# Distributed Video Systems

Chapter 6

Issues in Video Transmission and Delivery

Part 2 - Error Control and Recovery

Jack Yiu-bun Lee
Department of Information Engineering
The Chinese University of Hong Kong

## Contents

## 5.1 Introduction



- Types of Errors in Network Communications
  - Data corruption
    - received data is not the same as the one sent;
  - Data loss
    - transmitted data are not received by the receiver.
- Problems in Network Error Control
  - Error Detection
    - How to detect an error occurs, or a network packet is lost?
  - Error Recovery
    - How to correct data corruption, or recover a lost packet?
- Tradeoffs
  - Bandwidth, delay, and buffer!



## 5.2 Existing Retransmission-Based Schemes

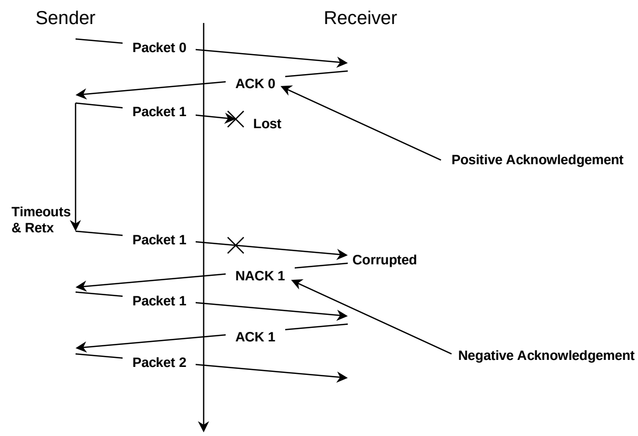


- Network Architecture
  - Full-duplex
- Error Detection
  - Error-detecting Codes
    - Checksum, CRC codes;
    - Suitable for detecting data corruption.
  - Timer
    - Assume loss if a packet does not arrive after a specific deadline time.
  - Sequence
    - Assume loss if packets arrive out-of-sequence.
- Error Recovery
  - Retransmit the lost/corrupted packet.

## 5.2 Existing Retransmission-Based Schemes

Jack Y.B. Lee

- Stop-and-Wait Automatic Repeat Request (SW-ARQ)
  - Transmission Scenario:

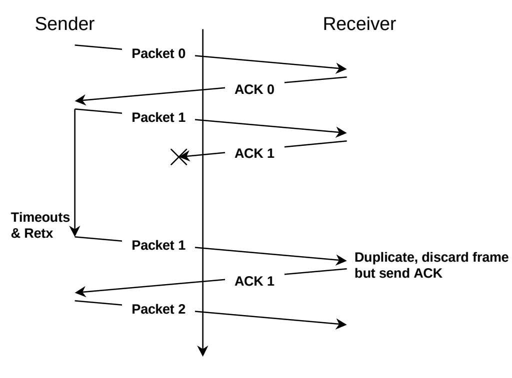





## 5.2 Existing Retransmission-Based Schemes

Jack Y.B. Lee

- Stop-and-Wait Automatic Repeat Request (SW-ARQ)
  - Transmission Scenario:
    - ACK loss



## 5.2 Existing Retransmission-Based Schemes



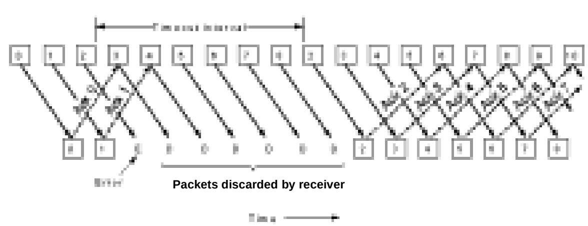
- Stop-and-Wait Automatic Repeat Request (SW-ARQ)
  - Advantages
    - No extra buffering needed at both sender and receiver.
      - Sender keeps the current transmitting packet only;
      - Receiver needs one buffer to receive the incoming packet.
  - Disadvantages
    - Very poor performance in case of packet loss/corruption.
      - Sender timeout is usually long;
      - No packet is transmitted during propagation of data and control (ACK, NACK) packets.
  - Applications
    - Suitable for networks having negligible loss/corruption rates (e.g. interprocess communications).



## 5.2 Existing Retransmission-Based Schemes



- Go-Back-*N* Automatic Repeat Request (GBN-ARQ)
  - Transmission Scenario:

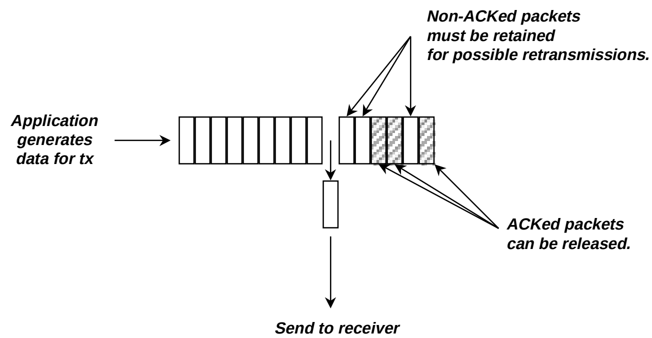



  - Improvements
    - Less sensitive to propagation delay since more than one transmitted packets can be outstanding.

## 5.2 Existing Retransmission-Based Schemes

Jack Y.B. Lee

- Go-Back-*N* Automatic Repeat Request (GBN-ARQ)
  - Tradeoffs
    - Extra buffering required at the sender (but not receiver).

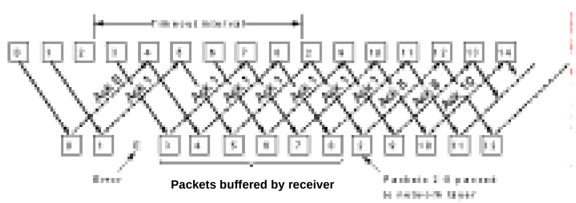





## 5.2 Existing Retransmission-Based Schemes

Jack Y.B. Lee

- Selective-Repeat ARQ (SR-ARQ)
  - Transmission Scenario:

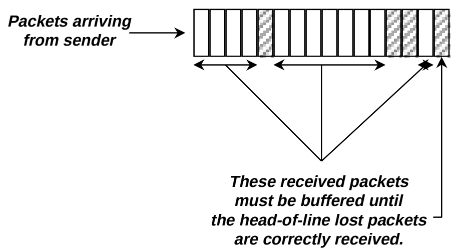



  - Improvements
    - Better throughput as duplicate retransmissions after an error are avoided.
  - Tradeoffs
    - Extra buffering required at both sender and receiver.

## 5.2 Existing Retransmission-Based Schemes

Jack Y.B. Lee

- Selective-Repeat ARQ (SR-ARQ)
  - Tradeoffs
    - Extra buffering required at both sender and receiver.




- In case of buffer overflow, the receiver will have to stop receiving packets. Possibly switching back to GBN-ARQ until the occupied buffers are released.



## 5.3 Existing Forward-Error-Correction Schemes

Jack Y.B. Lee

- Network Architecture
  - Only half-duplex is needed (sender-to-receiver).
- Error Detection and Recovery
  - Error-Correcting Codes
    - Parity, Reed-Solomon Codes;
    - Can detect as well as correct data corruption.
  - Erasure-Correcting Codes
    - Parity, Reed-Solomon Codes;
    - Can recover packet losses.

- Assumptions
  - System Dimensions:
    - One server, $N$ video clients.
  - Server-push service model:
    - Packet size $Q_S$ bytes, video bit-rate $R$ Bps.
    - Inter-packet-transmission time $T_S=Q_S/R$ seconds.
  - Constant-Bit-Rate (CBR) Video Service:
    - Consumption time of a video packet by the decoder is constant and equals to $T_S$ seconds.
  - Network conditions for client $i$ ($0 \le i < N$):
    - Probability of packet loss is independent and is $p_i$;
    - Average delay between client and server = $T_i$;
    - Delay jitter bounds are $T_i^+$ and $T_i^-$;





- Assumptions
  - Video quality requirement:
    - Playback continuity;
    - Maximum tolerable loss rate = $p_{max}$.
  - Client buffer management:
    - Each buffer stores one video packet;
    - Together, buffers are organized as a circular buffer.
  - Retransmission Scheme
    - SR-ARQ
- Performance Metrics
  - Network traffic overhead incurred in error recovery, excluding control traffics.
  - Client buffer requirement, which directly affects the startup delay and system response time.

## 5.4 Analyzing SR-ARQ for Video Delivery

Jack Y.B. Lee

- Video Packet Arrival Pattern
  - Let $T_{start}$ be the time the server sends the first video packet for the video session.
  - The arrival time at receiver *i* for packet *j*, denoted by $A_i^j$, is bounded by

$$\max\{T_{start} + \underbrace{T_{tran}}_{\text{Transmission time}} + \underbrace{jT_S}_{\text{time to send packet } j} + \underbrace{T_i + T_i^-}_{\text{propagation delay}}, T_{start}\} \le A_i^j \le T_{start} + T_{tran} + jT_S + T_i + T_i^+ \qquad (1)$$

Transmission time   time to send packet *j*   propagation delay



## 5.4 Analyzing SR-ARQ for Video Delivery

Jack Y.B. Lee

- Traffic Overhead
  - Residual packet loss probability after $K_i$ transmission attempts:

$$1 - \sum_{j=1}^{K_i} \left( p_i^{j-1} (1 - p_i) \right) = p_i^{K_i} \qquad (2)$$

  - To maintain video playback quality, we need to choose $K_i$ such that the loss limit is not exceeded:

$$p_i^{K_i} \le p_{\max} \qquad (3)$$

  - Rearranging, we can then obtain $K_i$ from

$$K_i \ge \left\lceil \frac{\ln p_{\max}}{\ln p_i} \right\rceil \qquad (4)$$

- Traffic Overhead
  - Given $K_i$, the expected number of transmissions for each video packet can be obtained from

$$\sum_{j=1}^{K_i} j\left(p_i^{j-1}(1-p_i)\right) \qquad (5)$$

  - Hence, the ratio of extra traffic overhead incurred (excluding the first transmission) in retransmission for receiver $i$ is given by

$$h_i = \frac{\text{traffic for retransmission}}{\text{traffic for data}} = \frac{\sum_{j=1}^{K_i} (j-1)\left(p_i^{j-1}(1-p_i)\right)}{1} \qquad (6)$$





- Traffic Overhead
  - Finally, the traffic overhead at the server link is just the sum of traffic overhead for all receivers:

$$H_{ARQ} = \sum_{i=0}^{N-1} h_i \qquad (7)$$

or

$$H_{ARQ} = \sum_{i=0}^{N-1} \sum_{j=1}^{K_i} (j-1)\left(p_i^{j-1}(1-p_i)\right)$$

## 5.4 Analyzing SR-ARQ for Video Delivery



- Client Buffer Requirement
  - Buffer Management
    - Let $L_{ARQ}(i)=Y_i+Z_i$ be the number of buffers (each $Q_S$ bytes) in receiver *i*.
    - The receiver starts playback once $Y_i$ buffers are filled with video data.
    - These $Y_i$ prefetched buffers are then used to absorb delay variations in packet arrivals to prevent video playback starvation (i.e. buffer underflow).
    - On the other hand, we reserve $Z_i$ empty buffers to cater for early-arrival packets to prevent buffer overflow.



## 5.4 Analyzing SR-ARQ for Video Delivery



- Client Buffer Requirement
  - Under this model, video playback effectively starts at time $A_i^{Y_1-1}$, which is the time video packet ($Y_i$–1) arrives at receiver *i*.
  - Hence the playback time for video packet *j* of receiver *i* is:

$$P_i^j = A_i^{Y_i-1} + jT_S \quad (8)$$

  - To maintain video playback continuity, we must ensure that all video packets arrive before playback deadline.
    - Formally, the arrival time for a video packet must be earlier than the playback time:

$$A_i^j \le P_i^j \quad \forall j \quad (9)$$

- Client Buffer Requirement
  - Substituting the upper bound for $A_i^j$ and the lower bound for $P_i^j$ into (9), we can then obtain the condition for continuity as

$$\begin{aligned}\max\{A_i^j\} &\le \min\{P_i^j\}\\ T_{start}+T_{tran}+jT_S+T_i+T_i^+ &\le \min\{A_i^{Y_i-1}+jT_S\}\\ &= \min\{A_i^{Y_i-1}\}+jT_S\\ &= T_{start}+T_{tran}+(Y_i-1)T_S+T_i+T_i^-+jT_S\end{aligned}$$

(10)

  - Rearranging we can obtain $Y_i$ as

$$Y_i \ge \frac{T_i^+ - T_i^-}{T_S}+1$$ (11)





- Client Buffer Requirement
  - Similarly, to prevent buffer overflow, we need to ensure that an empty buffer is always available when a new video packet arrives.
  - As the receiver buffers are managed as a circular buffer, we need to ensure that

$$\min\{A_i^{j+Y_i+Z_i-1}\} \ge \max\{P_i^j+T_S\}$$
$$T_{start}+T_{tran}+(j+Y_i+Z_i-1)T_S+T_i+T_i^- \ge T_{start}+T_{tran}+(j+Y_i)T_S+T_i+T_i^+$$

(12)

  - Rearranging, we can than obtain $Z_i$:

$$Z_i \ge \frac{T_i^+ - T_i^-}{T_S}+1$$ (13)

## 5.4 Analyzing SR-ARQ for Video Delivery
Jack Y.B. Lee

- Client Buffer Requirement
  - Hence, the total number of receiver buffers needed for receiver *i* is:

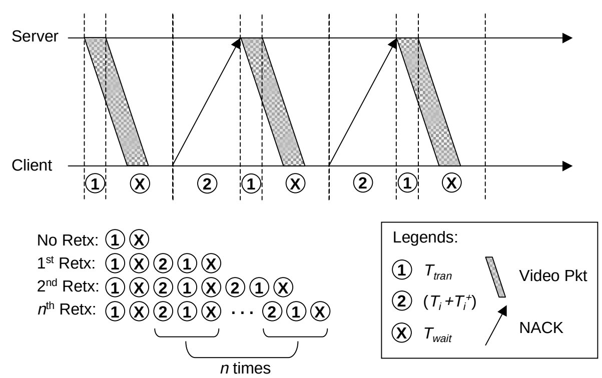
$$L_{ARQ}(i) = 2\left\lceil \frac{\left(T_i^+ - T_i^-\right)}{T_S} \right\rceil + 2 \qquad (14)$$

  - This is the amount of buffer required to absorb network delay and delay jitters to maintain video playback continuity.



## 5.4 Analyzing SR-ARQ for Video Delivery
Jack Y.B. Lee

- Client Buffer Requirement
  - To incorporate the effect of retransmission, we note that each retransmission attempt incurs a maximum additional delay of $T_{tran}+T_{wait}+T_i+T_i^+$, where $T_{wait}$ is the retransmission timeout:



- Client Buffer Requirement
  - Since at most $K_i$ transmissions (including retransmissions) are attempted for delivering a video packet, the upper bound for the arrival time of packet *j* of receiver *i* is modified to

$$A_i^j \le \underbrace{\left(T_{start} + T_{tran} + jT_S + T_i + T_i^+\right)}_{\text{Worst-case delay incurred in the first transmission}} + \underbrace{\left(K_i - 1\right)\left(T_{tran} + T_{wait} + T_i + T_i^+\right)}_{\text{Worst-case delay incurred in the next } (K_i-1) \text{ retransmissions}} \quad (15)$$

  - As the packet-loss probability is non-zero, it is possible that some of those first $Y_i$ packets are lost and requires retransmissions. If video packet ($Y_i$–1) is lost, the prefetch process (and hence playback) will be delayed.





- Client Buffer Requirement
  - To avoid unnecessary delay, the receiver starts playback when the expected arrival time for packet ($Y_i$–1) is reached:

$$P_i^0 = A_i^0 + \left(Y_i - 1\right)T_S \quad (16)$$

regardless of whether the packet is physically arrived or not. Hence in general, we have

$$P_i^j = A_i^0 + \left(Y_i + j - 1\right)T_S \quad (17)$$

- Client Buffer Requirement
  - Using the continuity condition, we can obtain $Y_i$ from

$$\max\{A_i^j\} \le \min\{P_i^j\}$$
$$T_{start} + T_{tran} + jT_S + T_i + T_i^+ + (K_i - 1)(T_{tran} + T_{wait} + T_i + T_i^+) \le \min\{A_i^0 + (Y_i + j - 1)T_S\}$$
$$= T_{start} + T_{tran} + T_i + T_i^- + (Y_i + j - 1)T_S \quad (18)$$

  - Rearranging, we then have:

$$Y_i \ge \frac{(K_i - 1)(T_{tran} + T_{wait} + T_i + T_i^+) + (T_i^+ - T_i^-)}{T_S} + 1 \quad (19)$$

  - Compared to the case without packet loss, more buffers are needed to absorb the extra delay incurred in retransmissions.





- Client Buffer Requirement
  - Similarly, to prevent buffer overflow, we need to ensure that:

$$\min\{A_i^{j+Y_i+Z_i-1}\} \ge \max\{P_i^j + T_S\}$$
$$T_{start} + T_{tran} + (j + Y_i + Z_i - 1)T_S + T_i + T_i^- \ge \max\{A_i^0 + (Y_i + j)T_S + T_i\} \quad (20)$$
$$= T_{start} + T_{tran} + T_i + T_i^+ + (Y_i + j)T_S$$

  - Rearranging, we can then obtain $Z_i$ as well:

$$Z_i \ge \frac{(T_i^+ - T_i^-)}{T_S} + 1 \quad (21)$$

  - Note that $Z_i$ is the same as the case without packet loss in (13).

## 5.4 Analyzing SR-ARQ for Video Delivery



- Client Buffer Requirement
  - Summing (19) and (21) we can then obtain the receiver buffer requirement for receiver *i*:

$$L_{ARQ}(i) = \left\lceil \frac{(K_i - 1)(T_{tran} + T_{wait} + T_i + T_i^+) + (T_i^+ - T_i^-)}{T_S} \right\rceil + \left\lceil \frac{T_i^+ - T_i^-}{T_S} \right\rceil + 2 \quad (22)$$

  - In practice, if the network delay and delay jitters are known *a priori*, then the retransmission timeout $T_{wait}$ can simply be set to equal to the maximum network delay $(T_i + T_i^+)$ and (22) can be simplified to

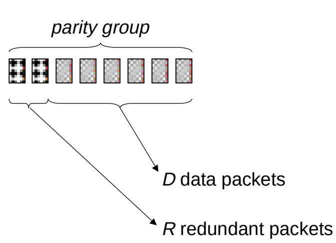
$$L_{ARQ}(i) = \left\lceil \frac{(K_i - 1)(T_{tran} + 2(T_i + T_i^+)) + (T_i^+ - T_i^-)}{T_S} \right\rceil + \left\lceil \frac{T_i^+ - T_i^-}{T_S} \right\rceil + 2 \quad (23)$$



## 5.5 Analyzing FEC for Video Delivery



- Errors to Control
  - Packet loss
- Forms of Redundancy
  - Redundant packets coded using erasure-correcting codes such as Reed-Solomon (RS) codes.

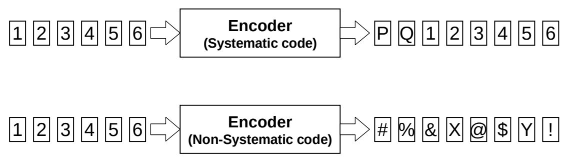


## 5.5 Analyzing FEC for Video Delivery

Jack Y.B. Lee

- Properties of Erasure Correcting Codes
  - An perfect ECC with *R* redundant packets can sustain up to *R* lost packets.


  - Systematic v.s. non-systematic codes





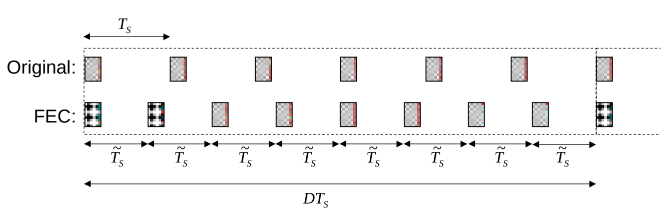

## 5.5 Analyzing FEC for Video Delivery

Jack Y.B. Lee

- Server Transmission Rate
  - Since *R* additional redundant packets must be transmitted for every *D* data packets, the inter-packet transmission time is shortened to:

$$\tilde{T}_S = T_S \frac{D}{D+R} \qquad (24)$$

in order to maintain the same video bit-rate.



## 5.5 Analyzing FEC for Video Delivery

Jack Y.B. Lee

- Residual Loss Probability
  - If there are more than R packet losses within the same parity group, a decoding error will occur.
  - In the worst case, the entire parity group will be lost.
  - Hence the upper bound for RLP is:

$$\varepsilon_i = \sum_{k=R+1}^{D+R} \binom{D+R}{k} p_i^k (1-p_i)^{R+D-k} \tag{25}$$

  - However, if the code is systematic, then only those packets that are lost are affected.
  - The remaining packets received for the parity group can still be used.



## 5.5 Analyzing FEC for Video Delivery

Jack Y.B. Lee

- Residual Loss Probability
  - Let there be *k* lost packets, the probability that *m* of them are video packets and (*k*–*m*) of them are redundant packets is given by

Number of ways to lose *m* out of *D* data packets.

Number of ways to lose (*k-m)* out of *R* redundant packets.

$$\Pr\{m \text{ data packets lost} \mid k \text{ packets lost}\} = \frac{\binom{D}{m}\binom{R}{k-m}}{\binom{D+R}{k}} \tag{26}$$

Number of ways to lose a total of *k* out of (*D+R*) packets in the parity group.

- Residual Loss Probability
  - Average number of data-packet losses per parity group given there are *k* lost packets is:

$$\sum_{m=1}^{\min\{D,k\}} m \Pr\{m \text{ data packets lost} \mid k \text{ packets lost}\} = \sum_{m=1}^{\min\{D,k\}} m \frac{\binom{D}{m}\binom{R}{k-m}}{\binom{D+R}{k}} = \frac{kD}{D+R} \quad (27)$$

  - Conditioning on *k* gives the residual loss probability:

$$\varepsilon_i = \sum_{k=R+1}^{D+R} \binom{D+R}{k} p_i^k (1-p_i)^{R+D-k} \frac{k}{D+R} \quad (28)$$





- Traffic Overhead
  - To maintain a residual loss probability of no more than $p_{max}$, we need a redundancy of at least

$$R_{FEC}(i) = \min\{R \mid \varepsilon_i \le p_{\max}\} \quad (29)$$

  - Using a redundancy of $R_{FEC}(i)$, the traffic overhead at the server link can be obtained from

$$H_{FEC}(i) = \frac{R_{FEC}(i)}{D} \quad (30)$$

## 5.5 Analyzing FEC for Video Delivery



- Client Buffer Requirement
  - Buffering to prevent underflow:

$$Y_i \ge \frac{\left(T_i^+ - T_i^-\right)}{\tilde{T}_S} + D + R \quad (34)$$

  - Buffering to prevent overflow:

$$Z_i \ge \frac{\left(T_i^+ - T_i^-\right)}{\tilde{T}_S} + 1 \quad (35)$$

  - Total amount of buffers needed:

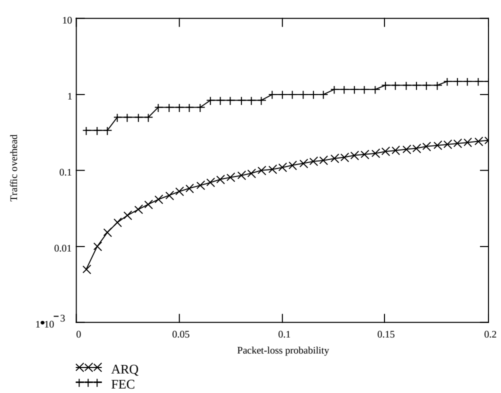
$$L_{FEC}(i) = 2\left\lceil \frac{\left(T_i^+ - T_i^-\right)}{\tilde{T}_S} \right\rceil + D + R + 1 \quad (36)$$



## 5.6 Performance Evaluation



- Traffic Overhead versus Packet Loss Rate



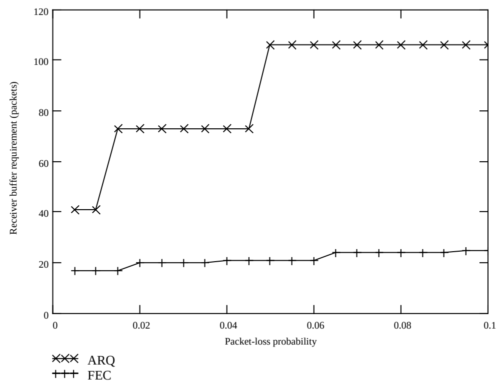

$N = 1$
$T_i = 100ms$
$T_i^+ = -T_i^- = 10ms$
$T_S = 6.83ms$
$D = 6$
$p_{\max} = 0.0001$

ARQ wins over FEC in terms of traffic overhead.

## 5.6 Performance Evaluation



- Client Buffer Requirement versus Packet Loss Rate




$N = 1$

$T_i = 100ms$

$T_i^+ = -T_i^- = 10ms$

$T_S = 6.83ms$

$D = 6$

$p_{\max} = 0.0001$

$Q_S = 1024 bytes$

FEC wins over ARQ in terms of buffer requirement (and hence delay).



## 5.7 Extension to Multicast Video Distribution



- Video Multicast
  - Server sends one copy of video to an arbitrary number of receivers.
  - The network is responsible for replicating video packets for multicast receivers.
- Error Control Under Multicast
  - Request Implosion Problem
    - Too many receivers generating error-control requests (e.g. ACK, NAK) may overload the video server.
  - Traffic Overhead Problem
    - The amount of traffic overhead incurred in error-control may also overload the video server.

- Retransmissions Under Video Multicast

These are client-dependent.

$$H_{ARQ} = \sum_{i=0}^{N-1} \sum_{j=1}^{K_i} (j-1)\left(p_i^{j-1}(1-p_i)\right)$$

Overhead increases with number of receivers because retransmissions are unicasted.

- FEC Under Video Multicast

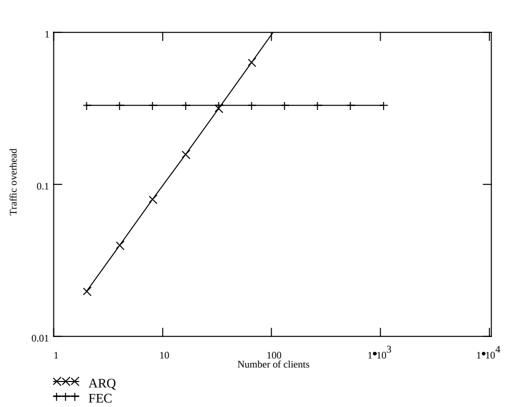
$$H_{FEC} = \frac{R_{FEC}}{D}$$

*Independent of the number of receivers in the system!*





- Traffic Overhead versus Number of Clients




$T_i = 100ms, \forall i$

$T_i^+ = -T_i^- = 10ms, \forall i$

$T_S = 6.83ms$

$D = 6$

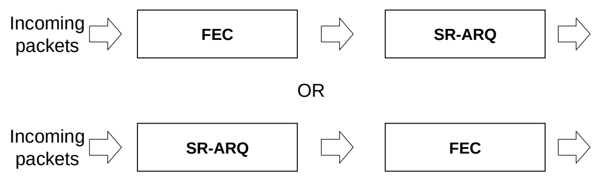
$p_{\max} = 0.0001$

$p_i = 0.01, \forall i$

ARQ wins when number of clients is small, FEC wins otherwise.

## 5.8 Hybrid ARQ/FEC Schemes

- Principle
  - Integrates the strengths of ARQ and FEC and avoids their weaknesses.
  - Perform error-recovery partially by FEC and partially by ARQ.
- Question
  - Should we perform FEC first, then ARQ; or perform ARQ first, then FEC?

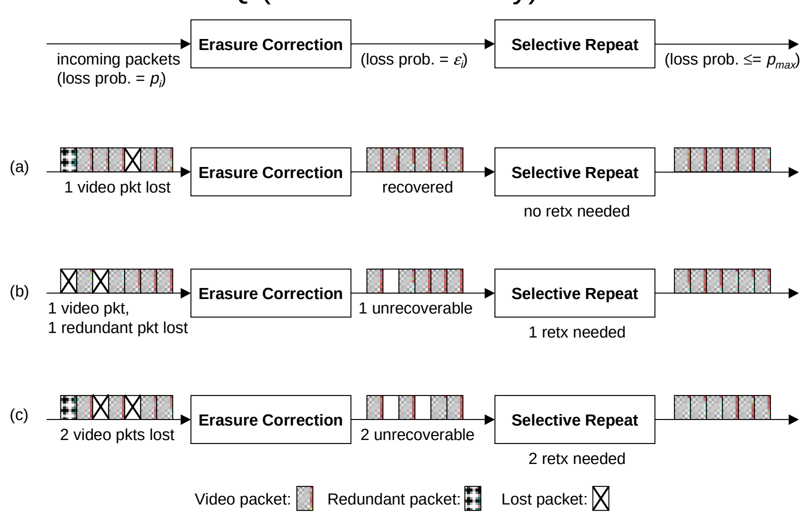







## 5.8 Hybrid ARQ/FEC Schemes

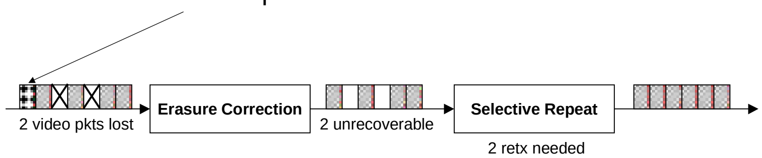
- FEC then ARQ (Passive Recovery)



## 5.8 Hybrid ARQ/FEC Schemes



- FEC then ARQ (Passive Recovery)
  - Observation in Case (c)
    - Two data packet losses; two retransmissions.
    - The redundant packet is NOT utilized at all.



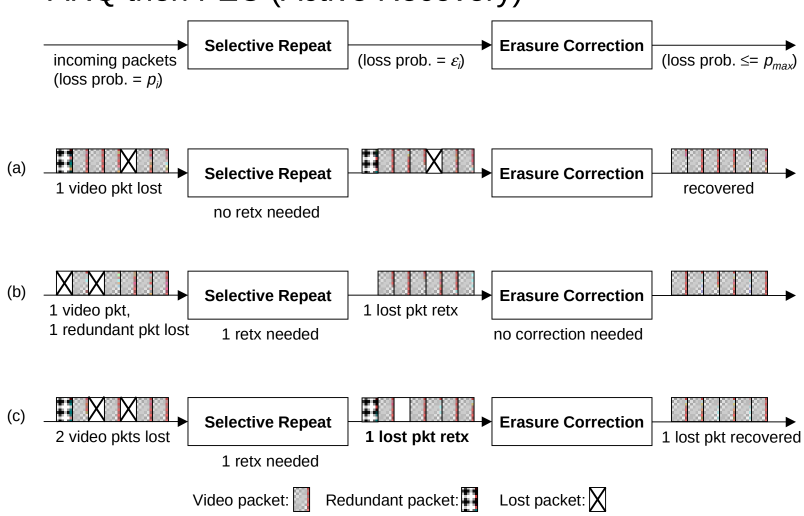

- Ideally, we should only need one retransmission if we can make use of the redundant packet as well.



## 5.8 Hybrid ARQ/FEC Schemes



- ARQ then FEC (Active Recovery)



## 5.8 Hybrid ARQ/FEC Schemes

- ARQ then FEC (Active Recovery)
  - When to initiate retransmission request?
    - As soon as a packet loss is detected
      - Possibly redundant retransmission because the loss could be recoverable by erasure correction already.
    - Better approach:
      - Wait for the entire parity group to arrive;
      - Count the number of lost packets, say *X*;
      - Request retransmission for *X*-*R* lost packets.
    - Tradeoff
      - Additional buffer and delay are incurred at the client.





## 5.8 Hybrid ARQ/FEC Schemes

- Performance Analysis
  - Traffic Overhead
    - Passive Recovery

$$H_{hybrid}(R) = \frac{R}{D} + \sum_{i=0}^{N-1}\left(\sum_{j=1}^{K_i}(j-1)\left(\varepsilon_i^{j-1}(1-\varepsilon_i)\right)\right)$$

where $K_i = \left\lceil \frac{\ln p_{\max}}{\ln \varepsilon_i} \right\rceil$

- Active Recovery

$$H_{hybrid}(R) = \frac{R}{D} + \frac{1}{D}\sum_{i=0}^{N-1}\sum_{m=R+1}^{D+R}\sum_{j=1}^{K_i-1}\binom{D+R}{m} j(m-r)p_i^{m+j-1}(1-p_i)^{D+R-m+1}$$

## 5.8 Hybrid ARQ/FEC Schemes

- Performance Analysis
  - Client Buffer Requirement
    - Passive Recovery

$$L_{hybrid}(i) = \left\lceil \frac{(K_i - 1)\left((D-1)\tilde{T}_S + T_{tran} + 2\left(T_i + T_i^+\right)\right) + \left(T_i^+ - T_i^-\right)}{\tilde{T}_S} \right\rceil + \left\lceil \frac{T_i^+ - T_i^-}{\tilde{T}_S} \right\rceil + D + R + 1$$

    - Active Recovery

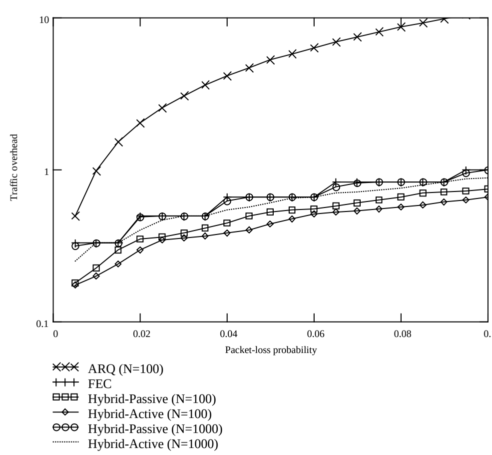
$$L_{hybrid}(i) = \left\lceil \frac{(K_i - 1)\left(\tilde{T}_S + T_{tran} + 2\left(T_i + T_i^+\right)\right) + \left(T_i^+ - T_i^-\right)}{\tilde{T}_S} \right\rceil + \left\lceil \frac{T_i^+ - T_i^-}{\tilde{T}_S} \right\rceil + D + R + 1$$





## 5.9 Performance Comparisons

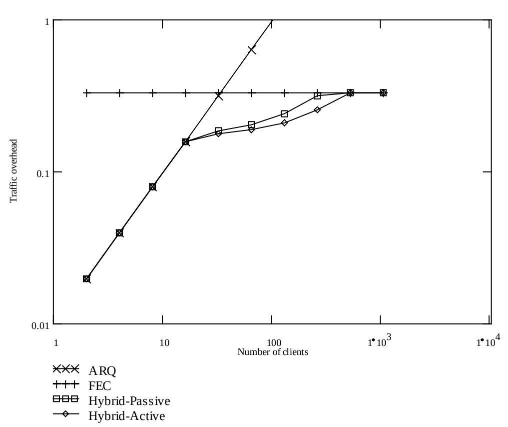
- Traffic Overhead v.s. Packet-Loss Probability



## 5.9 Performance Comparisons

Jack Y.B. Lee

- Traffic Overhead v.s. Number of Users

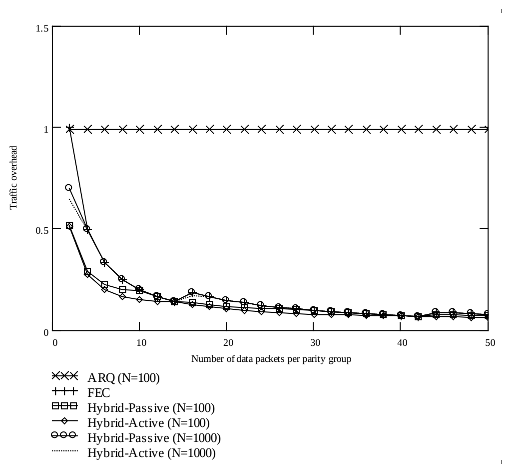





## 5.9 Performance Comparisons

Jack Y.B. Lee

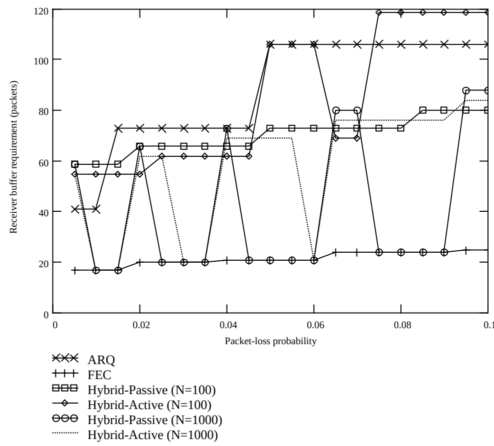
- Traffic Overhead v.s. Parity Group Size



## 5.9 Performance Comparisons

Jack Y.B. Lee

- Client Buffer Requirement v.s. Packet-Loss Rate

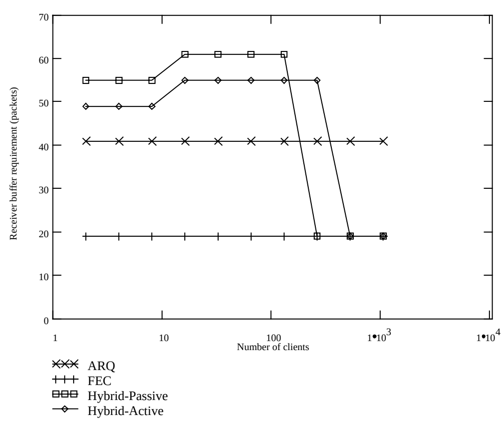





5.9 Performance Comparisons

Jack Y.B. Lee

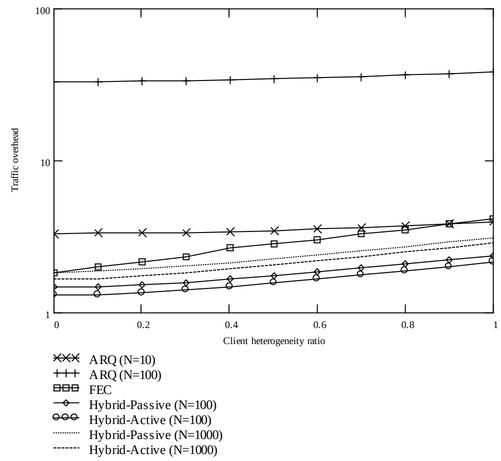
- Client Buffer Requirement v.s. Number of Users



## 5.9 Performance Comparisons



- Traffic Overhead versus Client Packet Loss Heterogeneity






## 5.10 Other Approaches



- Multicast Retransmission
  - Reduce duplicate retransmissions to multiple receivers.
- Multicast Parity Retransmission
  - Retransmit parity/redundant packets instead of the lost data packets to allow other receivers to recover different lost packets.
- Multicast Retransmission Requests
  - Make use of request-suppression scheme among receivers to remove duplicate requests.
- Hierarchical Retransmission
  - Use intermediate nodes/receivers to carry out retransmissions for leave nodes to reduce load at server.

# References



[0] Y.B.Lee, H.C.Chan, "An Efficient Error-Control Algorithm for Multicast Video Distribution," *Submitted for publication*, 1998.

[1] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing," *IEEE/ACM Transactions on Networking*, vol.5(6), Dec 1997, pp.784-803.

[2] J. Lin, and S. Paul, "RMTP: A Reliable Multicast Transport Protocol," *Proc. IEEE INFOCOM '96*, San Francisco, CA, USA, March 1996.

[3] X.R. Xu, A.C. Myers, H. Zhang, and R. Yavatkar, "Resilient Multicast Support for Continuous-Media Applications," *Proc. 7th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'97)*, St. Louis, MO, USA, 1997, pp.183-194.

[4] C. Papadopoulos, G. Parulkar, and G. Varghese, "An Error Control Scheme for Large-scale Multicast Applications," *Proc. IEEE INFOCOM '98*, San Francisco, CA, USA, March 1998.

[5] A.M. Costello, "Search Party: Using Randomcast for Reliable Multicast with Local Recovery," *Technical Report No. UCB//CSD-98-1011*, University of California Berkeley, CA, August 1998.

[6] J. Nonnenmacher, E.W. Biersack, and D. Towsley, "Parity-Based Loss Recovery for Reliable Multicast Transmission," *In Proceedings of ACM SIGCOMM'97*, pp.289-300, Cannes, France, September, 1997.

[7] G. Carle, and E.W. Biersack, "Survey of Error Recovery Techniques for IP-Based Audio-Visual Multicast Applications," *IEEE Network*, vol.11(6), Nov/Dec 1997, pp.24-36.

[8] S. Pejhan, M. Schwartz, and D. Anastassiou, "Error Control Using Retransmission Schemes in Multicast Transport Protocols for Real-Time Media," *IEEE/ACM Transactions on Networking*, vol.4(3), June 1996, pp.413-427.

[9] M.T. Lucas, B.J. Dempsey, and A.C. Weaver, "Distributed Error Recovery for Continuous Media Data in Wide-Area Multicast," *Technical Report CS95-52*, Department of Computer Science, University of Virginia, July 1995.

[10] X. Li, S. Paul, P. Pancha, and M. Ammar, "Layered Video Multicast with Retransmission (LVMR): Evaluation of Error Recovery Schemes," *Proc. 7th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'97)*, St. Louis, MO, USA, 1997, pp.161-172.