CSci 5105

Introduction to Distributed Systems

Communication: Streaming, Multicast

Data Streaming

- Data stream
  - sequence of data packets
  - Continuous or discrete
- Asynchronous
  - No timing constraints between packets (discrete)
- Synchronous
  - Timing constraints between packets: max delay
  - Timing is fundamental to correct interpretation (audio, video)

Data Streaming

- Isochronous
  - Contraints on min and max delay
  - Jitter

Timing Constraints

- Intra-stream synchronization
- Quality of service (QoS)
  - The required bit rate at which data should be transported
  - The maximum delay until a session has been set up
  - The maximum end-to-end delay
  - The maximum delay variance, or jitter
  - The maximum round-trip delay

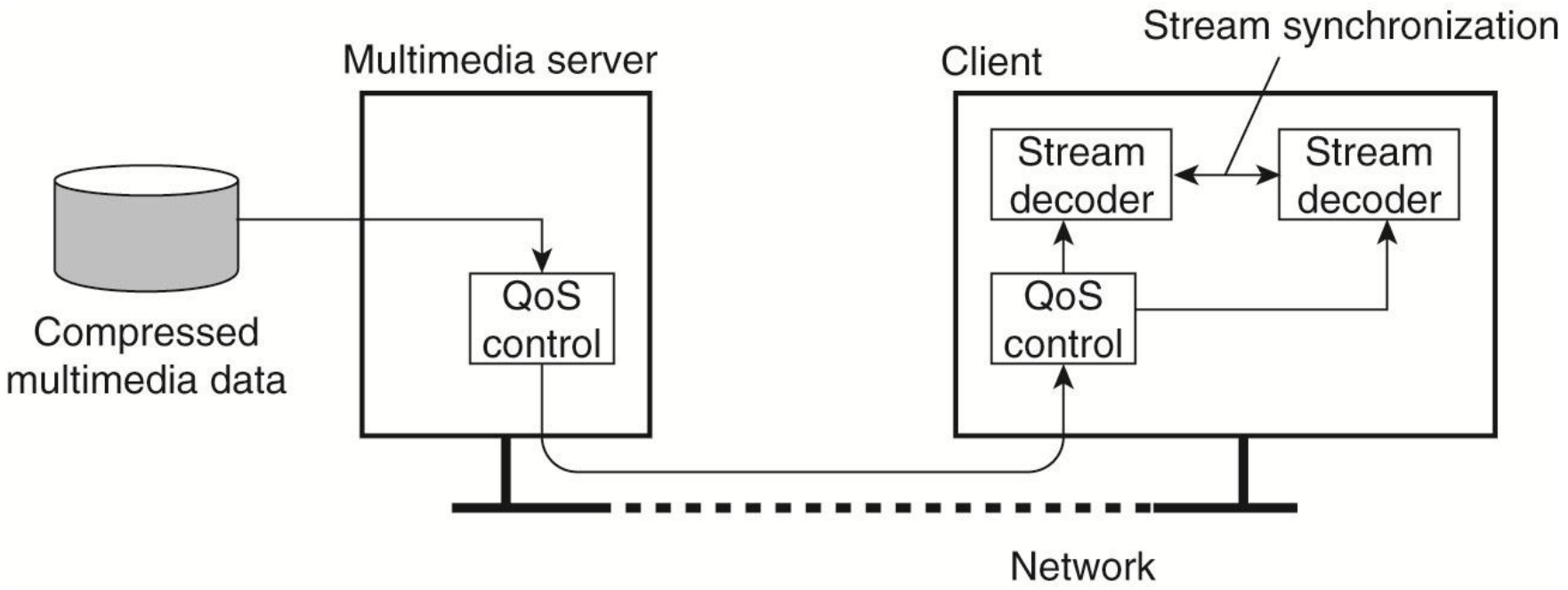
Streaming Types

- Live or Stored Data
  - Latter provides more opportunity for optimization
- Simple stream
  - single data stream
- Complex
  - multiple sub-streams

Complex Streams

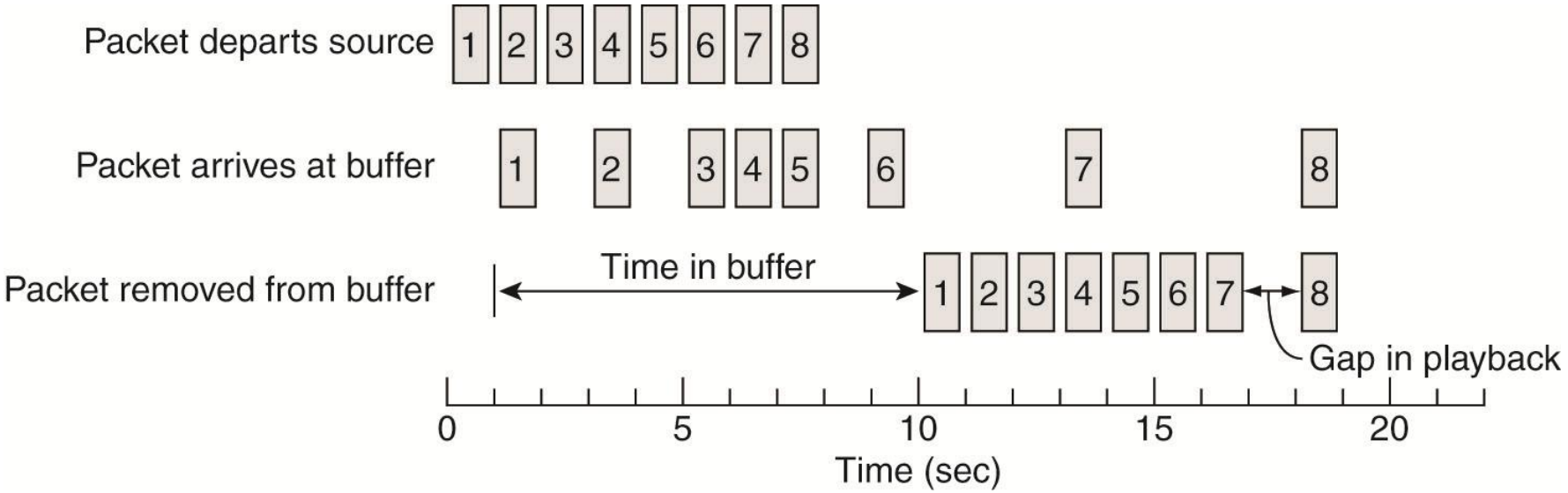
- Continuous + discrete
  - Slide show + audio or video
- Continuous + continuous
  - Video + audio
- Need to synchronize across streams!

Data Stream Architecture



Enforcing QoS

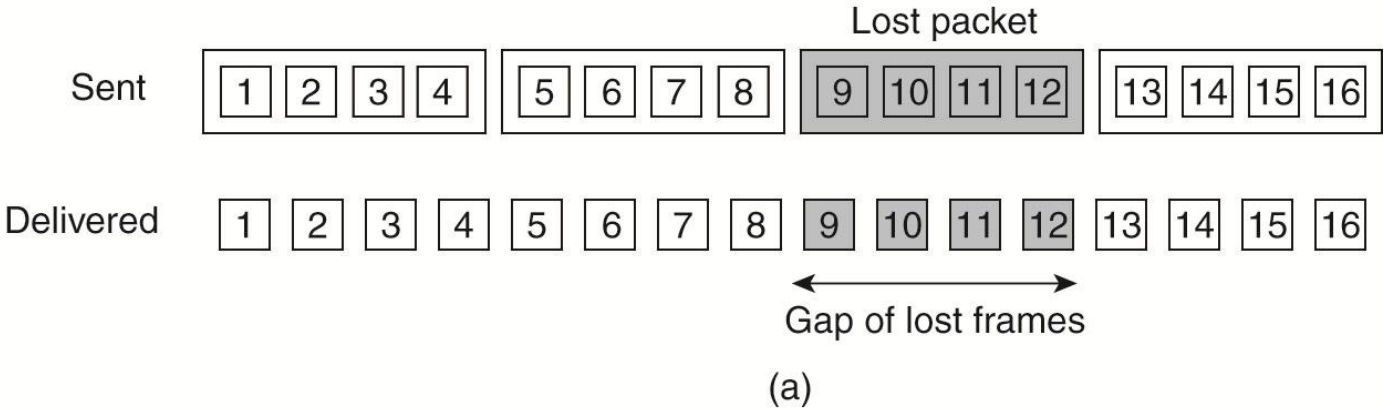
- Buffering
  - Buffer size = 8

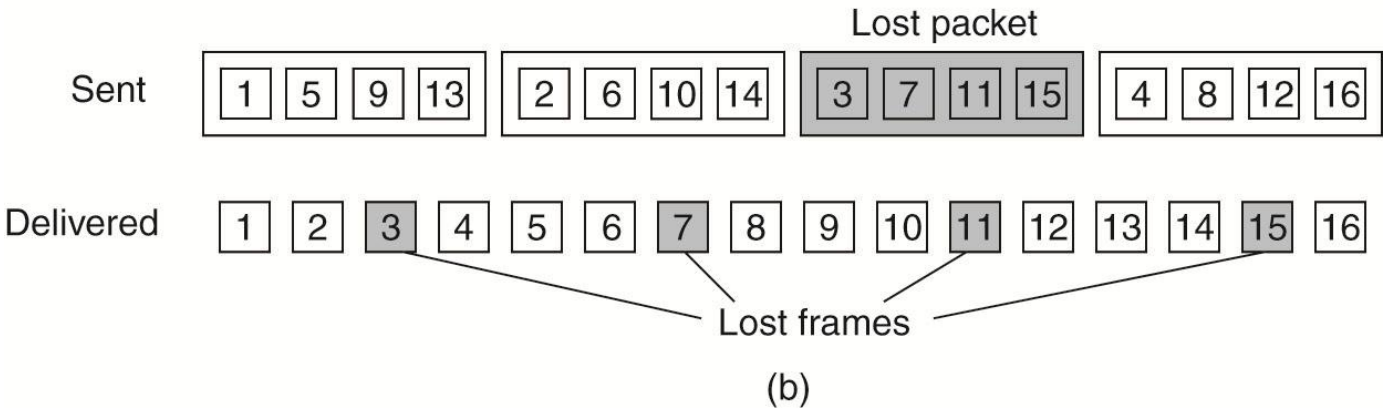


  - Oops packet 8 is outside the window
  - Solution/tradeoff?

More QoS

- Interleaving





Complex Stream Synchronization

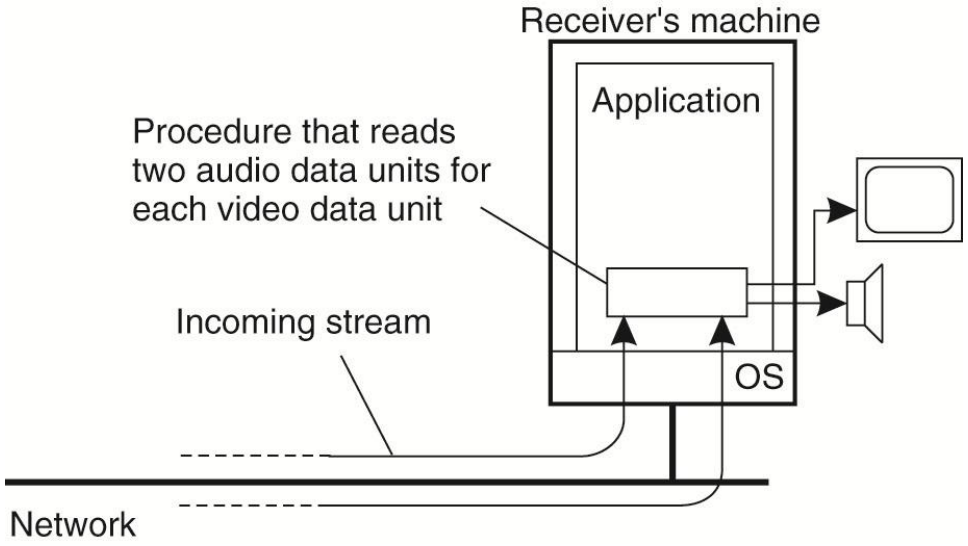
- Discrete + Continuous
  - Slide show + audio or video
- Continuous + continuous
  - Video + audio
  - More challenging

Syncrhonization Location

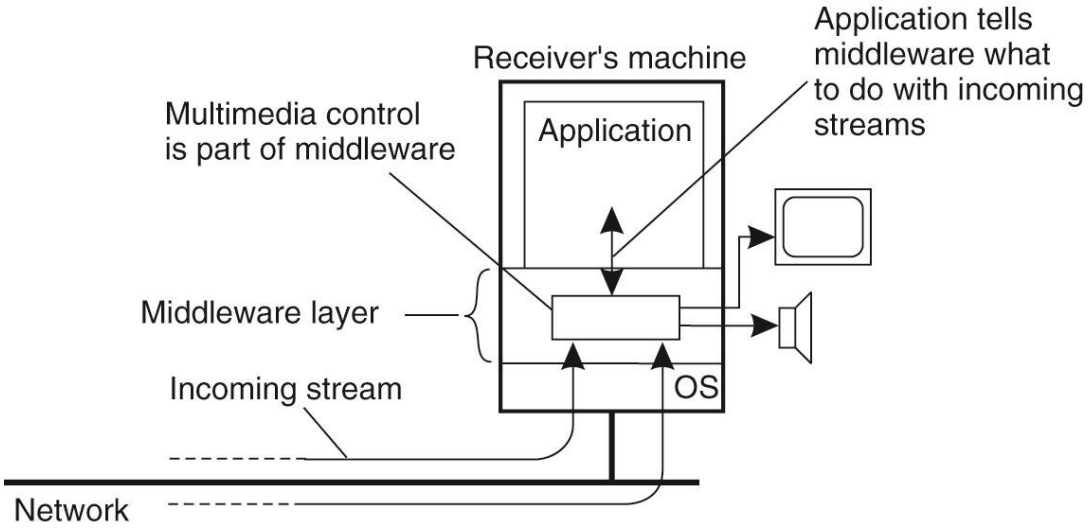
- Sender synchronization
  - Pair up associated packets of each stream (audio and video) and send together
  - Downsides? More complex protocol. MPEG4
- Receiver synchronization
  - Send distinct streams to receiver and let receiver pair them up
  - Difficulties?

Receiver-side Architectures

- Low-level



- High-level



Multicast

- RPC is point-to-point
- Multicast
  - one sender, multiple receivers
  - emulate using a set of point-to-point messages

- Efficient multicast
  - enable suppression of duplicate messages
  - better link utilization
  - examples where multicast would be useful?

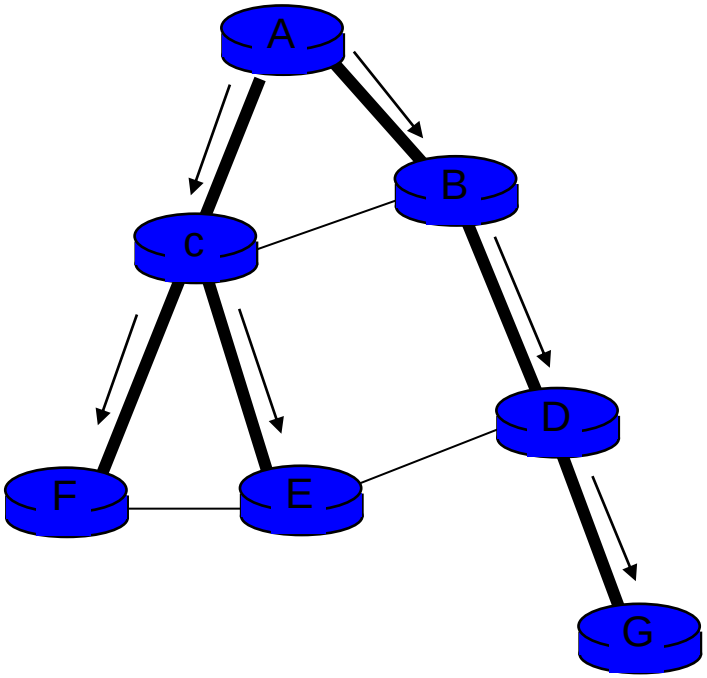
Multicast Types

- IP multicast
  - Specific address format
  - Sender can send to an address group
  - Requires special router support
- Application-level multicast
  - over peer-to-peer network
  - perhaps even in a MoM network

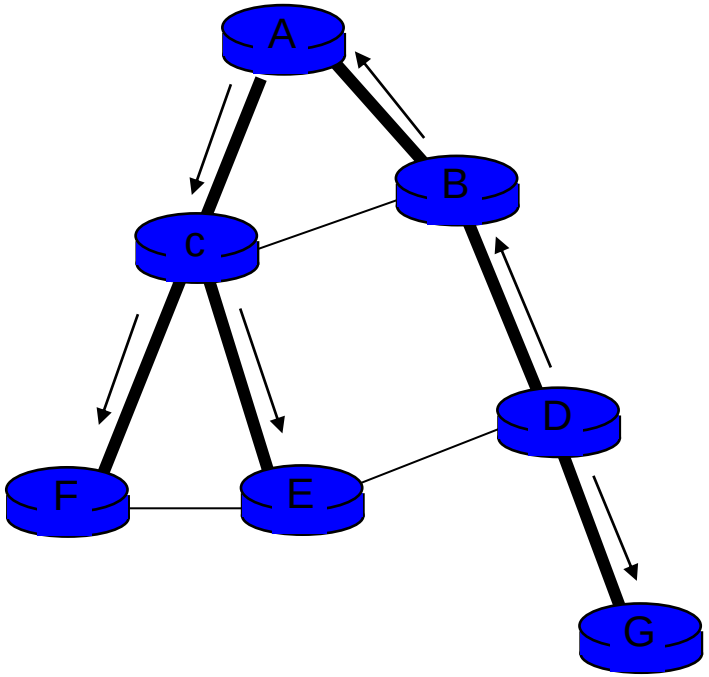
Multicast Trees

- Fundamental to all forms of multicast is the construction of a multicast tree
- How tree gets built is complex
  - Given set of nodes, consider all links, routers
- May have to rebuild the tree as nodes come and go – incremental?

Spanning Tree



(a) multicast initiated at A



(b) multicast initiated at D

Epidemic Communication

- Imagine you want to spread information rapidly in a very large distributed system
- No central coordination
- All nodes may report local information and information from neighbors

Information Dissemination Models

- Anti-entropy propagation model
  - Node P picks another node Q at random
  - Subsequently exchanges updates with Q

- Approaches to exchanging updates
  - P only pushes its own updates to Q
  - P only pulls in new updates from Q
  - P and Q send updates to each other (push-pull is the best)

Single node infected … propagates to entire network in log N

Gossiping

- Variation
- P contacts Q to push information I
- If Q reports that it already has I
  - P reduces interests in propagating by 1/k

- Interesting Case
  - Propagating deletes “death certificates”
  - When to garbage collect those?

Applications

- Want to know the average load of a billion node network?
- Nodes i and j exchange load and update their value as (load i + load j)/2
- Eventually converges
- What about max?