Dealing with Packet Loss

Kevin Jeffay
Department of Computer Science
University of North Carolina at Chapel Hill
jeffay@cs.unc.edu
November 18, 1999

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

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

**Retransmission-Based Error Correction**

**The retransmission “budget”**

If:

\[
gap \text{ length} + \left(3 \times \frac{\text{one-way transmission time}}{\text{playout latency}}\right) <\]

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 requested 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

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
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
  - $\text{average receiver buffering delay} \geq (1.5 \times \text{RTT}) + \text{gap length}$

- 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

- **Example:** 3-way XOR

```
Sample 1  ×  Sample 2  ×  Sample 3  =  Repair Sample
```

The Incidence of Consecutive Packet Loss

The INRIA unicast IVS experiments

- Packet loss from INRIA to UCL
The Incidence of Consecutive Packet Loss
The INRIA unicast IVS experiments

- Frequency distribution of consecutive packet losses from INRIA to UCL

The Incidence of Consecutive Packet Loss
The INRIA unicast IVS experiments

- Packet loss from INRIA to Maryland at 3 pm (9 am EST)
The Incidence of Consecutive Packet Loss
The INRIA multicast IVS experiments

Packet loss from INRIA to UCL

Frequency distribution of consecutive packet losses from INRIA to UCL
Packet Loss on the Internet Today
Audio packet loss for UNC-UW-UNC ProShare xmission

- Percentage of audio packet loss during a 5 minute interval
  - Each line represents a 5 second average
Forward Error Correction
Recoding/transcoding of original sample

- If losses are infrequent, perhaps we can get by with lower quality repairs

- **Example**: 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 worthwhile for large packets
  - (This is due in large part to quality of LPC coded speech)
Forward 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