

# Multi-layer Active Queue Management and Congestion Control for Scalable Video Streaming

Seong Kang, Yuping Zhang,  
Min Dai, and Dmitri Loguinov

Texas A&M University  
College Station, TX 77843

# Overview

- Motivation
- FGS background
- Analysis of video streaming
- Preferential streaming framework
- MKC congestion control
- Conclusion

# Motivation

- Video streaming is an important part of the existing Internet
- To offer a high-quality streaming environment to end-users, many video applications require network QoS
- Many proposals attempt to provide some form of network QoS
  - DiffServ for “better-than best-effort” performance to end flows
  - AQM for QoS service with much less overhead than DiffServ

# Motivation 2

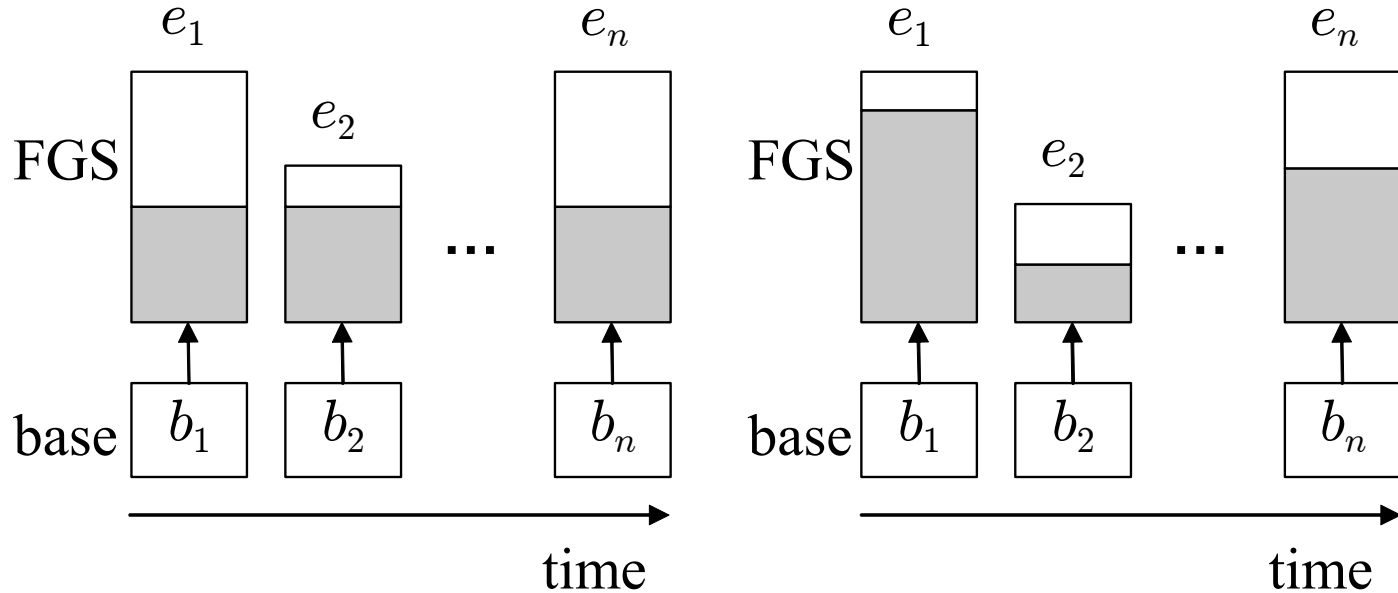
- However, existing work does not provide scalable, low-overhead, low-delay, and retransmission-free platform
- Our work aims to fill this void

# FGS Background

- FGS is the streaming profile of the ISO/IEC MPEG-4 standard
- Method of compressing residual video signal into a single enhancement layer
- Allows application servers to scale the enhancement layer to match variable network capacity during streaming
- Typically coded at some fixed bitrate and can be rescaled to any desired bitrate

# FGS Background 2

- Scaling of MPEG-4 FGS using fixed-size (left) and variable-size (right) frames



# Analysis of Video Streaming

- We investigate the performance of video streaming using MPEG-4 FGS as an example
- Consider best-effort streaming based on independent Bernoulli loss
- Lemma 1: Given long-term network packet loss  $p$ , the expected number of useful packets recovered per frame is:

$$E[Z_j] = \frac{1-p}{p} \sum_{k=1}^{\infty} (1 - (1-p)^k) q_k$$

– where  $q_k$  is a PMF of frame sizes

# Analysis of Video Streaming 2

- Note that exact distribution of frame sizes is application-specific
  - It depends on coding scheme, frame rate, variation in scene complexity, and bitrate of the sequence
- When all frames have the same fixed size  $H$ , the expectation becomes:

$$E[Z_j^H] = \frac{1-p}{p} (1 - (1-p)^H)$$



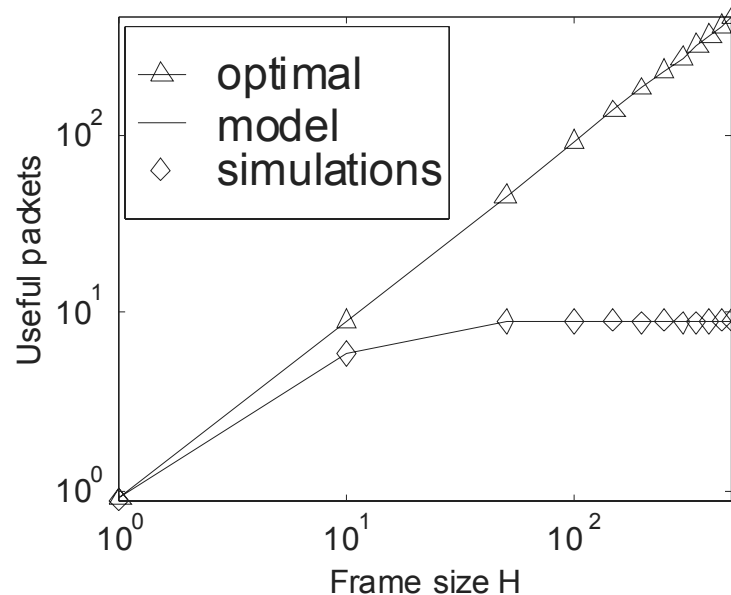
# Analysis of Video Streaming 3

- The model is compared to simulation results:

Packet loss $p$	$H = 100$		$H = 1,000$	
	Simulations	Model	Simulations	Model
0.0001	99.49	99.49	950.61	951.57
0.001	95.06	95.06	630.46	631.67
0.01	62.60	62.57	98.88	98.99
0.1	8.98	8.99	9.04	9.00
0.2	4.01	4.00	4.03	4.00
0.9	0.10	0.11	0.11	0.11

# Analysis of Video Streaming 3

- $H$  becomes larger, the expectation tends to  $(1 - p) / p$ 
  - This means that the recovered useful percentage of each frame tends to zero
- The following simulation results shows this:



# Analysis of Video Streaming 4

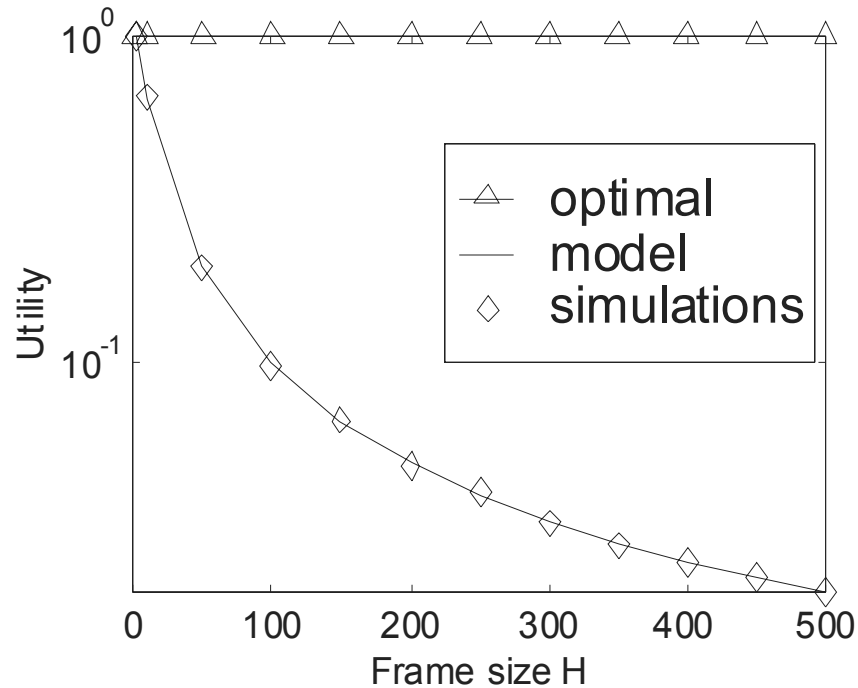
- Next we analyze *Utility* of received video
- Define the utility as following:

$$U = \frac{E[Z_j^H]}{H(1-p)} = \frac{1 - (1-p)^H}{pH}$$

- $U$  drops zero inverse proportionally to  $H$
- This means the decoder receives “junk” data with probability 1 as sending rates become higher

# Analysis of Video Streaming 5

- Simulation results of  $U$  for  $p = 0.1$



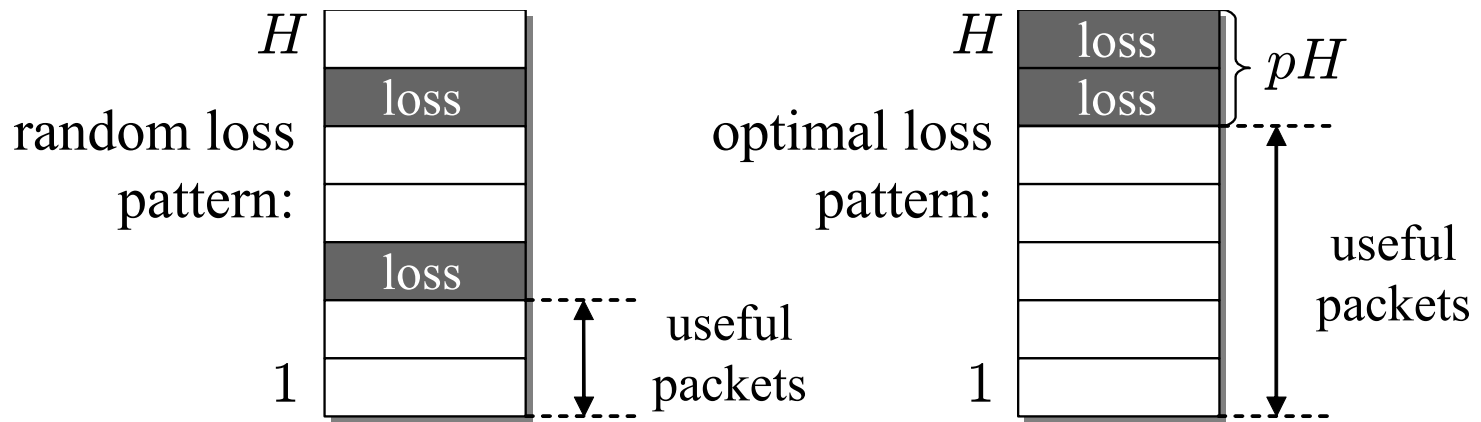
- For example,  $U = 0.1$  for  $p = 0.1$  and  $H = 100$

# Analysis of Video Streaming 6

- Next, discuss “optimal” streaming that can provide maximum end-user utility (i.e.,  $U = 1$ )
- Given packet loss  $p$ , how can we achieve the optimal utility ?
- Recall that consecutive lower portions of the FGS layer can enhance the base layer
  - Any gaps in the delivered data typically render the remainder of the layer useless
- Thus, to achieve optimality, routers must drop the upper parts of the FGS layer during congestion

# Analysis of Video Streaming 7

- The difference between ideal and random drop patterns:



- The main question now is whether optimal streaming is possible in practice and how to achieve it
- We address this issue next

# Preferential Video Streaming

- We introduce a new video streaming mechanism called *Partitioned Enhancement Layer Streaming* (PELS)
- Operates in conjunction with priority-queuing AQM routers in network paths
- Applications
  - partition the enhancement layer into two layers,
  - mark their packets using different priority classes
- Routers discriminate between packets based on their priority
  - No per-flow management is required

# Preferential Video Streaming 2

- The PELS framework consists of three parts
  - Router queue management
  - FGS partitioning and packet coloring
  - Selection of  $\gamma$  (fraction of the lowest priority section of FGS layer)

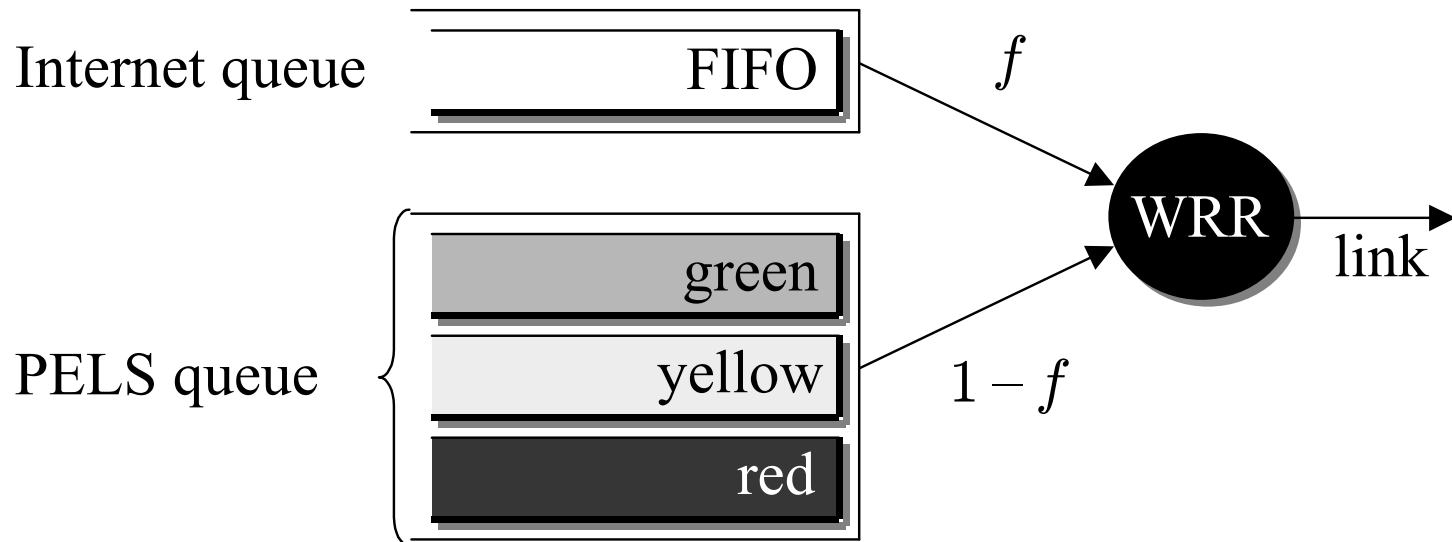


# Router Queue Management

- PELS architecture maintain two types of queues to separate video traffic from the rest of flows
  - Internet queue – FIFO
  - PELS queue
    - Subdivided to green, yellow, and red queues
    - *Use strict* priority discipline
- We employ weighted round-robin (WRR) scheduling between the PELS and Internet queues
  - Ensure fair share of network bandwidth between PELS flows and other Internet traffic
  - Allows de-centralized administrative flexibility

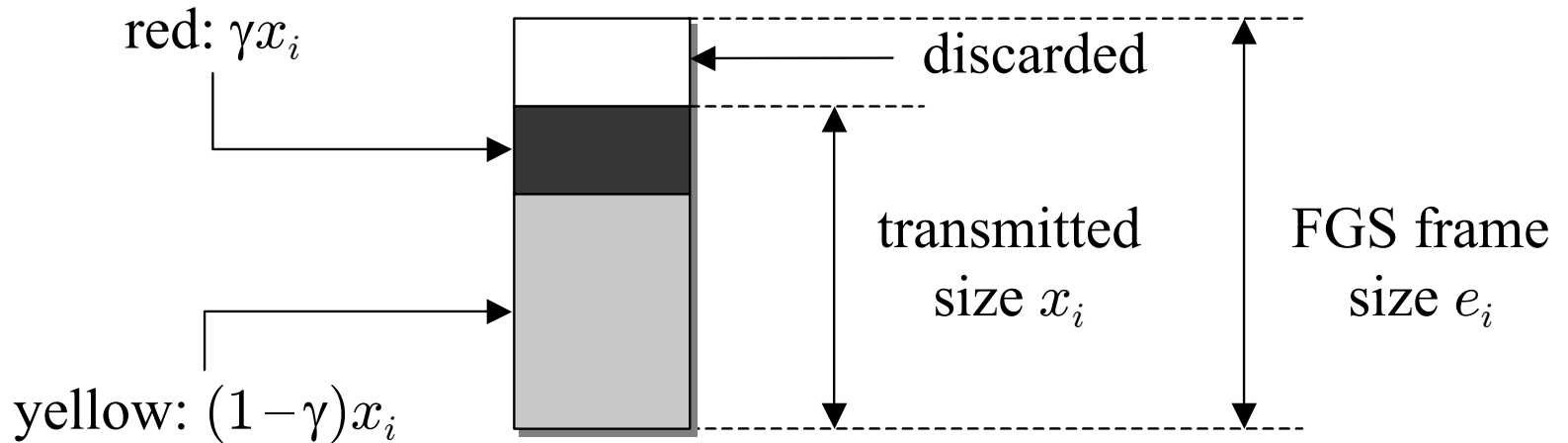
# Router Queue Management 2

- Internet and PELS queues served by WRR with weights  $f$  and  $1 - f$ :



# FGS Partitioning and Packet Coloring

- Next we show one possible partitioning of FGS layer:



- FGS bytes divided into two priority classes (yellow and red):

# FGS Partitioning and Packet Coloring 2

- In an ideal network with stationary packet loss  $p$ , server can select  $\gamma$  such that  $\gamma x_i$  is equal to  $p x_i$ 
  - This ensure that all red packets are lost
  - $(1 - p)$  yellow packets are recovered for decoding
  - This is the best scenario under any circumstances
- In practice, however, any slight increase in  $p$  creates a best-effort FIFO situation for yellow packets
- Thus, proper and dynamic selection of  $\gamma$  is important

# Selection of $\gamma$

- $\gamma$  should be adjusted according to packet loss
  - keep the resulting red loss at a certain threshold  $p_{thr}$
  - Increase  $\gamma$  when  $p$  increase and decrease it as  $p$  decrease
- We use a proportional controller:
$$\gamma_i(k) = \gamma_i(k-1) + \sigma (p_i(k-1)/p_{thr} - \gamma_i(k-1))$$
  - adjusts  $\gamma$  based on measured packet loss and target red packet loss  $p_{thr}$
- Lemma 2: This controller is stable *iff*  $0 < \sigma < 2$

# Selection of $\gamma$ 2

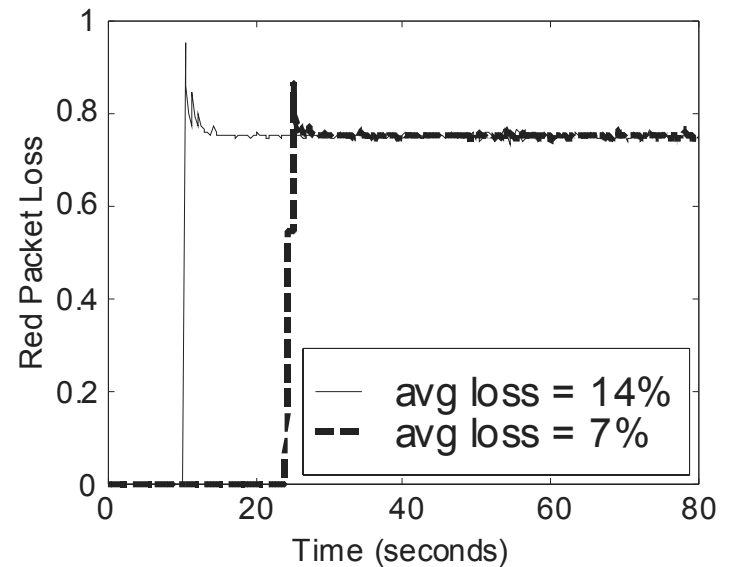
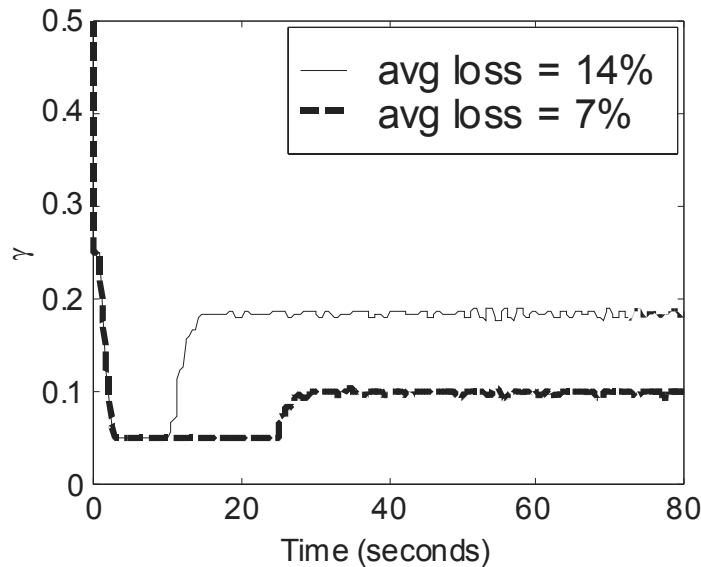
- Assuming arbitrary round-trip delay  $D_i$ , we have the following:

$$\gamma_i(k) = \gamma_i(k - D_i) + \sigma (p_i(k - D_i) / p_{thr} - \gamma_i(k - D_i))$$

- Lemma 3: This is also stable *iff*  $0 < \sigma < 2$
- Next, we derive the effect that both controllers have on the packet loss in the red queues
- Lemma 4: Assuming stationary packet loss  $p$ , both controllers converge red packet loss  $p_R$  to  $p_{thr}$

# Selection of $\gamma$ 3

- The evolution of  $\gamma$  (left) and corresponding red loss rates  $p_R$  (right)



# Utility of PELS

- The utility of received video in PELS is lower-bounded by the following:

$$U \geq \frac{1 - p/p_{thr}}{1 - p}$$

- For example,  $U \geq 0.96$  for  $p = 0.1$  and  $p_{thr} = 0.75$
- This shows that although PELS does not achieve “optimality” but comes very close to it



# MKC Congestion Control

- Congestion control is necessary for streaming applications to provide a high level of video quality to end users
- We study Kelly controls as an example of one possible scheme that supplements PELS
  - Note that PELS is *independent* of congestion control
  - PELS can be utilized with any end-to-end or AQM scheme
- However, the classical discrete Kelly control shows stability problem when feedback delay is large (Zhang *et al.* [34])

# MKC Congestion Control 2

- Thus, we introduce modified max-min Kelly control (MKC):

$$r_i(k) = r_i(k - D_i) + \alpha - \beta r_i(k - D_i) p_l(k - D_i^{\leftarrow})$$

- Packet loss  $p_l$  is fed back from the *most-congested* router  $l$ :

$$p_l(k) = \frac{\sum_{j \in S_l} r_j(k - D_j^{\rightarrow}) - C_l}{\sum_{j \in S_l} r_j(k - D_j^{\rightarrow})}$$

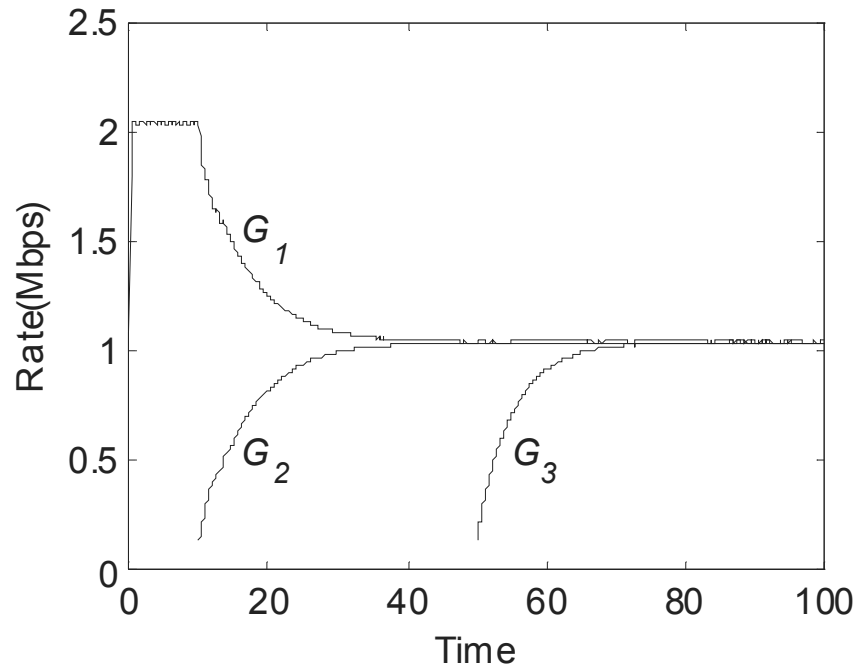
# MKC Congestion Control 3

- This provides max-min fairness instead of proportional resource allocation
- Lemma 5: MKC is stable under heterogeneous delays *iff*  $0 < \beta < 2$
- Lemma 6: Regardless of feedback delay, each flow reaches the stationary rate:

$$r_i^* = \frac{C_l}{N} + \frac{\alpha}{\beta}$$

# MKC Congestion Control 4

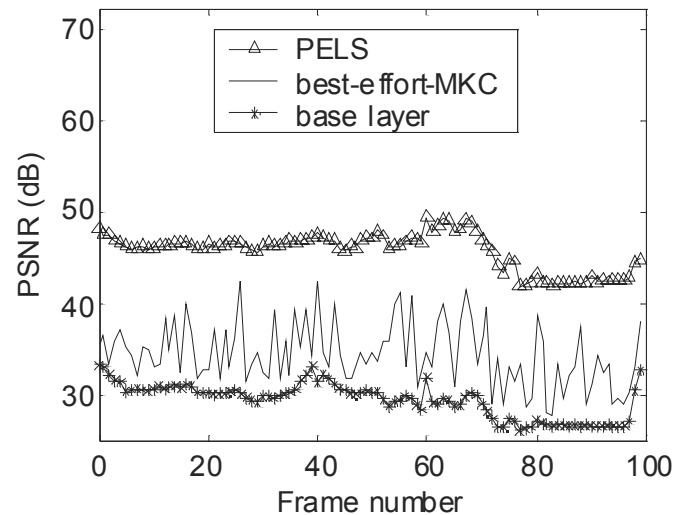
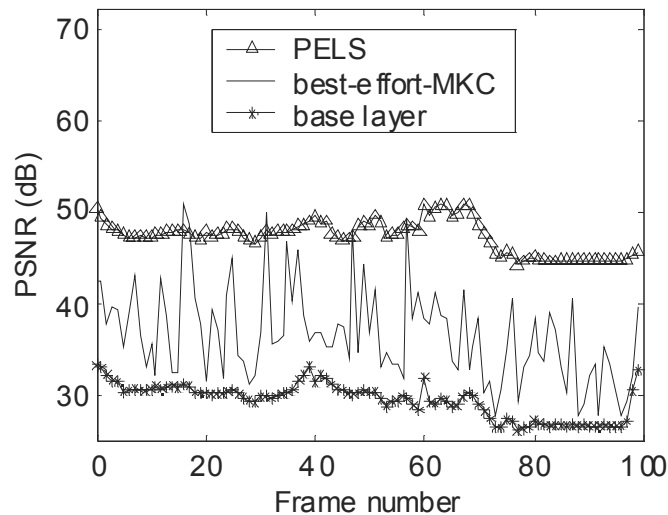
- Convergence of MKC and max-min fairness



- Notice that proportional fairness would favor flows  $G_2$  and  $G_3$  by giving them double the rate of  $G_1$

# PSNR Quality Evaluation

- PSNR of Forman sequence reconstructed with 10% (left) and 19% (right) packet loss



- PELS enhances base-layer PSNR by 60% ( $p = 0.1$ ) and 55% ( $p = 0.19$ ) while best-effort-MKC improves it by 24% and 16% on average, respectively

# Conclusion

- Best-effort streaming is far from optimal
  - We proved this using Bernoulli loss model
  - Stochastic model of arbitrary packet loss and its effect will be presented in a future paper
- Our results in the current paper indicate that PELS offer an appealing framework for video streaming
  - Provide provably optimal utility to future multi-media applications
  - Low-overhead and scalable AQM