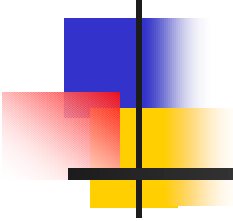


Multicast Transmission of Scalable Video using Receiver-driven Hierarchical FEC



Wai-tian Tan • Avidesh Zakhor
University of California, Berkeley



Problem Overview

- Challenges: Heterogeneity
 - Bandwidth heterogeneity
 - Different bandwidths from a source to its receivers
 - Solved by: scalable video + layered multicast
 - What about other forms of heterogeneity?
 - Packet loss rate
 - Different receivers of a source may experience different loss rate and desire different levels of protection
 - Latency tolerance
 - Receivers have different latency tolerances, e.g., passive vs. interactive viewing of a live lecture



Goal

- Provide a hierarchical FEC (HFEC) framework that is:
 - **Flexible** to cater to needs of different receivers by allowing them to individually trade-off latency for reliability
 - **Efficient** in bandwidth utilization
 - **Architecturally simple** in not requiring network support beyond IP-multicast



Outline

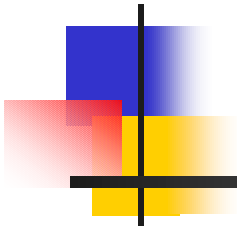
- Previous work – multicast error control
- The Hierarchical FEC Framework
 - Overview
 - Choice of error control code
 - Choice of scalable compression
- Experimental Results
 - Sender/Receiver implementations
 - Experimental setup and results
- Appendix: Multicast Rate Control



Previous Work - Multicast error control

- Most works on multicast error control focus on providing **perfect** End-to-end reliability
 - Over-restricting and expensive for video applications
- Retransmissions: guarantees perfect reliability
 - SRM - *Floyd et.al, 1997*
 - LVMR - *Li et.al, 1997*
- Hybrid FEC+Retransmission are used to reduce the amount of necessary retransmissions for perfect reliability
 - Pejhan, Schwartz, 1996
 - Nonnenmacher et.al., 1998

The Hierarchical FEC (HFEC) Framework





HFEC - Overview

- Use Hierarchical FEC to provide different levels of protection to different users
 - Code video in layered/hierarchical manner
 - Code FEC data in layered/hierarchical manner
 - Carry each data and FEC layer in a different multicast group/address
 - Each receiver decides on how many data and FEC layers to subscribe based on past reception history
- Introduce latency to FEC layers to achieve interleaving [Bolot 1997]

HFEC – Timing illustrated

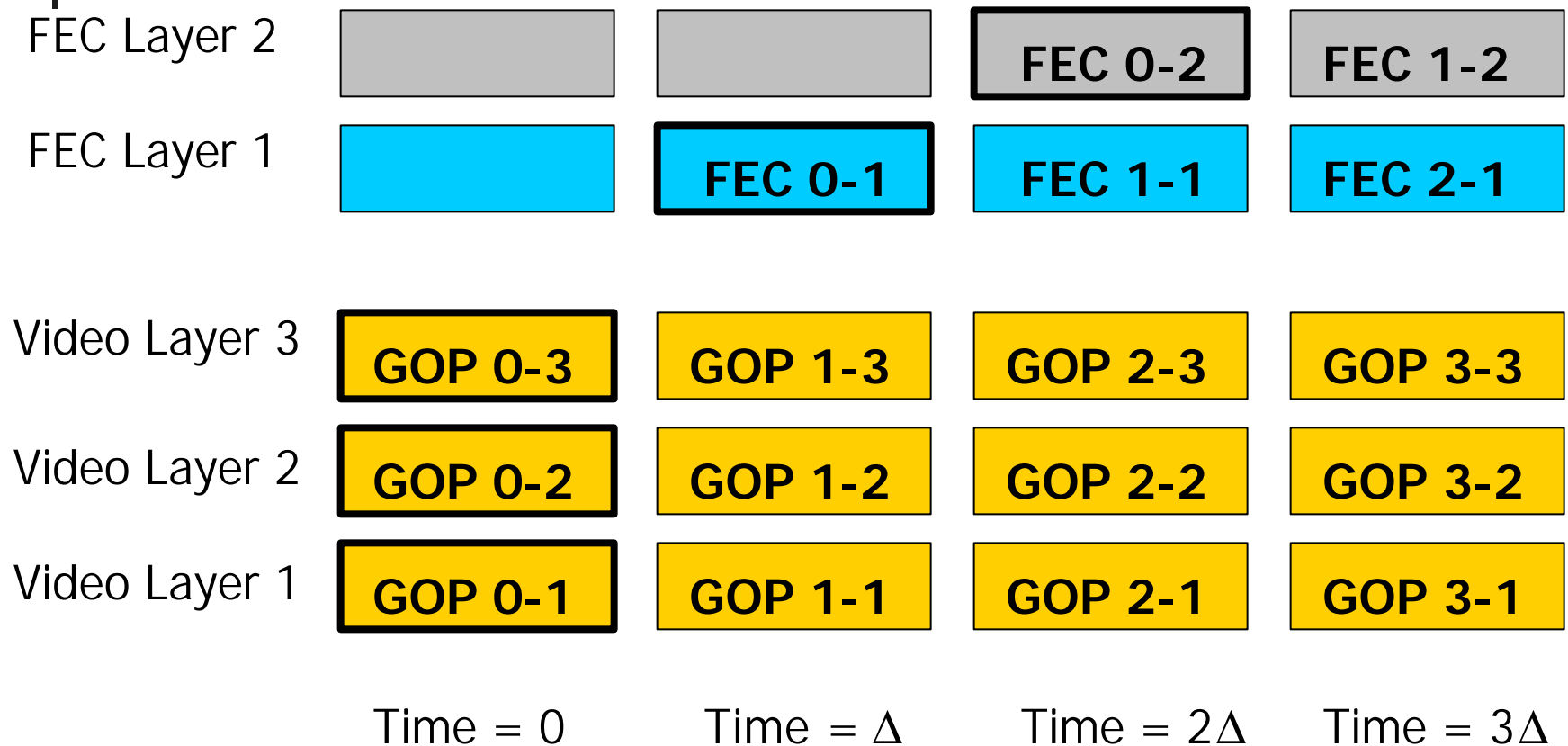


Fig.1 : Timing diagram for Hierarchical FEC scheme



HFEC – Timing illustrated (cont'd)

- Groups of frames are compressed in a scalable manner (3 layers shown) and transmitted in the same time slot.
 - E.g., GOP-0-1, GOP-0-2, GOP-0-3 are successive refinements to the same group of frames
- FEC layers are also compressed in a layered fashion but transmitted in later times.
 - E.g., FEC-0-1, FEC-0-2 protects GOP-0-1 to GOP-0-3
- Interleaving:
 - Counteracts bursty losses
 - Forms trade-off between reliability and latency

HFEC – Meeting needs of receivers

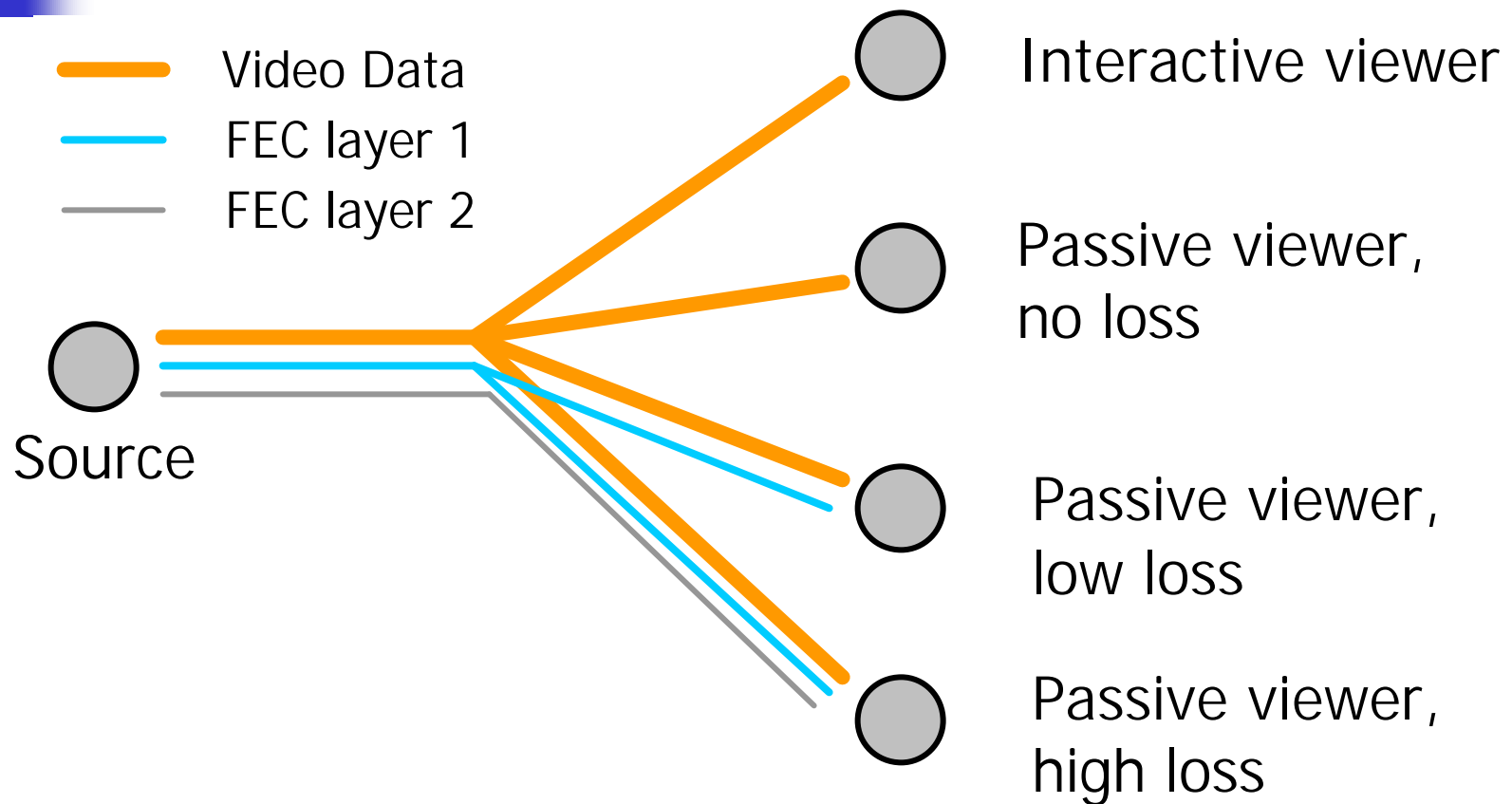


Fig.2 : HFEC Meeting needs of different receivers



HFEC – Advantages revisited

- Flexible
 - Allow receivers to adjust reliability according to experienced loss rate and latency requirements
- Efficient
 - FEC layers only reach receivers that need them
 - Multicast delivery of FEC and data layers further saves bandwidth when compared to unicast
- Architecturally simple – pure FEC scheme
 - No complex architecture for multicast retransmissions
- Scalable video:
 - Facilitates unequal error protection
 - No overall data expansion: reduce data rate to make room for FEC

HFEC – Partitioning into layers

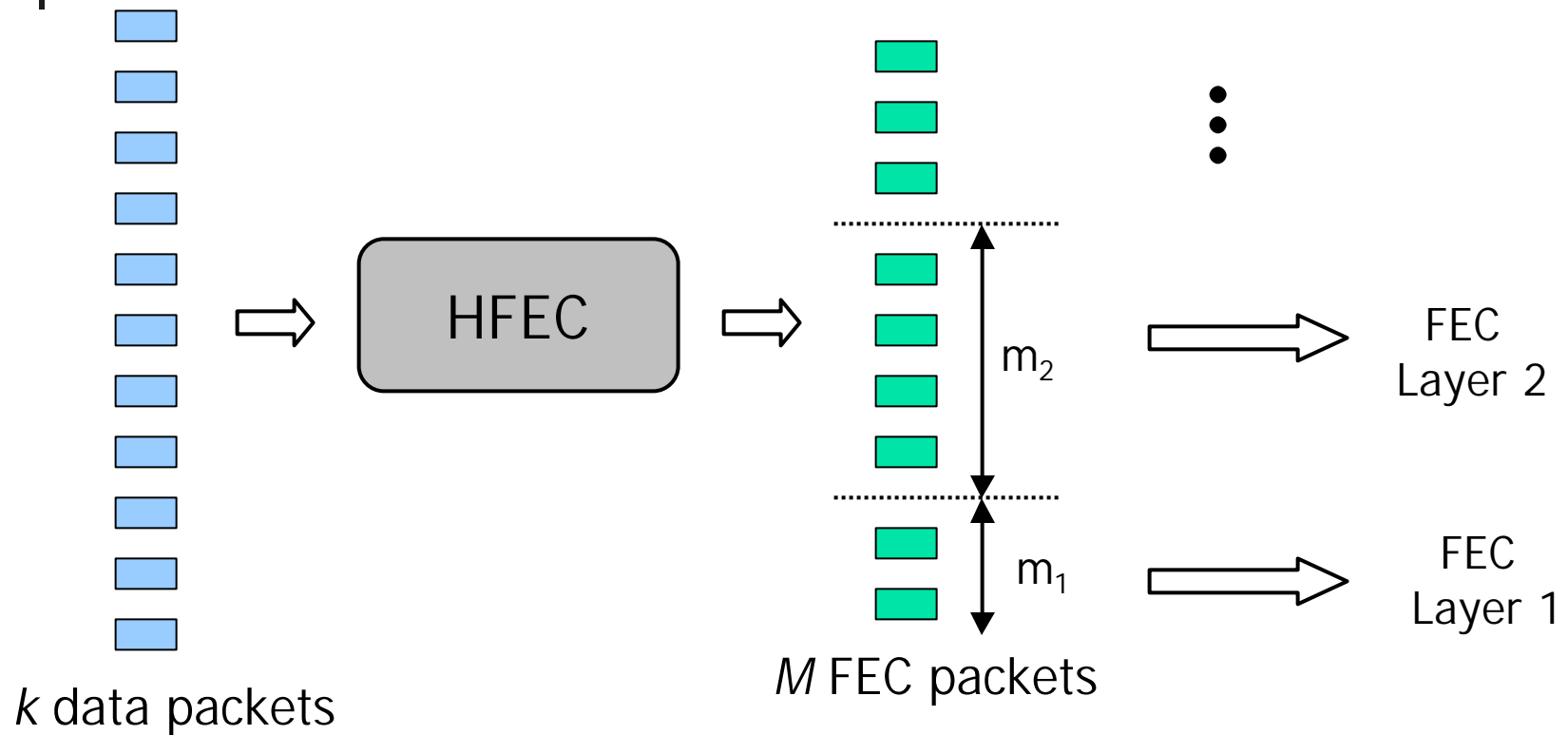


Fig.3 : Partitioning FEC packets into FEC layers



HFEC – Design problems

- $M = m_1 + m_2 + \dots + m_w$ FEC packets correct at most M packet losses
 - How many losses can we make the first $m_1, m_1 + m_2, \dots$ FEC packets correct ?
- Given m_1 optimally constructed FEC packets that correct m_1 packet losses
 - How to construct m_2 FEC packets so that the $m_1 + m_2$ packets correct as many losses as possible.
 - In general, how to optimally construct the subsequent m_i FEC packets



MDS Codes - Advantages

- Use maximal distance separable (MDS) codes for HFEC:
 - $M = m_1 + m_2 + \dots + m_w$ FEC packets corrects M packet losses (optimal)
 - The first $S_j = m_1 + m_2 + \dots + m_j$, $j < w$, FEC packets corrects S_j losses (optimal)
- No loss in error correction capabilities by imposing a hierarchical structure
- Example: (punctured) Reed-Solomon codes



MDS Codes – Computing cost

- For a $(77, 70)$ punctured Reed-Solomon code over $GF(256)$
 - 7 parity packets are generated from every 70 data packets
 - The 7 parity packets can be arbitrarily distributed into FEC layers. All layers together provide 10% protection
 - For 1 Mbps bit-stream, FEC encoding and decoding each roughly consumes 5% of a 400 MHz Pentium-II



Scalable Video – Error resilience

- Why error-resilience ?
 - FEC can only provide fixed levels of protection but losses are bursty
 - Interleaving of FEC layers only alleviates impact of bursty losses and does not guarantee packet delivery
 - Finite reaction time for rate-control mechanism to react to losses due to addition of new traffic
 - Avoid reacting to network transients
 - Cost of changing subscription levels in multicast
- How to achieve error-resilience ?
 - Minimize data dependency between packets



Scalable Video – A resilient codec

- Error-resilient scalable compression
 - Minimal dependency among packets
 - High utilization of survived packets leading to superior reconstruction video quality
 - Good speed performance
 - Real-time software encoding/decoding
 - ICIP 98
 - Tan, Zakhor, "Real-time Internet Video Using Error Resilient Scalable Compression and TCP-friendly Transport Protocol". *IEEE Trans. Multimedia*, June 1999

Scalable Video – Packet dependency

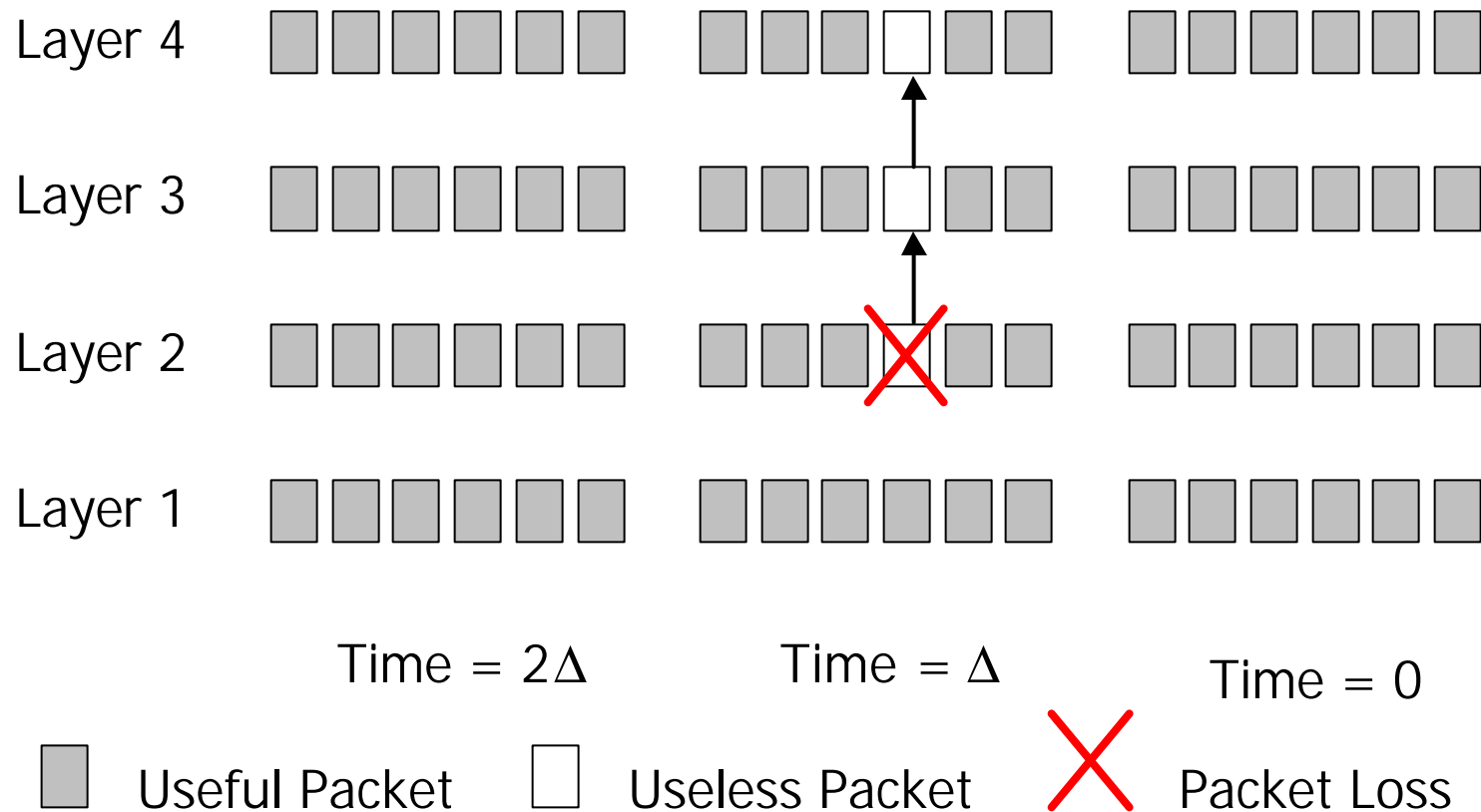


Fig.4 : Packet dependence in proposed scalable video



Experimental Results



Implementation - Sender

- Source layout
 - 4 data layers of 100 kbps each
 - 4 FEC layers of 50 kbps each
 - FEC layers only protects first data layer
 - Simplifies optimization task at receivers
 - A good approximate solution since first layer carry much higher importance than subsequent layers
 - Each FEC layer delayed an additional 1/3 second
- Source material
 - Repeated transmission of a high motion scene from *Raiders of the Lost Ark* (12 fps)
 - GOP size of 4 is used



Implementation – Receiver

- Rate control – estimating available bandwidth
 - A simple approach
 - Each receiver determines its rate independently
 - Rate is determined as a function of measured packet loss rate and round-trip-delay from source
 - Can be replaced by any rate control mechanisms
- Rate allocation – partitioning available bandwidth
 - Seeks partitioning of available bandwidth between data and FEC layers by minimizing expected distortion
 - Expected distortion is given by packet loss rate, rate-distortion characteristics of source material, and data dependency between packets are needed

Experimental Setup

- Two experiments are performed back to back using topology in Fig.5
- In the **control** experiment, both receivers at Gatech and ISI run rate control and allocate all the rates to data
- In the **HFEC** experiment, part of the rates are allocated to FEC layers to yield higher image quality

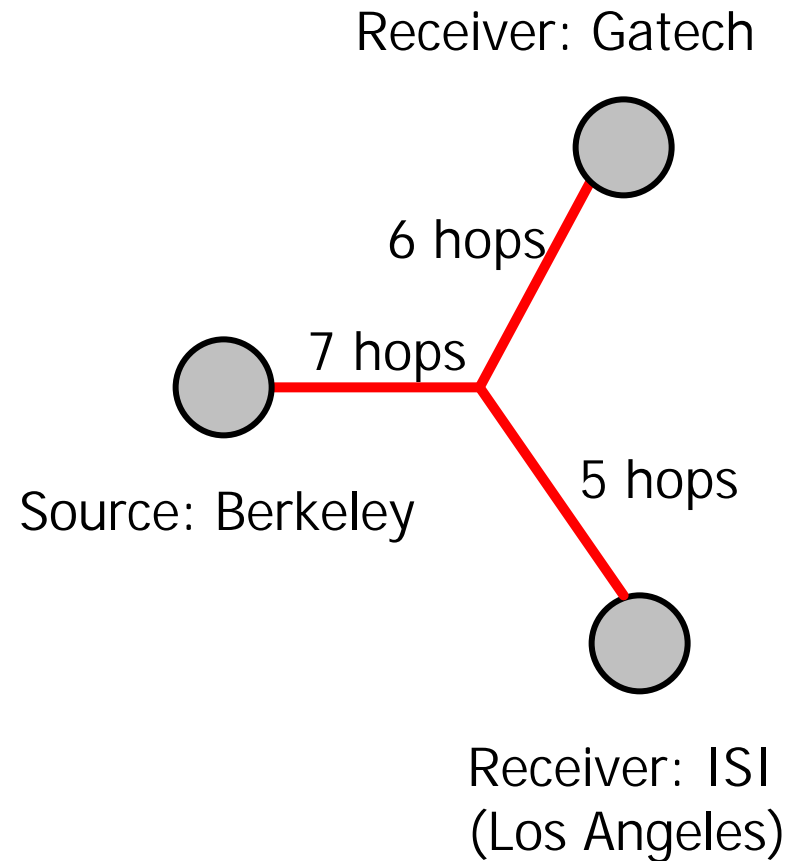


Fig.5 : Topology for experiment

Result – Packet loss rates

- No loss is experienced at the receiver in Gatech in both the control and HFEC experiments
- For the ISI receiver, packet loss rates are given by Fig. 6
- Similar loss rates across experiments indicates that network conditions did not change significantly

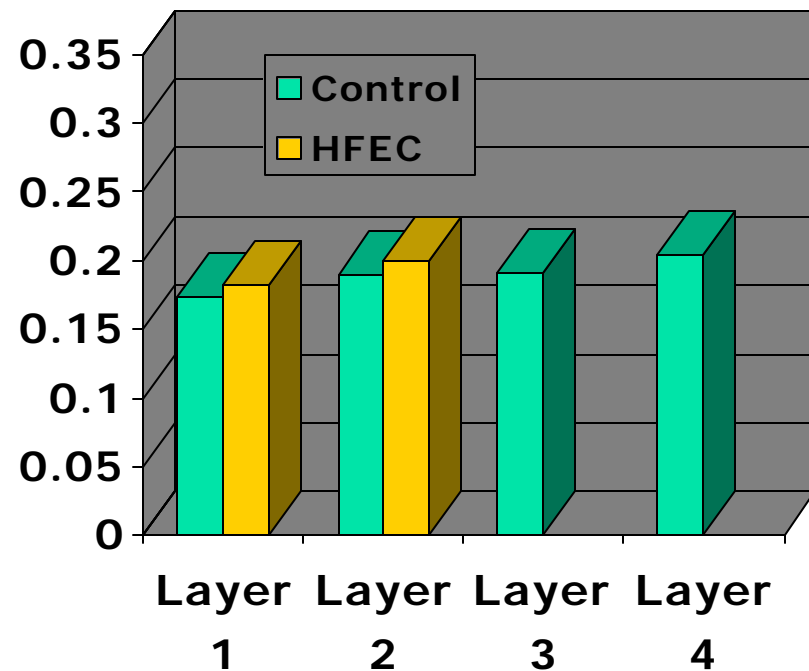


Fig.6 : Observed packet loss rate
For different data layers (Berkeley-ISI)

Result – MSE traces

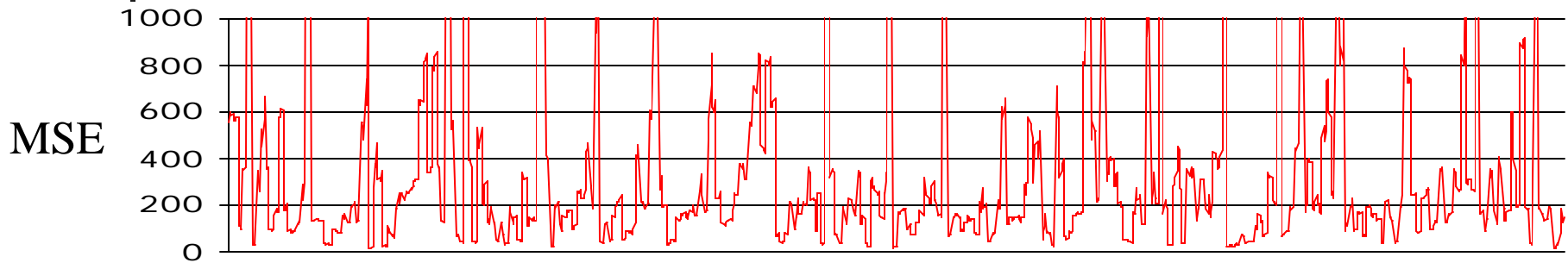


Fig. 7: MSE trace for Control experiment (Berkeley-ISI)

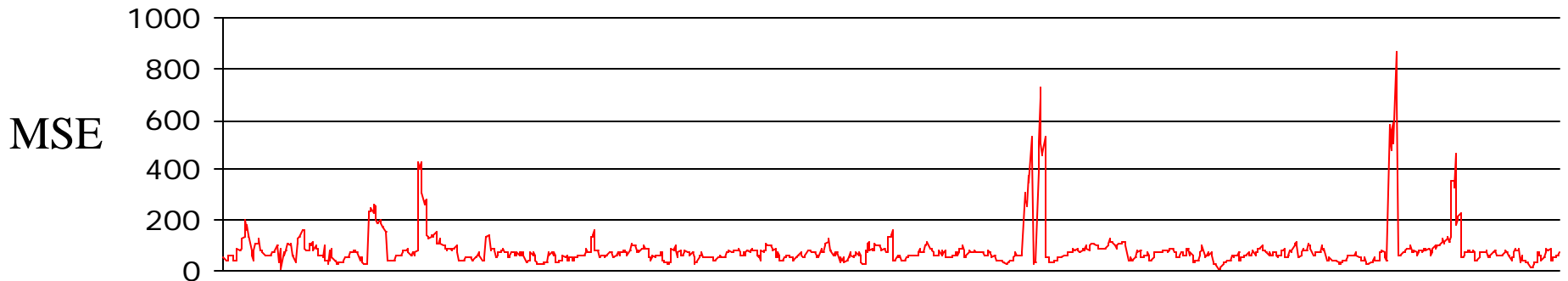


Fig. 8: MSE trace for HFEC experiment (Berkeley-ISI)



Result – MSE traces (Cont'd)

- In control experiment
 - 400 kbps allocated to data
 - Average MSE is 333
 - Compared to original images before compression
 - Much variability in MSE due to bursty losses
- In HFEC experiment
 - 200 kbps for data and 200 kbps for FEC
 - Of the 300 transmitted GOPs, 130 has at least 1 lost packet at first data layer, and 116 are corrected by FEC
 - Average MSE is 85

Result – Packet loss trace

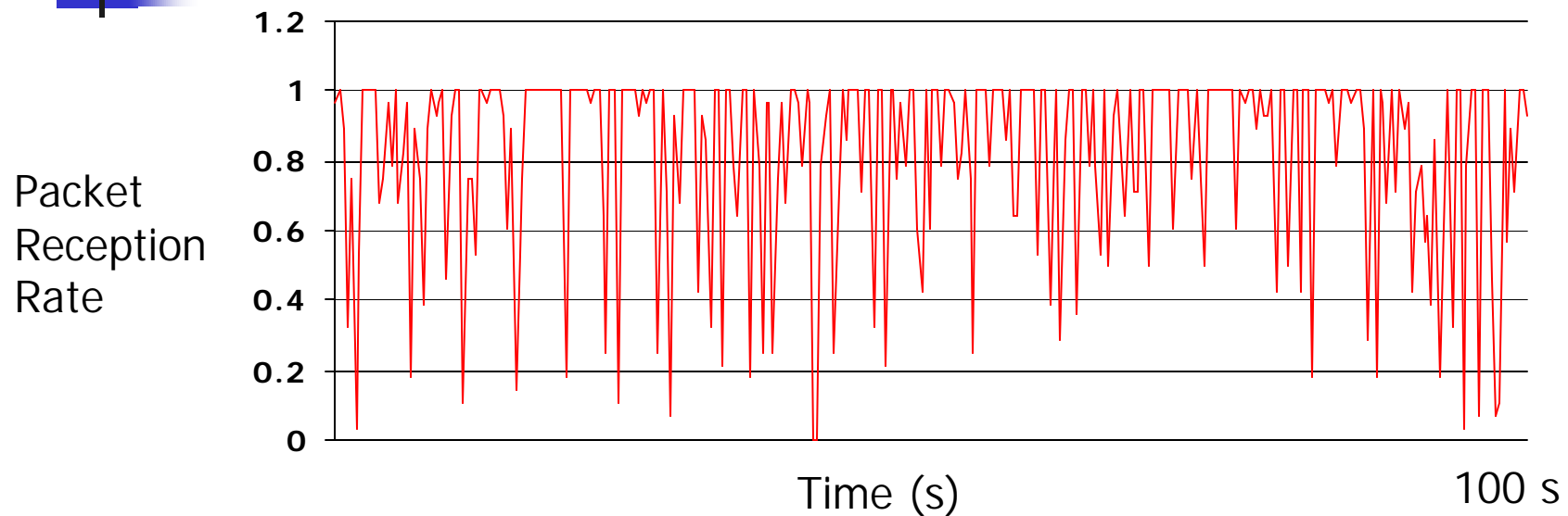


Fig. 9: Packet reception rate for HFEC experiment (Berkeley-ISI)

- High variability in packet loss rate (measured in 1/3 s intervals)
- Delayed transmission of FEC layers is necessary₂₆



Summary

- Presented HFEC as a flexible, efficient, and architecturally simple framework to allow receivers to tradeoff latency for reliability
- Presented experimental results over MBONE to illustrate the potential utility of the scheme



Appendix

Multicast Rate Control – An Equation-based Approach



Outline

- Traditional multicast rate control
 - Challenges and Approaches
 - RLM and RLC
- Equation-based rate control (ERC)
 - What and Why
 - Design considerations
- Performance: fairness
 - Multiple sessions
 - With and without cross traffic



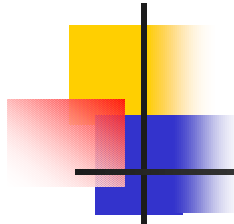
Multicast Rate Control - Traditional

- Regardless of current rate:
 - On congestion drop layers
 - Otherwise probe for bandwidth to add layers
 - Motivated by TCP rate control
- Challenges
 - IP-multicast
 - Unaware of network topology
 - Unaware of rates of other receivers
 - No mechanism for cross group communications
 - Given IP-multicast, how to achieve fair allocation when multiple sessions compete for bandwidth



Receiver-driven Layered Multicast

- **PHILOSOPHY:** synchronization through communication
 - Limits probing activity and share information about a probe to everyone
 - Add layer only when one's OWN probe succeed
 - Ignore losses due to other people's probe
- Problems
 - Cost of communication – non-scalability
 - Cannot synchronize across multiple multicast sessions



RLC

- No explicit synchronization
- Mimics TCP additive-increase, multiplicative-decrease behavior
 - Total layer sizes increase geometrically
 - E.g., layer sizes 1,1,2,4,8,16,32,...
 - Receivers can only join at synchronization points (s.p.). Separation between s.p. across layers proportional to layer size
- Knows for free when other receivers within a session can add layers
- Synchronization fails across sessions



ERC - What

- Estimates bandwidth using a continuous strictly monotonic function $R = F(p)$
 - “p” is packet loss rate.
 - Can include RTT as well
- Move towards target rate R
 - Add/drop/no-change w.r.t. current rate
 - Action depends on current rate



ERC - Why

- No explicit synchronization – scalability
- Fair share of bandwidth across bottleneck
 - Sessions sharing a bottleneck will measure similar packet loss rates and therefore computer similar throughput



Performance - Fairness



Fairness Function

$$F(X_1, X_2, \dots, X_N) = 1 - \frac{1}{2(N-1)} \sum_{i=1}^N \left| \frac{X_i}{\mathbf{m}} - 1 \right|$$

- X_i : throughput of flow i .
- \mathbf{m} : mean of all X_i
- $F \in [0, 1]$ measures fairness:
 - $F = 1$ iff all X_i are the same
 - $F = 0$ iff exactly one X_i is non-zero
 - $F(1, 3, 5, 7, 9) = 0.7$, $F(50, 51, 52) = 0.990$



Linear or Exponential Layers

- Linear layers
 - Exploits available bandwidth better
 - More stable behavior as adding a layer is a less drastic event
 - All layers have same rate
- Exponential layers advantages:
 - Represents a wide range of bit-rates with relatively few rates
 - Fewer network resources
 - Useful for sources not finely scalable
 - Layers have rate $R, R, 2R, 4R, 8R, 16R, \dots$

Experimental Setup - Dumbbell

- Use *ns* simulator
- Fix
 - - RTT
 - b : Average bandwidth per flow (500 kbps)
- Vary
 - N : number of flows
 - g : granularity, or the no. of layers that can pass through b
 - M : Max-to-mean ratio, i.e., max rate / b .
 - Limits throughput if $M \leq N$

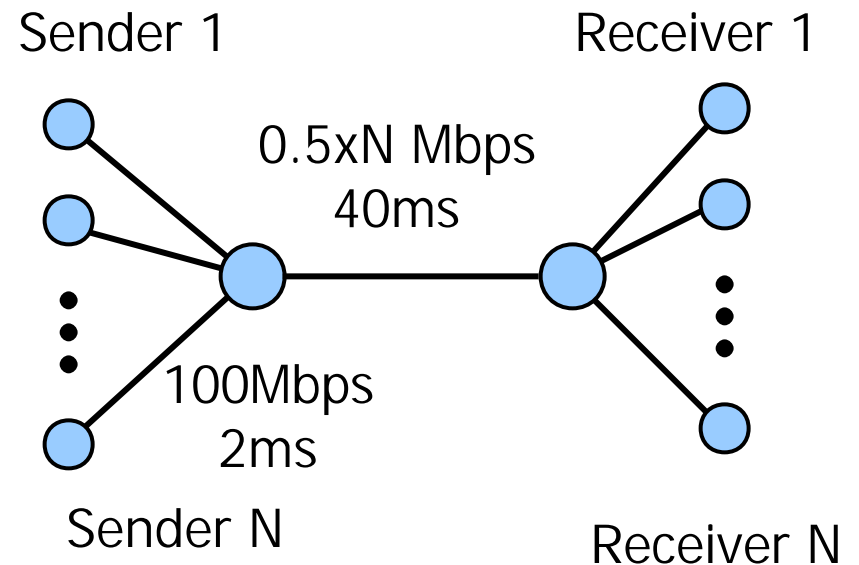


Fig. A1: "Dumbbell" Topology for inter-session fairness



Experimental Setup (Cont'd)

- Each simulation runs for 1000s
- Each source starts at a random time between 0 to 500s
- Average throughput of each flow at the last 300s is measured
- Reported fairness indices are averaged over 3 to 5 independent runs

Results – Fairness Comparison

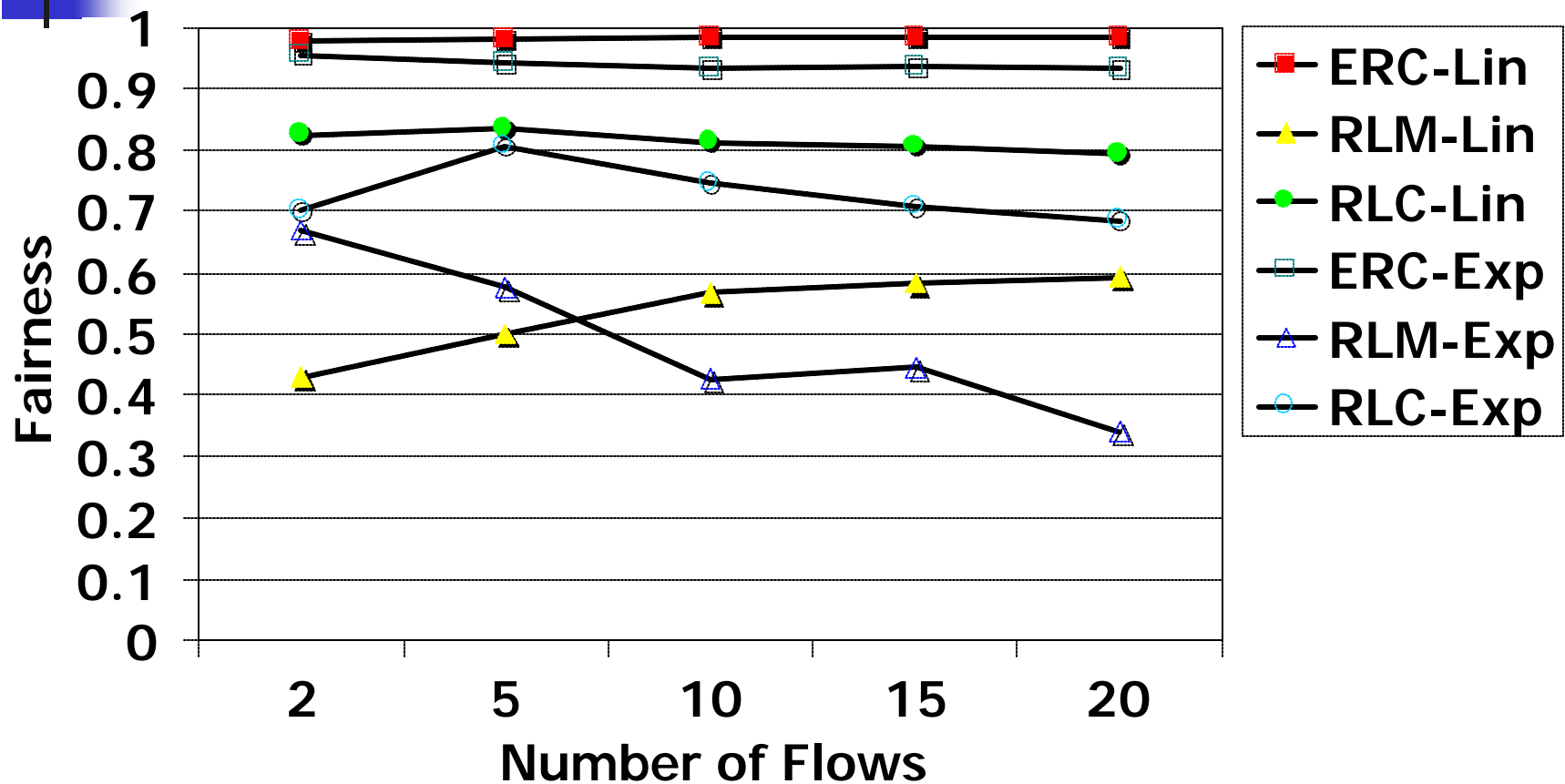


Fig. A2: Fairness comparison for ERC, RLM and RLC using Linear and Exponential layering (average over 3 runs, M fixed)

Results – Fairness of ERC

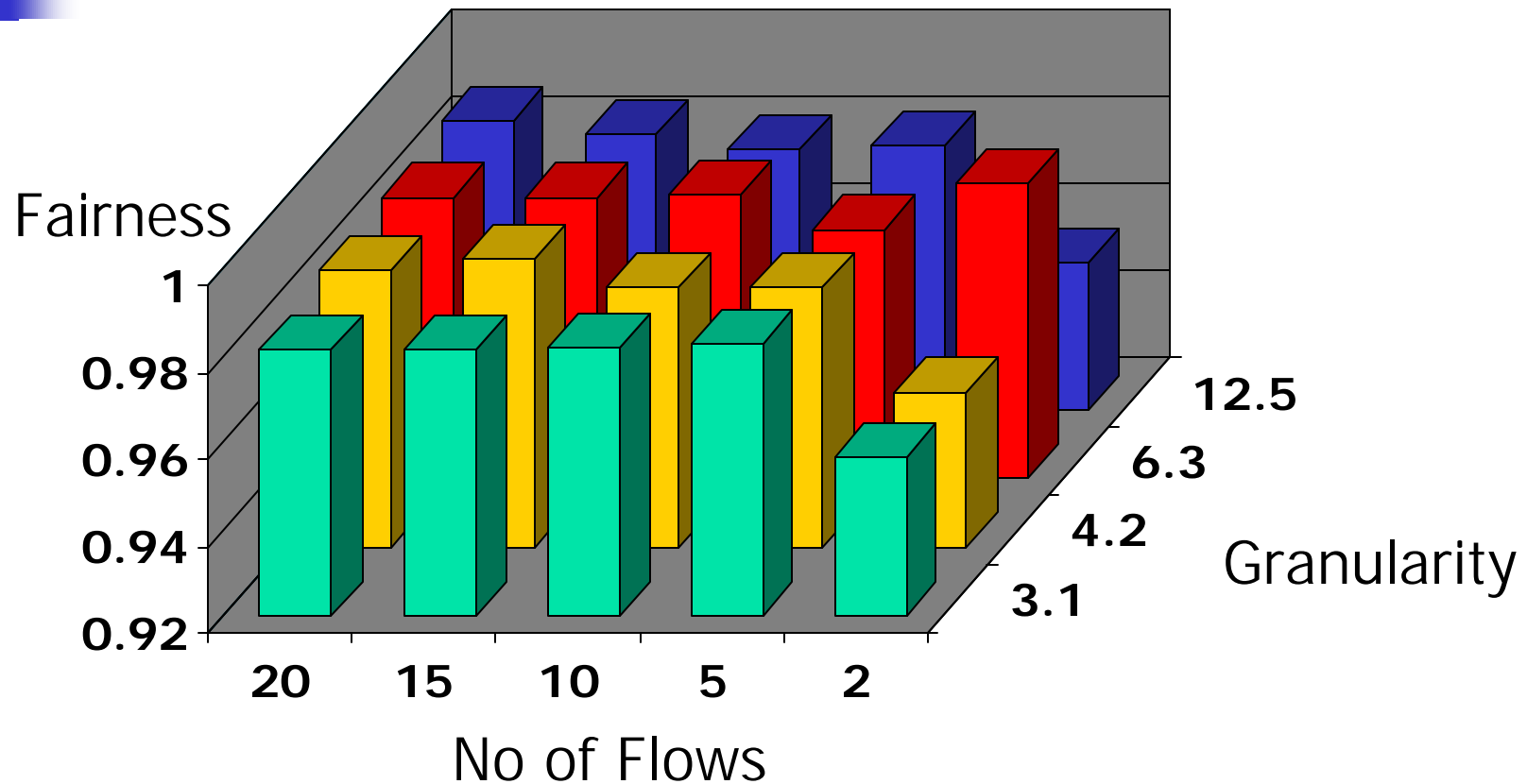


Fig. A3: Fairness of ERC with varying number of flows and granularity (Max-mean ratio fix at 3, average over 5 runs)

Results – Fairness of ERC

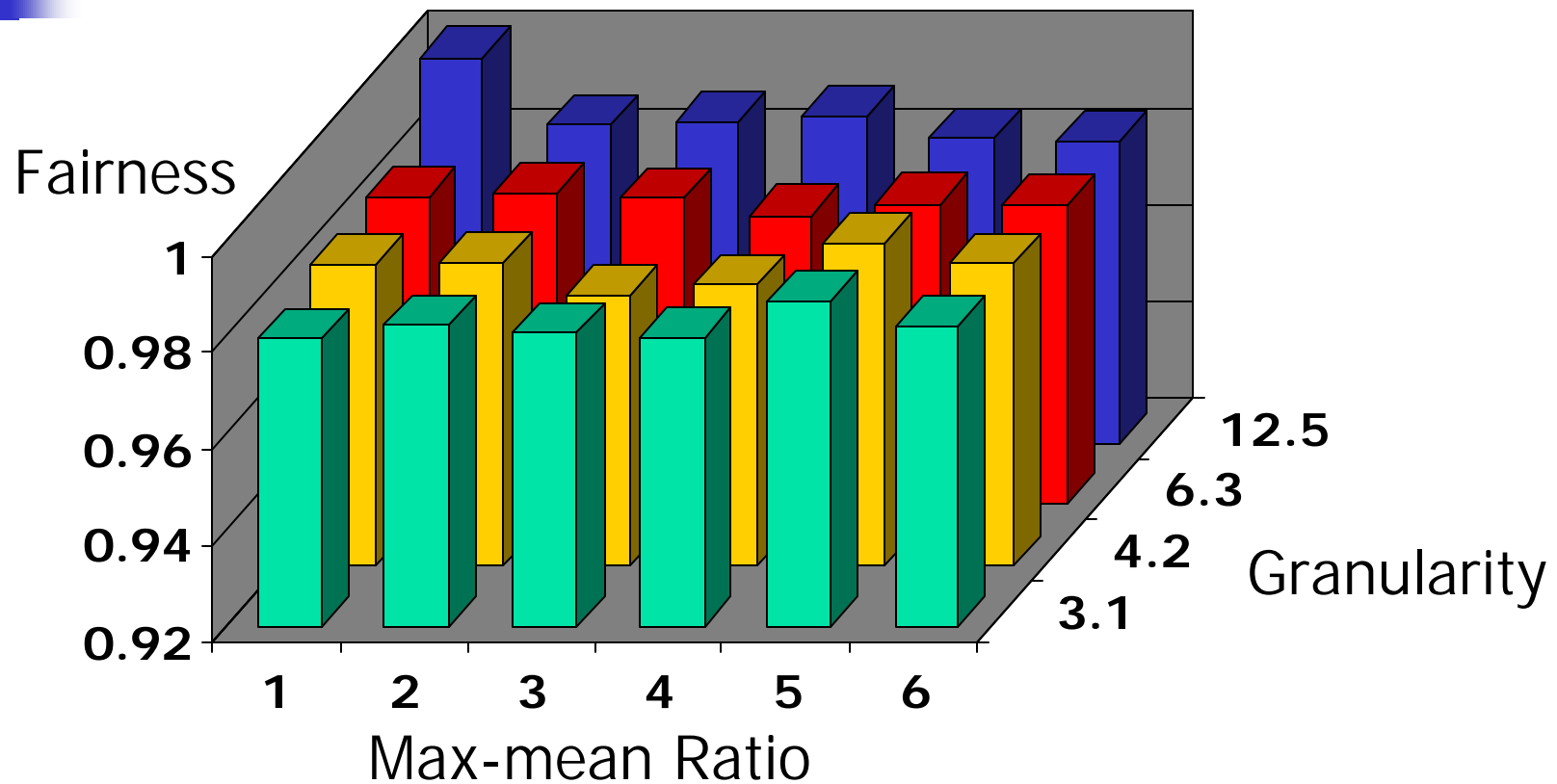


Fig. A4: Fairness of ERC with varying number of max-mean ratio and granularity (10 flows, average over 5 runs)

ERC Trace – linear

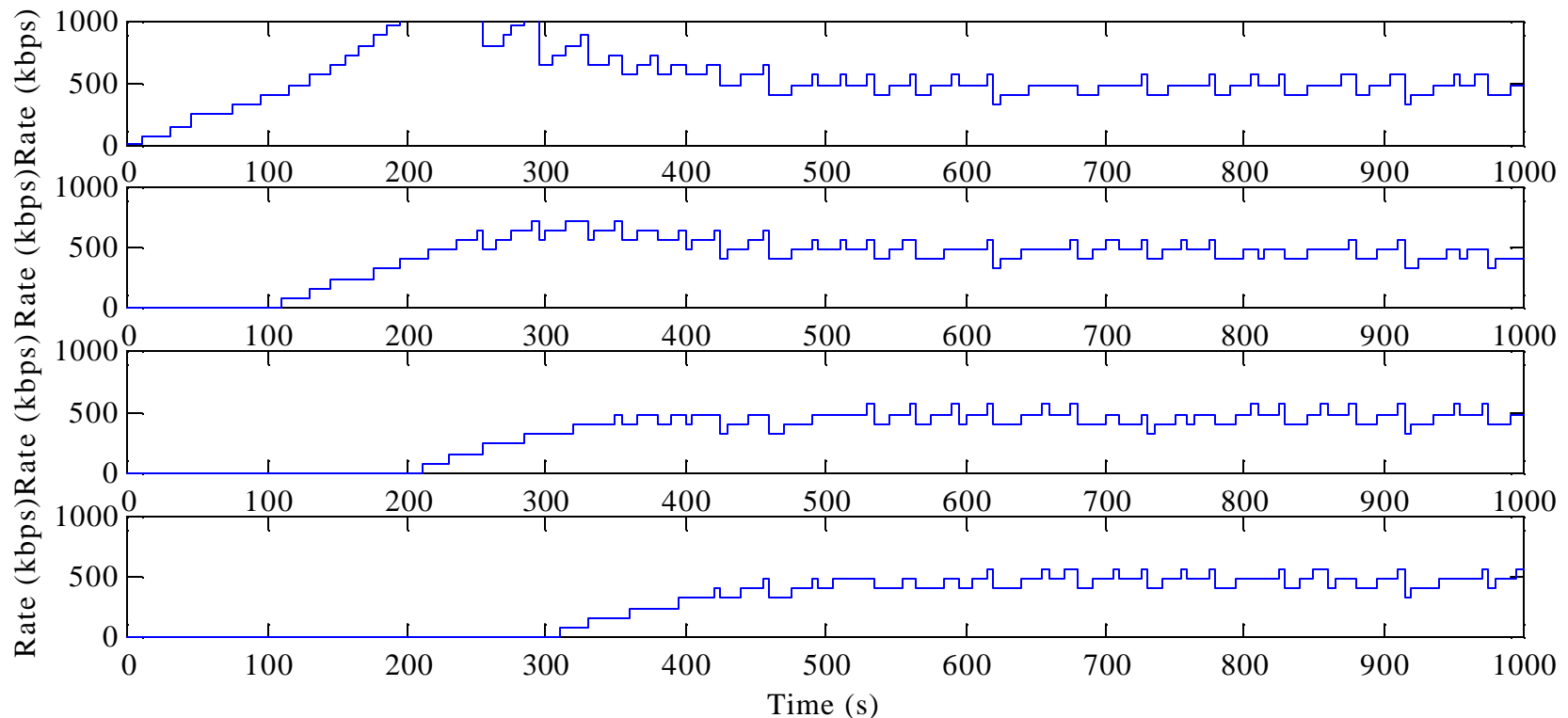


Fig. A5: Sharing of bandwidth by 4 ERC flows on “dumbbell”
Starting at time 0, 100, 200 and 300

RLM Trace – linear

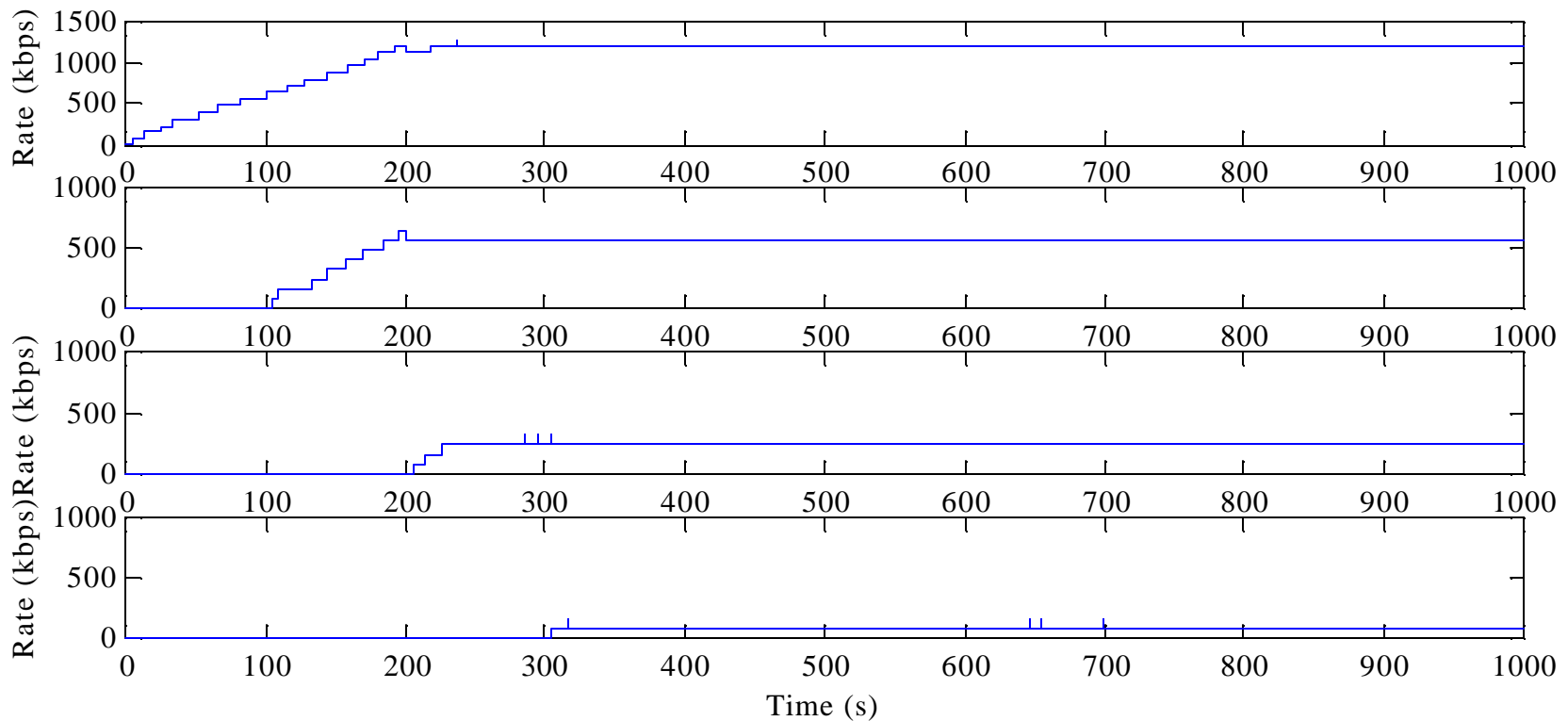


Fig. A6: Sharing of bandwidth by 4 RLM flows on “dumbbell”
Starting at time 0, 100, 200 and 300

RLC Trace – exponential

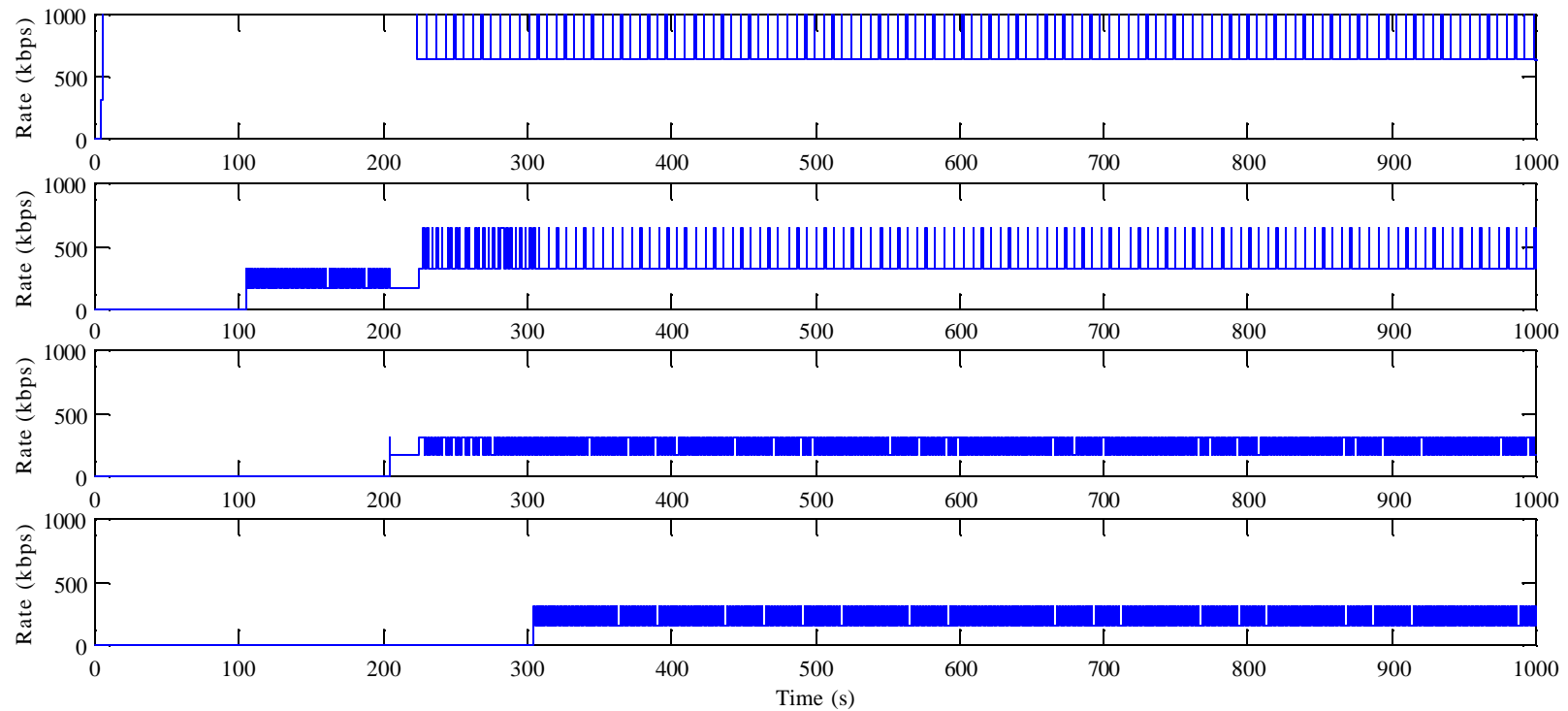


Fig. A7: Sharing of bandwidth by 4 RLC flows on “dumbbell”
Starting at time 0, 100, 200 and 300

ERC+TCP

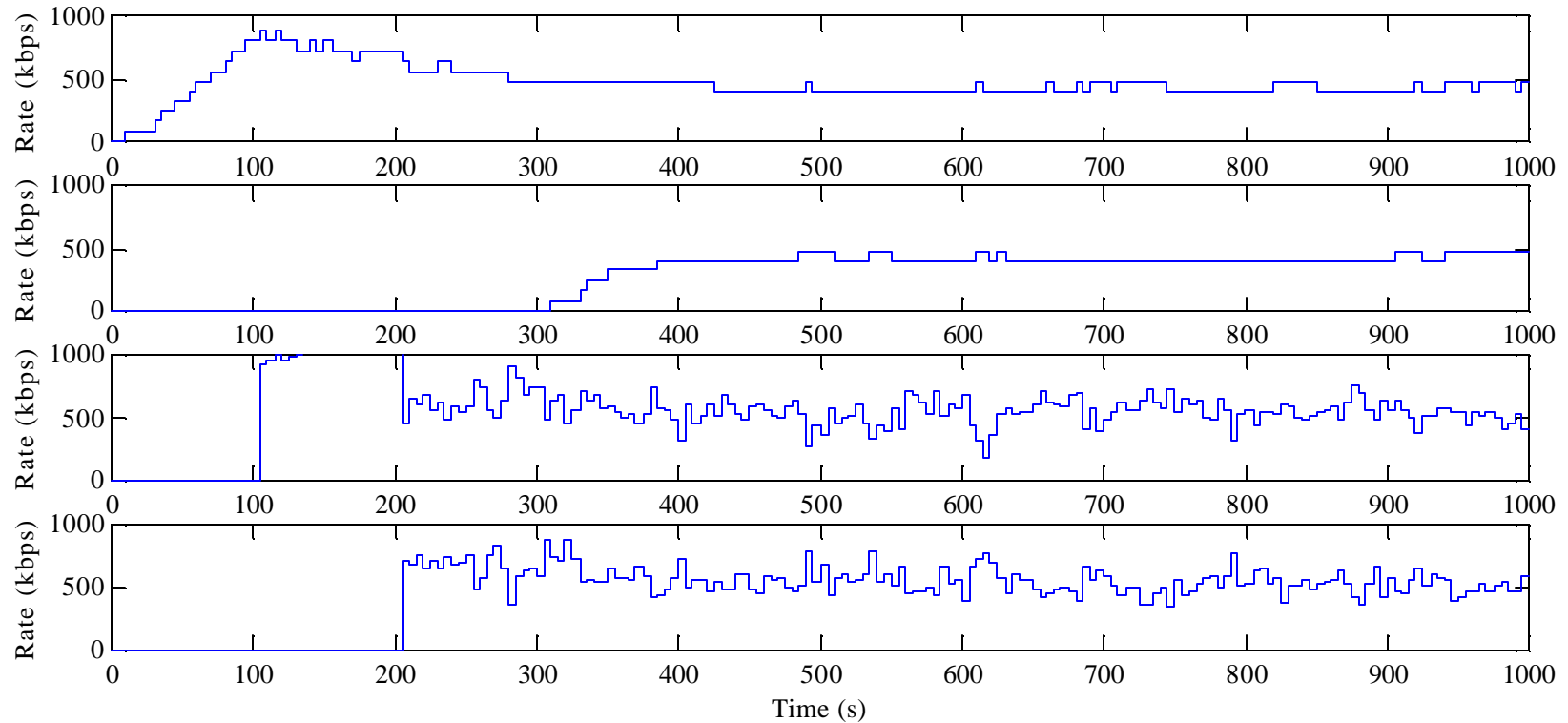


Fig. A8: Sharing of bandwidth by 2 ERC (top) and 2 TCP (bottom) flows on "dumbbell"

RLM+TCP

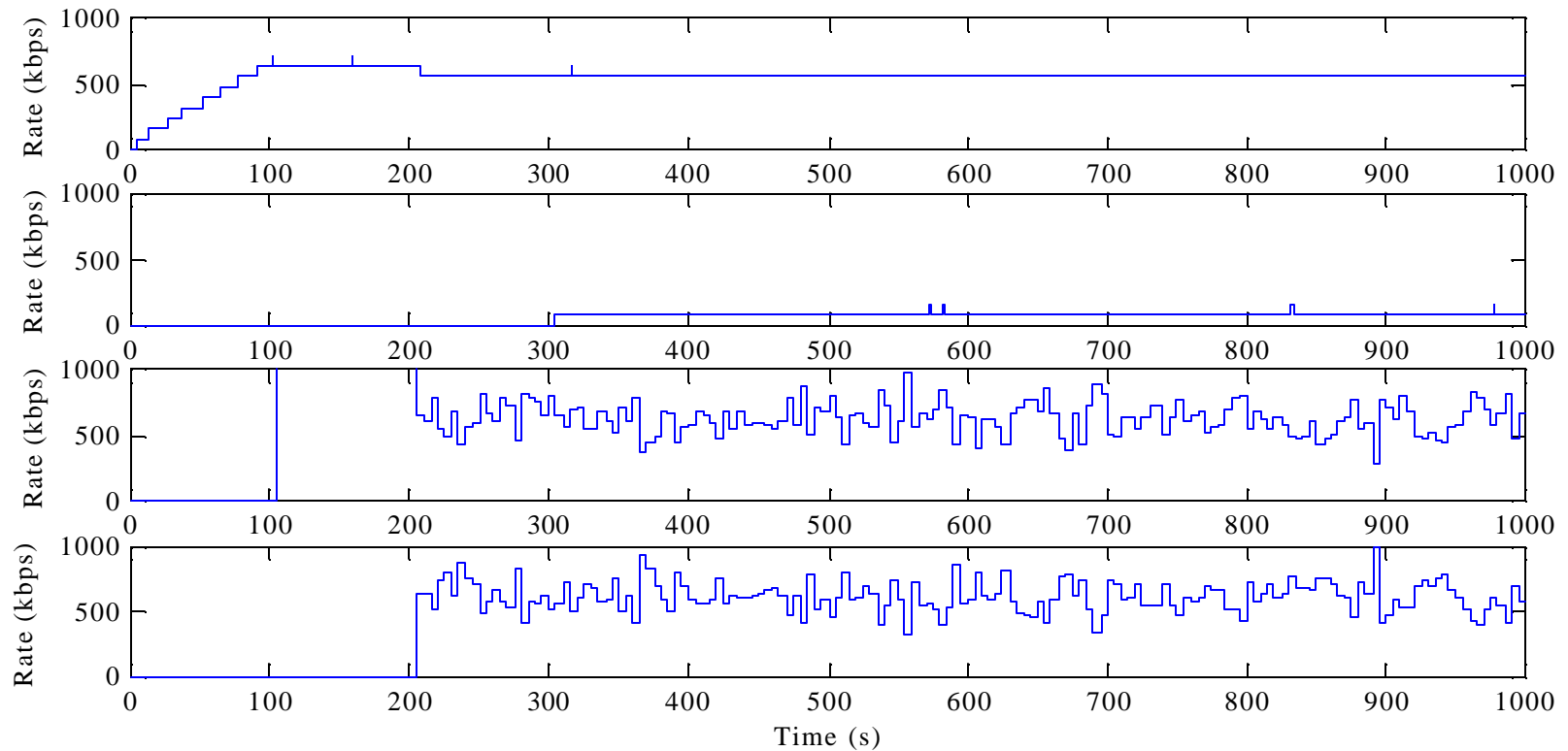


Fig. A9: Sharing of bandwidth by 2 RLM (top) and 2 TCP (bottom) flows on "dumbbell"

RLC (Exp) + TCP

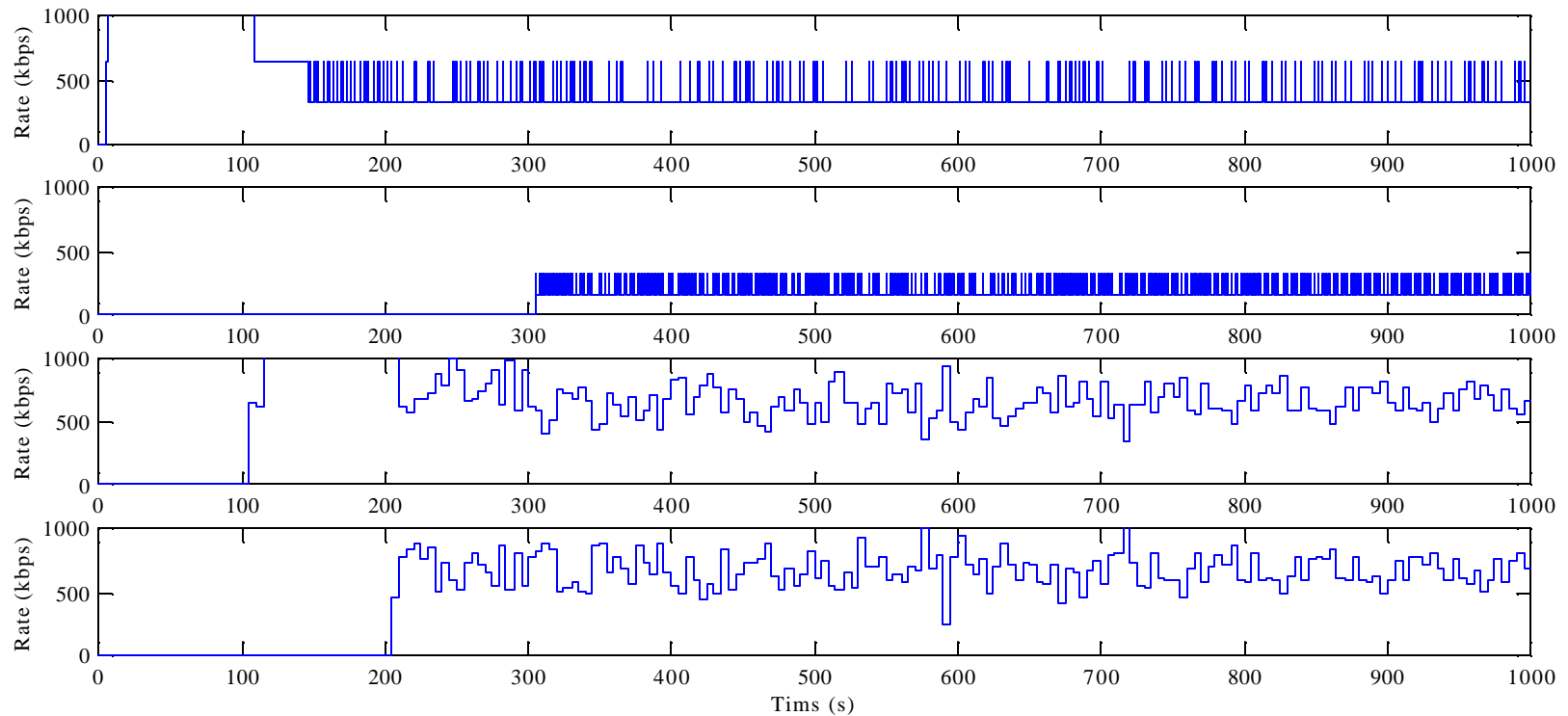


Fig. A10: Sharing of bandwidth by 2 RLC (top) and 2 TCP (bottom) flows on "dumbbell"



Choosing Equation for ERC



Equation Parameters

- Packet loss rate p
 - When p is close to 1, we wish the rate $R(p)$ to be close to 0
 - When p is close to 0, we wish the rate to be large
 - Unicast rate control: $R(p) \propto 1/\sqrt{p}$
- Round Trip Time (RTT)
 - TCP throughput proportional to $1/RTT$
 - Reduces efficiency as two receivers (in a session) with different RTT have different rate through a bottleneck

Dependence on RTT

- A single source with 4 receivers
- Compare inefficiency with and without RTT dependency on rate
- For bottleneck bandwidth b and rate R_f for flow f , inefficiency is defined as:

$$\text{mean} \left(\frac{b - R_f}{b} \right)$$

- Ideally, inefficiency ≈ 0

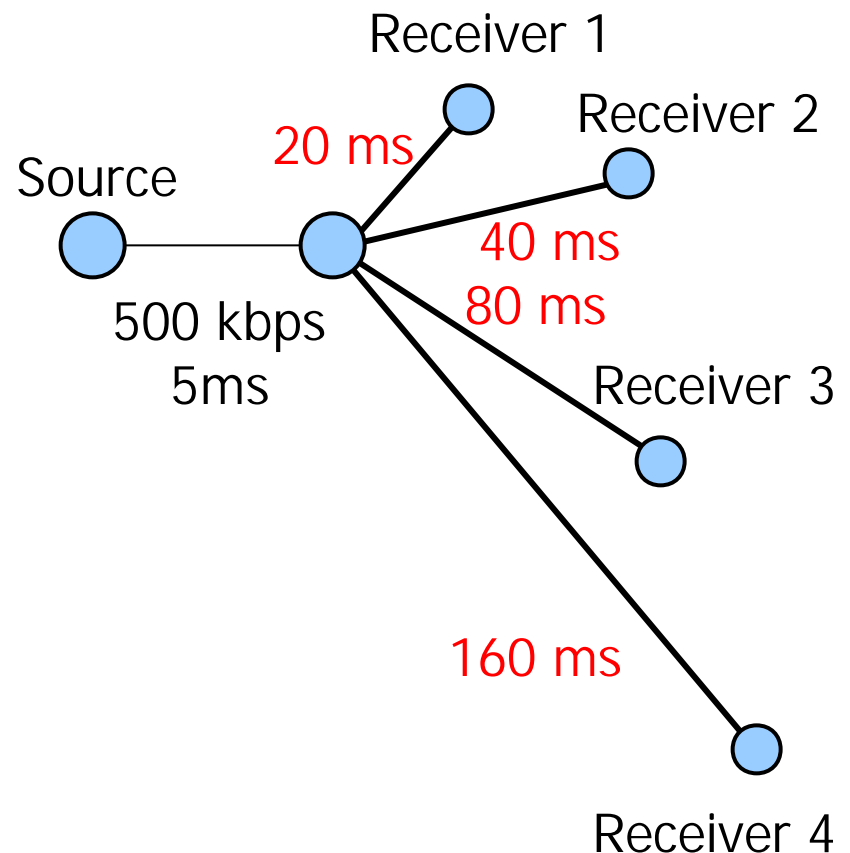


Fig. A5: "Tree" Topology



Dependence on RTT

- Using $R \mu \approx 1/(RTT \sqrt{p})$
 - Rates (kbps)
 - 493 (Receiver 1)
 - 323 (Receiver 2)
 - 182 (Receiver 3)
 - 80 (Receiver 4)
 - Inefficiency: 0.461
- Using $R \mu \approx 1/\sqrt{p}$
 - Rates (kbps)
 - 495 (Receiver 1)
 - 498 (Receiver 2)
 - 498 (Receiver 3)
 - 493 (Receiver 4)
 - Inefficiency: 0.01

Dependence on RTT

- Add simultaneous TCP connection to every receiver
- Compare throughputs between TCP and ERC flows with and without RTT dependence

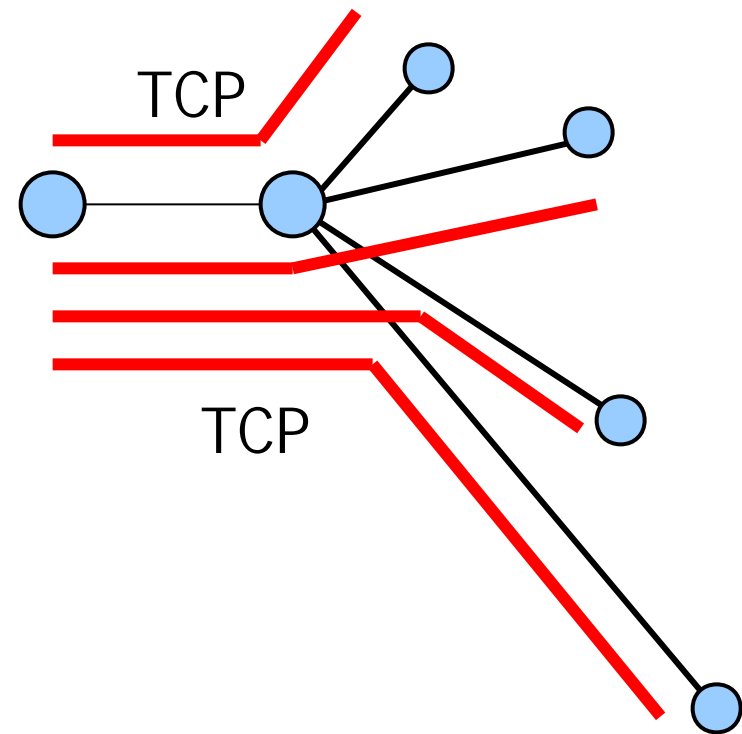


Fig. A6: Simultaneous transmission of ERC and TCP on "Tree" Topology



Dependence on RTT

- Using $R \propto 1/(\text{RTT} \sqrt{p})$
 - TCP-Rate/ERC-Rate
 - 0.95 (Receiver 1)
 - 1.10 (Receiver 2)
 - 0.79 (Receiver 3)
 - 1.21 (Receiver 4)
 - Max/Min ≈ 1.53
- Using $R \propto 1/\sqrt{p}$
 - TCP-Rate/ERC-Rate
 - 0.92 (Receiver 1)
 - 0.65 (Receiver 2)
 - 0.32 (Receiver 3)
 - 0.1 (Receiver 4)
 - Max/Min ≈ 9.2