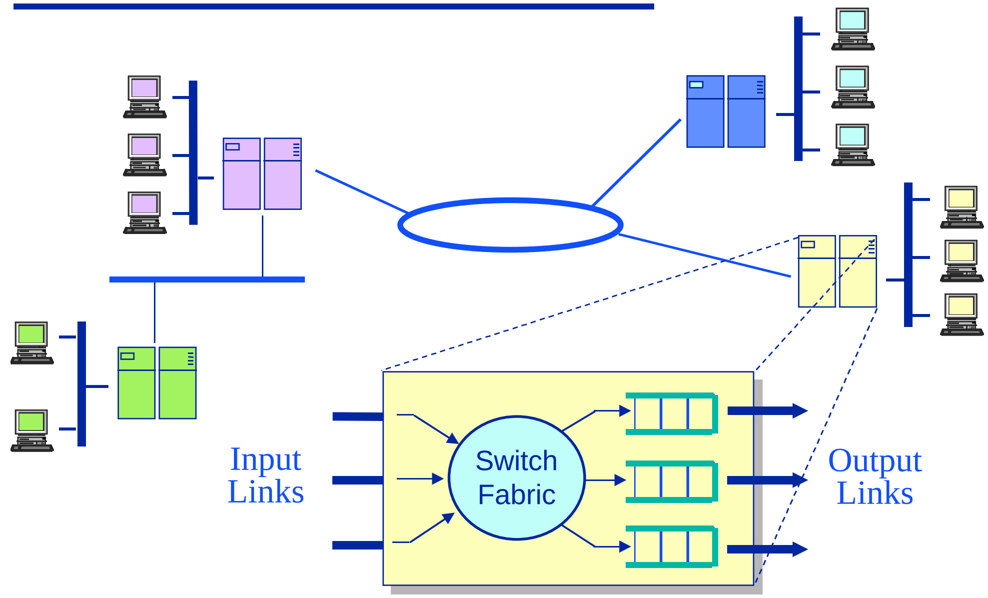
Best-Effort Multimedia Networking

Congestion control

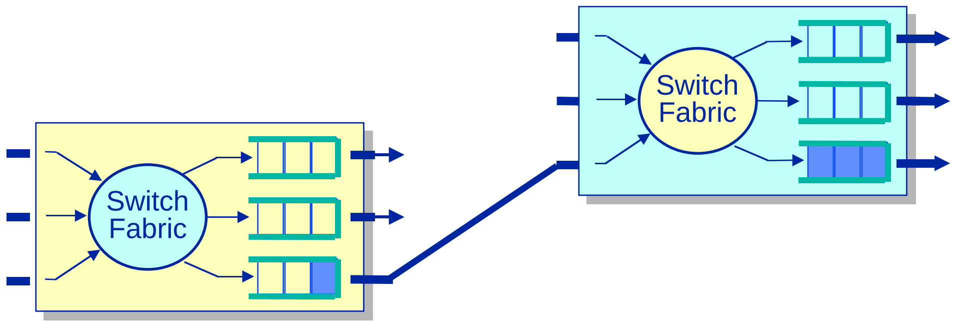
- Delay-jitter buffering, retransmission, and forward error correction *ameliorate* the effects of variation in end-to-end delay and packet loss
  - » They do not attempt to address the root cause
- Congestion control aims to eliminate or reduce these effects

Congestion Control

What is congestion?

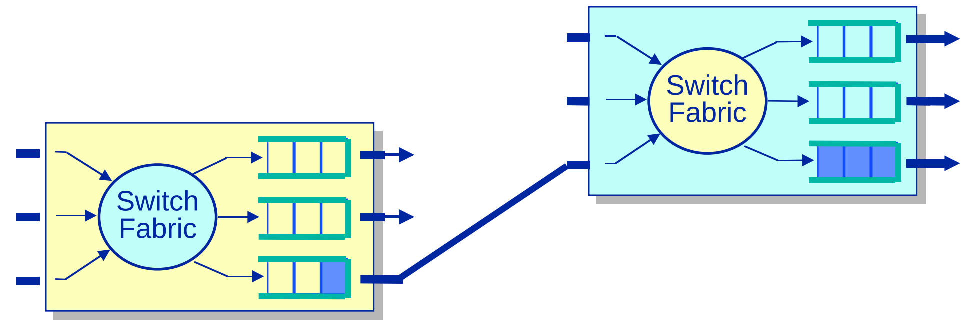


Congestion Control
The nature of congestion



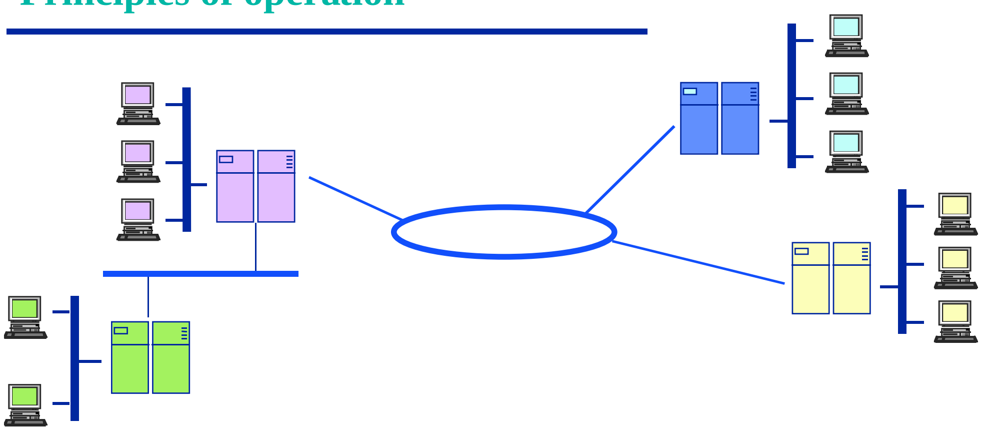
- What causes congestion?
  - » Did our multimedia stream(s) cause the network to be congested?
  - » Are there simply too many connections competing for too little bandwidth?

Congestion Control
The adaptive, best-effort, congestion control problem



- How can we make the best use of the (time varying) bandwidth that is available to our streams?
  - » How can we determine what this bandwidth is?
  - » How can we track how it changes over time?
  - » How can we match our codec(s)'s output the bandwidth "available" to our application?

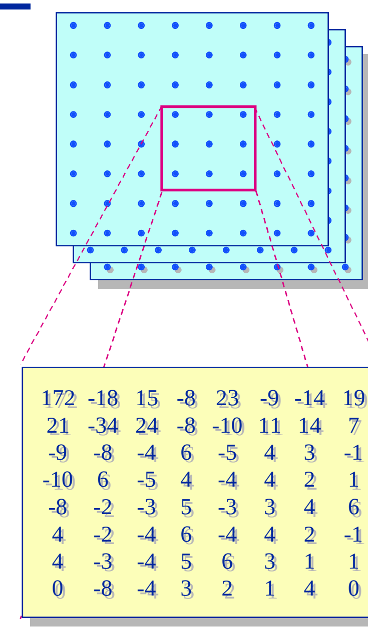
Adaptive, Best-Effort Congestion Control
Principles of operation



- Receivers periodically report throughput & loss statistics
- Sender adapts to match the bandwidth available
  - » Assume sufficient bandwidth exists for some useful execution of the system

Canonical Adaptive Congestion Control
Video bit-rate scaling

- Temporal scaling
  - » Reduce the resolution of the stream by reducing the frame rate
- Spatial scaling
  - » Reduce the number of pixels in an image
- Frequency scaling
  - » Reduce the number of DCT coefficients used in compression
- Amplitude scaling
  - » Reduce the color depth of each pixel in the image
- Color space scaling
  - » Reduce the number of colors available for displaying the image

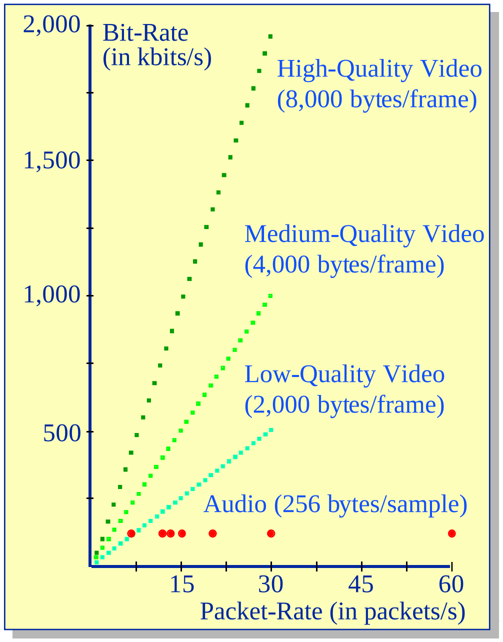


172	-18	15	-8	23	-9	-14	19
21	-34	24	-8	-10	11	14	7
-9	-8	-4	6	-5	4	3	-1
-10	6	-5	4	-4	4	2	1
-8	-2	-3	5	-3	3	4	6
4	-2	-4	6	-4	4	2	-1
4	-3	-4	5	6	3	1	1
0	-8	-4	3	2	1	4	0

UNC Adaptive Congestion Control

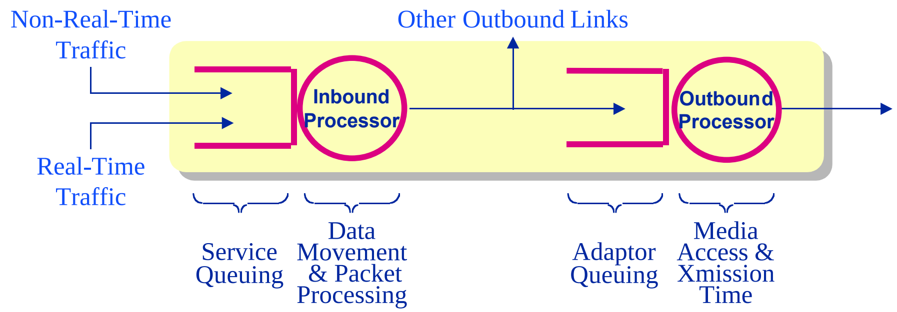
2-Dimensional media scaling

- Canonical approach to congestion
  - » Reduce (video) bit-rate
- Alternate approach
  - » View congestion control as a search of a 2-dimensional *bit-rate* x *packet-rate* space
  - » Scale bit- and packet-rates simultaneously to find a sustainable *operating point*



Bit- and Packet-Rate Scaling

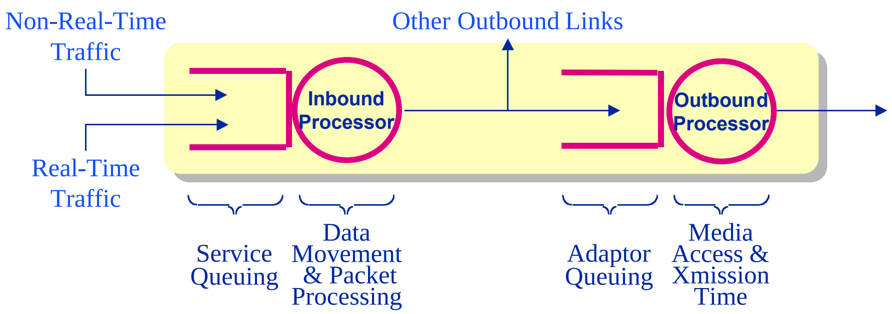
An analytic model of media scaling



- Capacity constraints
  - » the network is incapable of supporting the desired bit rate in any form
- Access constraints
  - » the network can not support the desired bit rate with the current packaging scheme

Two Types of Congestion Constraints

Two dimensions of adaptation

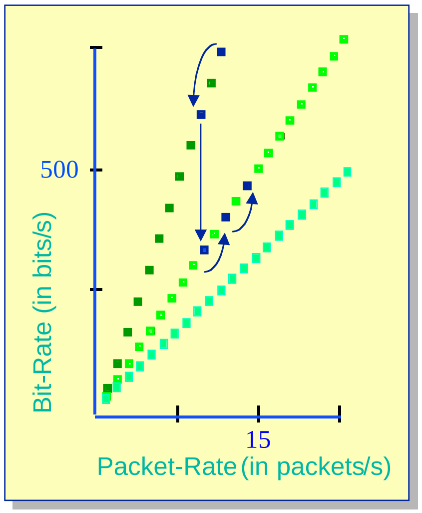


- Reduce the packet-rate to adapt to an access constraint
  - » Change the packaging or send fewer video frames
  - » Primary Trade-off: higher latency (potentially)
- Reduce the bit-rate to adapt to a capacity constraint
  - » Send fewer video frames or fewer bits per video frame
  - » Primary Trade-off: lower fidelity

2-Dimensional Scaling Example

The "Recent Success" heuristic

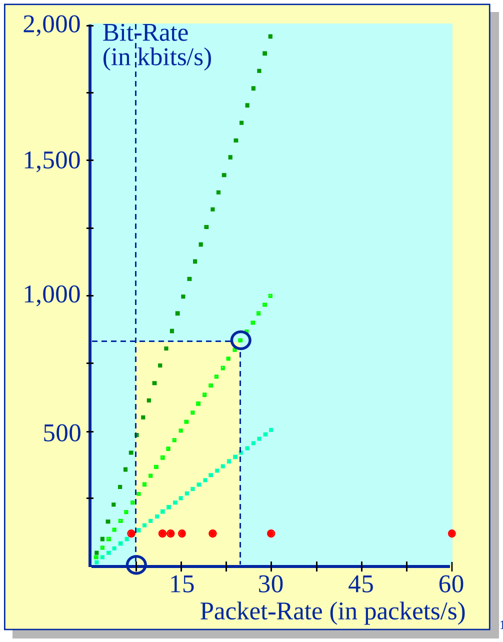
- Initial operating point:
  (*high quality*, 12 *fps*)
- First adaptation:
  (*high quality*, 10 *fps*)
  - » congestion persists
- Second adaptation:
  (*medium quality*, 10 *fps*)
  - » congestion relieved
- First probe:
  (*medium quality*, 12 *fps*)
- Second probe:
  (*medium quality*, 14 *fps*)



2-Dimensional Media Scaling

Finding a sustainable operating point

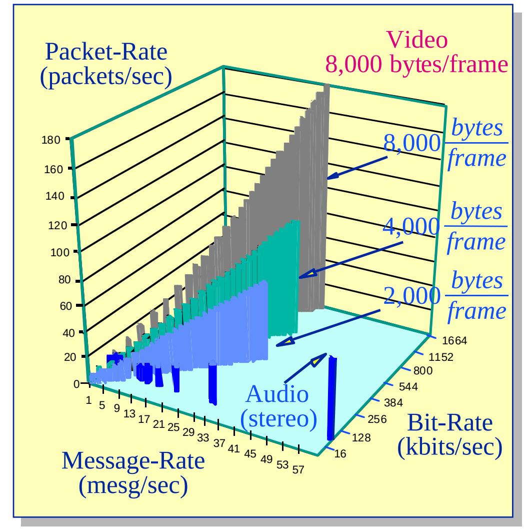
- The search space can be pruned by eliminating
  - » points that lead to inherently high latency
  - » points that lead to high latency given the state of the network



2-Dimensional Media Scaling

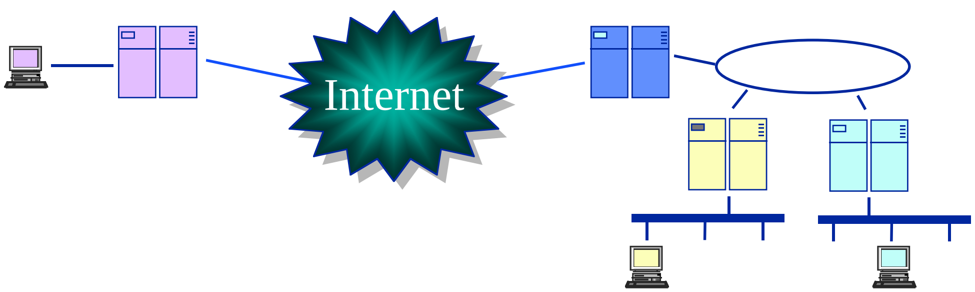
Dealing with effects of fragmentation

- The problem
  - » A sender can only (directly) effect the *message rate*, not the *packet rate*
- Does fragmentation render message-rate scaling obsolete?



Adaptive, 2-Dimensional Media Scaling

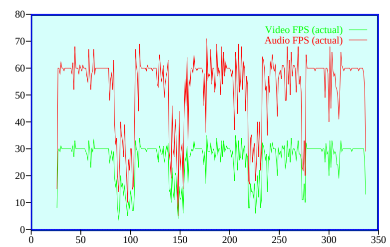
Does it work?



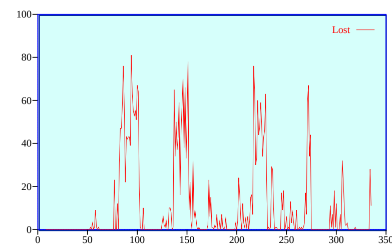
- Campus-szed internets — yes!
  - » It “solves” the first-mile/last-mile problem
- The Internet? — *well...*
  - » Does our necessary condition for success hold?
  - » Does it hold often enough to be useful?
  - » How much “room” is there for 2-D scaling in most codecs?

Media Scaling Evaluation on the UNC Campus

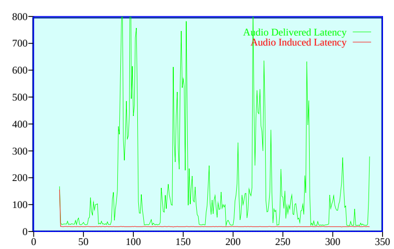
Performance with no media scaling



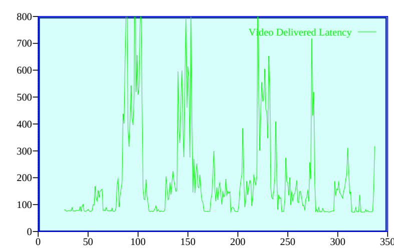
Throughput (frames/sec)



Packet Loss



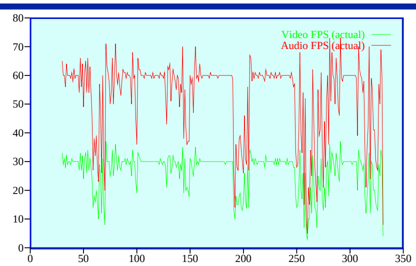
Audio Latency (ms)



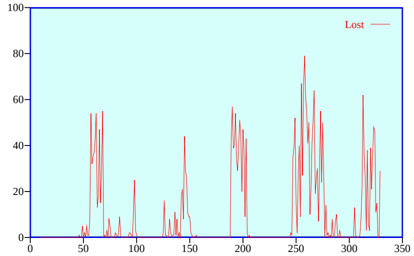
Video Latency (ms)

Media Scaling Evaluation on the UNC Campus

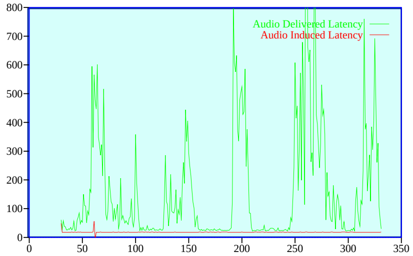
Performance with video scaling only



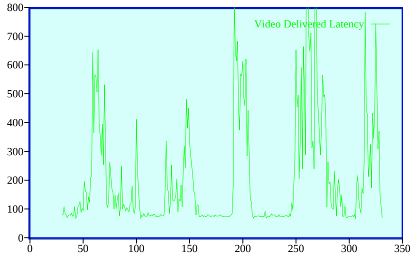
Throughput (frames/sec)



Packet Loss



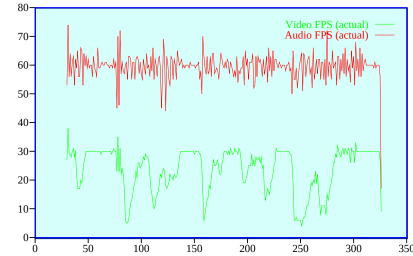
Audio Latency (ms)



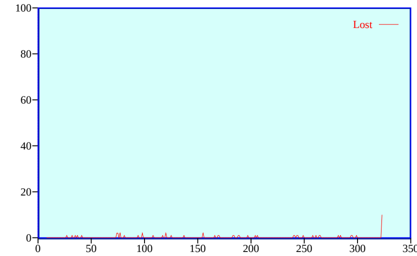
Video Latency (ms)

Media Scaling Evaluation on the UNC Campus

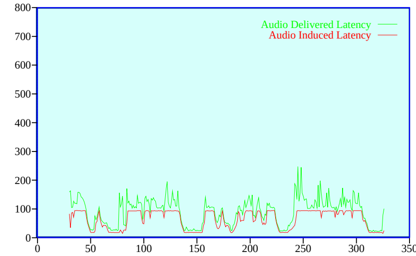
Performance with 2-dimensional scaling



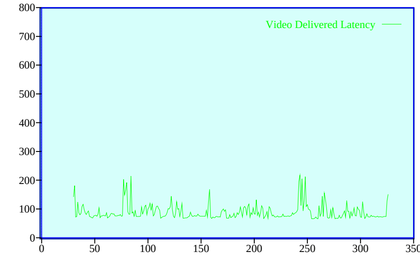
Throughput (frames/sec)



Packet Loss



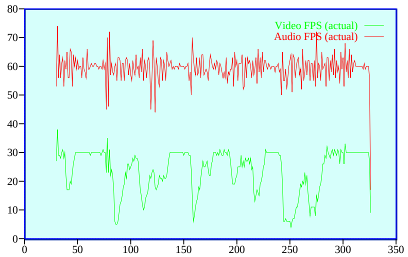
Audio Latency (ms)



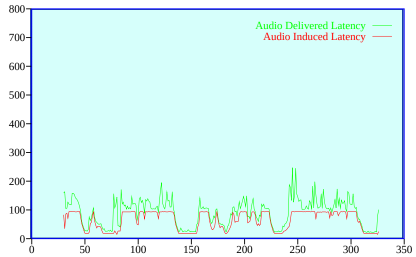
Video Latency (ms)

Media Scaling Evaluation on the UNC Campus

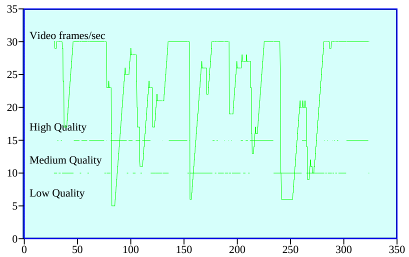
2-dimensional adaptation over time



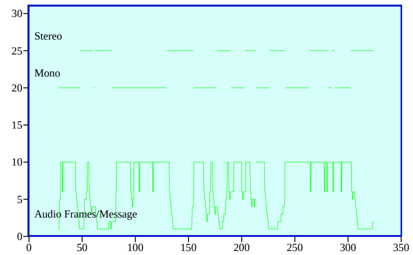
Throughput (frames/sec)



Audio Latency (ms)



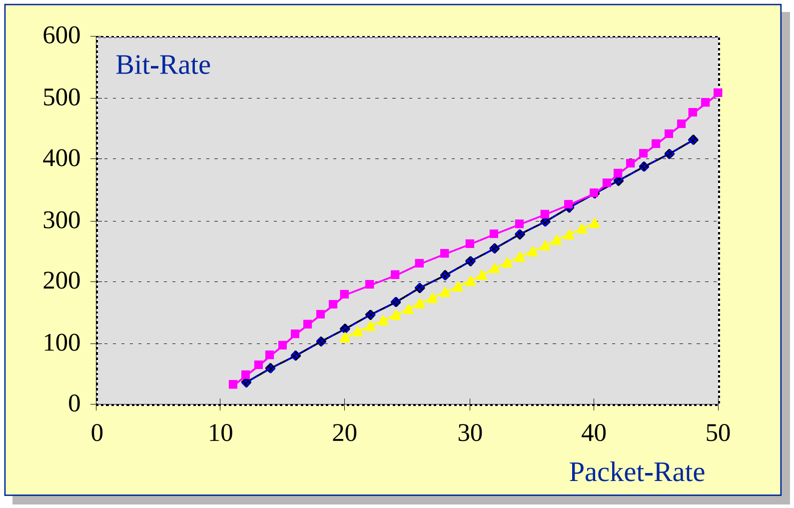
Video Adaptation Over Time



Audio Adaptation Over Time

Media Scaling Evaluation on the Internet

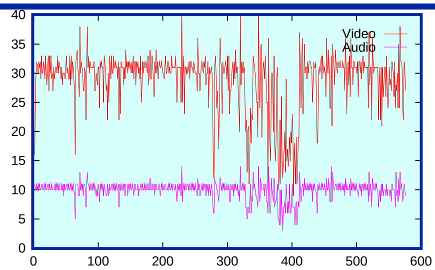
Media scaling in Intel's ProShare™ codec



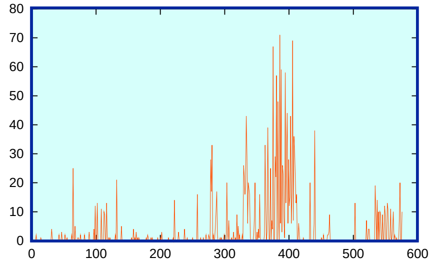
Proshare operating points

Media Scaling Evaluation on the Internet

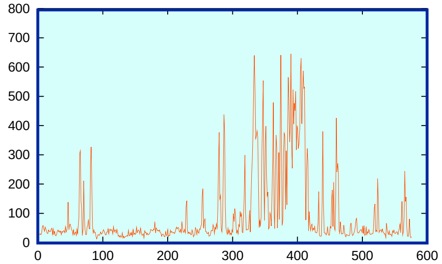
ProShare with no media scaling



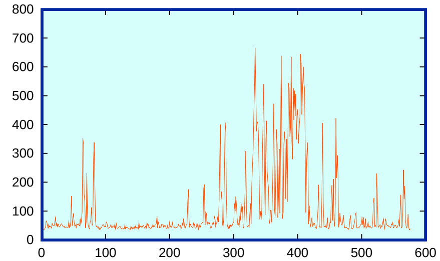
Throughput (frames/sec)



Packet Loss



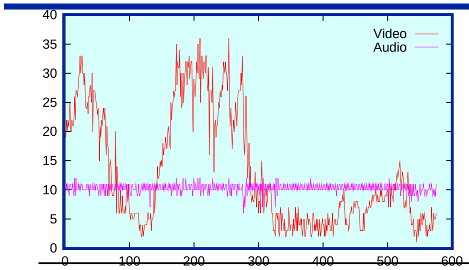
Audio Latency (ms)



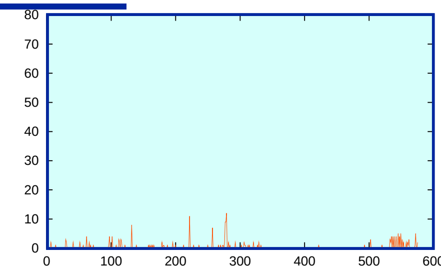
Video Latency (ms)

Media Scaling Evaluation on the Internet

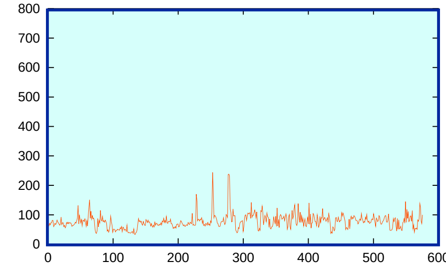
ProShare with 2-dimensional media scaling



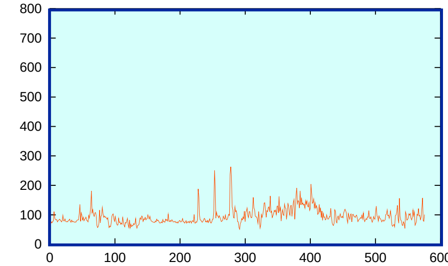
Throughput (frames/sec)



Packet Loss



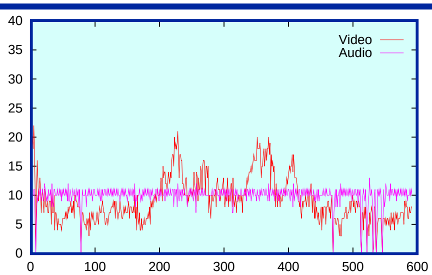
Audio Latency (ms)



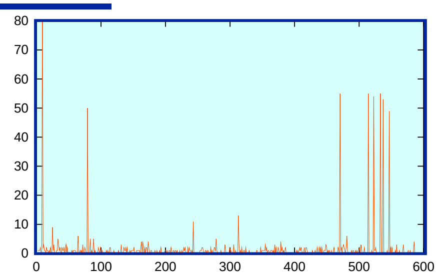
Video Latency (ms)

Media Scaling Evaluation on the Internet

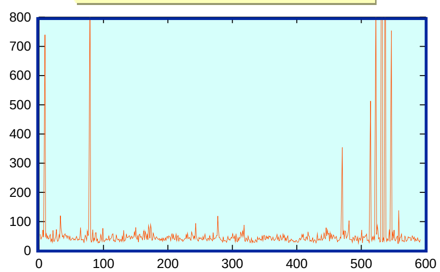
ProShare with video scaling only



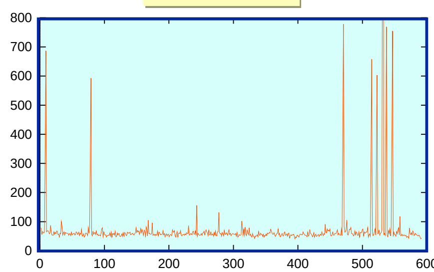
Throughput (frames/sec)



Packet Loss



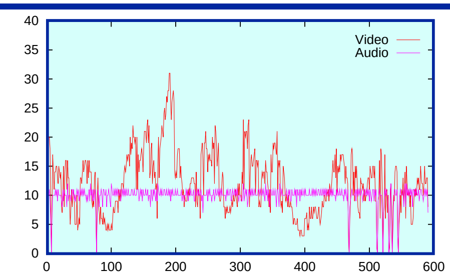
Audio Latency (ms)



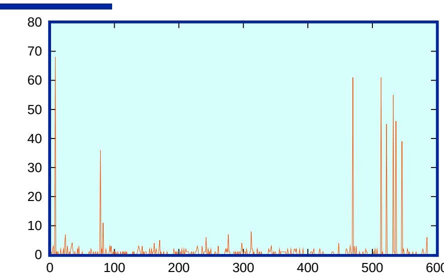
Video Latency (ms)

Media Scaling Evaluation on the Internet

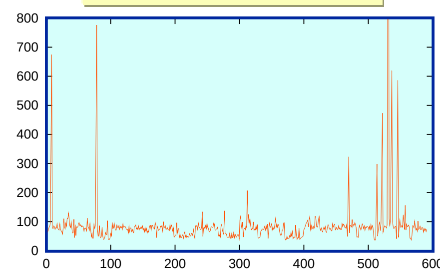
ProShare with 2-dimensional media scaling



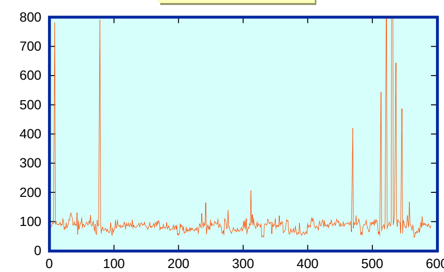
Throughput (frames/sec)



Packet Loss



Audio Latency (ms)



Video Latency (ms)

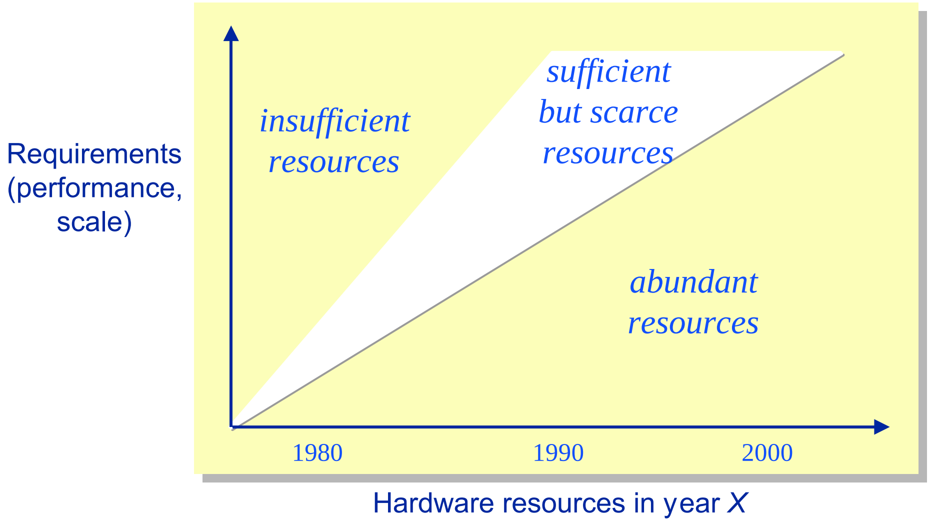
Sustainability Results
Adaptive methods on the Internet

- Results of an Internet performance study from UNC to UVa
  - » Repeated trials from 10 am to 7 PM weekdays
  - » Trials separated by at least two hours
  - » Scattered over three months

Time Slot	Sustainable	Not Sustainable	% Sustainable
10:00-12:00	6	3	67%
12:00-14:00	4	4	50%
14:00-16:00	1	11	8%
16:00-18:00	3	9	25%
18:00-20:00	4	5	44%
Percentage	36%	64%	

Real-time data delivery on the Internet Today
What's the problem?

- Do we need more bandwidth or just better management of the existing bandwidth?



Real-time data delivery on the Internet Today
Where do we go from here?

- Provide "best-effort" service by adapting media streams
  - » Monitor & provide feedback on performance
  - » Bias transmission and processing of media to ameliorate the effects of congestion

- Provide true quality-of-service through reservation of resources in the network
  - » Requires coordination amongst all parties
    - admission control
    - policing
    - ...