Midterm Review

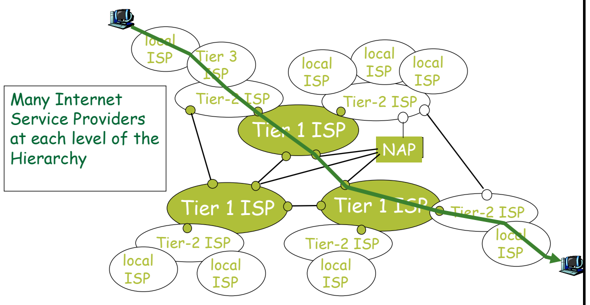
EECS 122: Lecture 1

Department of Electrical Engineering and Computer Sciences
University of California
Berkeley

Review: Check List

- Big Picture
  - Layers
  - Where protocols are implemented
  - Switching Techniques
- Applications
  - DNS
  - HTTP
  - SMTP
- Transport
- Network Layer: Routing Protocols
  - Class-Based; Classless Addressing
  - Dijkstra; Bellman-Ford
  - BGP
- Network Layer Forwarding
  - Architecture: Input, Output
  - Fabric Architectures: Shared Bus, Shared Mem, Switched
  - Basic Performance Metrics (Packet Delay, Little's Law)
- QoS
  - Max Min Fairness
  - Mechanisms for QoS
  - Scheduling: Fairness, GPS, WFQ
  - Intserv, Diffserv
  - Internet Multimedia
    - Streaming and VOIP

The internet consists of many networks



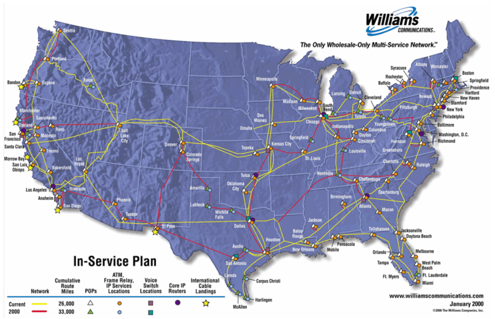
Many Internet Service Providers at each level of the Hierarchy

---

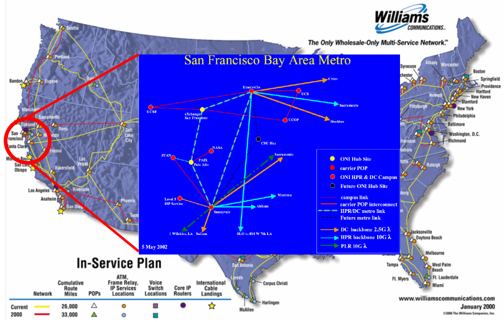
Two fundamentally different ways to forward information

- **Circuit Switched**
  - Information is exchanged in units of "calls"
  - Network resources are reserved for the duration of the call
  - Example: The Phone Network
    - Once a call goes through, subsequent calls cannot degrade call quality
- **Packet Switched**
  - Information is exchanged in units of "packets"
  - Typically, no resources are reserved
  - **Datagram**: Each packet is forward independently
    - Example: The Internet
  - **Virtual Circuit**: All the packets from a given stream take the same path through the network
    - Example: ATM, ISDN, Intserv

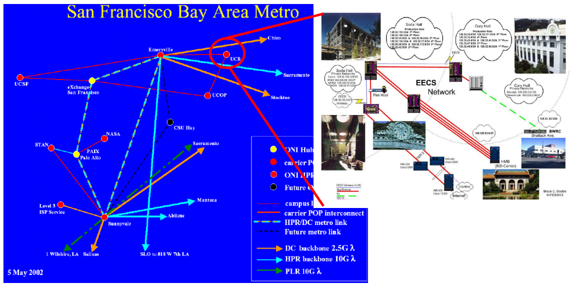
Backbone Network



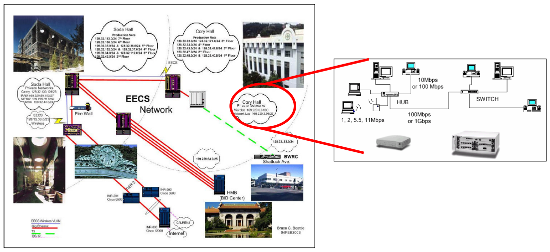
Metropolitan Area Network



Campus Network

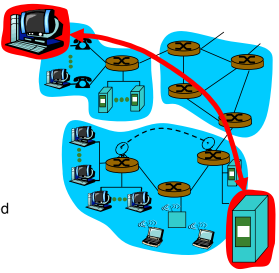


Local Area Network



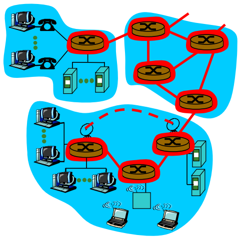
The network edge:

- end systems (hosts):
  - run application programs
  - e.g. Web, email
  - at “edge of network”
- client/server model
  - client host requests, receives service from always-on server
  - e.g. Web browser/server; email client/server
- peer-peer model:
  - minimal (or no) use of dedicated servers
  - e.g. Gnutella, KaZaA, Skype



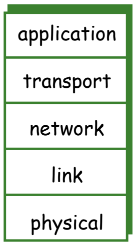
The Network Core

- Many interconnected “sub-networks”
- Many different architectures
- Advertises a “service” to the end devices
  - E.g. Phone network v/s the Internet

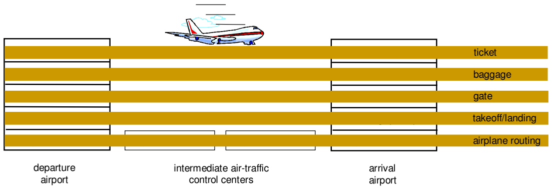


Internet protocol stack

- application: supporting network applications
  - FTP, SMTP, HTTP
- transport: host-host data transfer
  - TCP, UDP
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements
  - PPP, Ethernet
- physical: bits “on the wire”

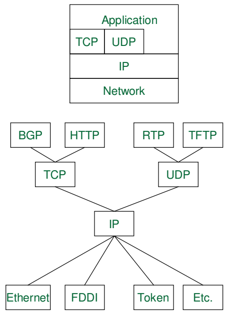


Layering of airline functionality



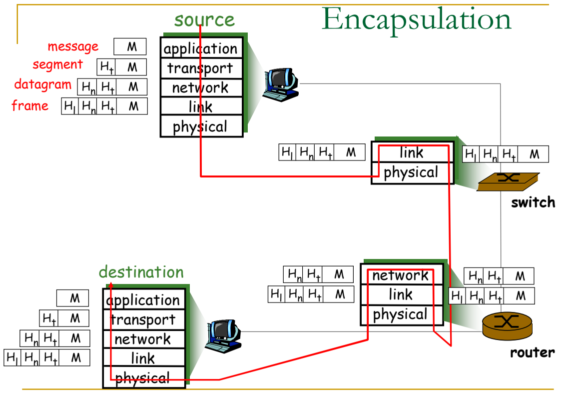
Layers: each layer implements a service
- via its own internal-layer actions
- relying on services provided by layer below

An Advanced View of Internet Layering



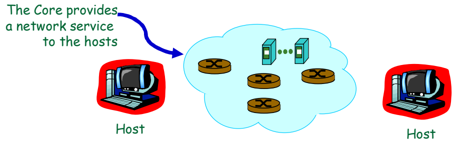
- Almost Any kind of application can write directly on IP
  - Including new transport protocols
- IP cannot be avoided
- As long as the routers speak IP, any application that can make do with datagram service can be written and implemented on the end devices.
  - No co-ordination, standards activity etc. is required!!

Encapsulation



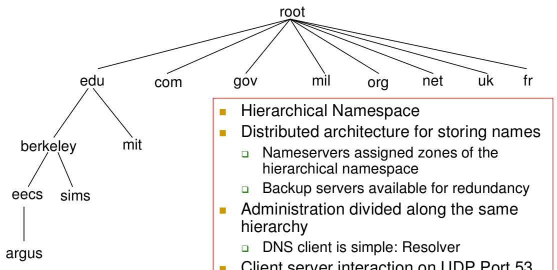
Back

Application Protocols



- Host-Host:
  - HTTP, SMTP
- Host-Network:
  - DNS
- Network-Network:
  - Routing Protocols (e.g. OSPF)

DNS Features



- Hierarchical Namespace
- Distributed architecture for storing names
  - Nameservers assigned zones of the hierarchical namespace
  - Backup servers available for redundancy
- Administration divided along the same hierarchy
  - DNS client is simple: Resolver
- Client server interaction on UDP Port 53 (but can use TCP if desired)

Three kinds of DNS servers

- Top-level domain (TLD) servers: responsible for com, org, net, edu, etc, and all top-level country domains uk, fr, ca, jp.
  - Network solutions maintains servers for com TLD
  - Educause for edu TLD
- Authoritative DNS servers: organization's DNS servers, providing authoritative hostname to IP mappings for organization's servers (e.g., Web and mail).
  - Can be maintained by organization or service provider
- Local DNS Server: When a host makes a DNS query, query is sent to its local DNS server
  - Acts as a proxy, forwards query into hierarchy
  - The Local server is not a part of the DNS hierarchy

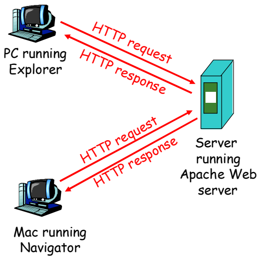
How does a name get resolved

- Query "walks" its way up and down the hierarchy
  - Iterated query
    - I don't know, but here's who to ask next
  - Recursive query
    - I don't know right now, but I'll get back to you…

HTTP

HTTP: hypertext transfer protocol

- Web's application layer protocol
- client/server model
  - *client:* browser that requests, receives, "displays" Web objects
  - *server:* Web server sends objects in response to requests
- Stateless Protocol



HTTP

Uses TCP:

- client initiates TCP connection (creates socket) to server, port 80
- server accepts TCP connection from client
- HTTP messages (application-layer protocol messages) exchanged between browser (HTTP client) and Web server (HTTP server)
- TCP connection closed

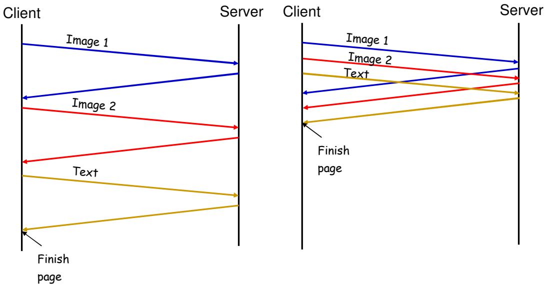
HTTP 1.0 : Non Persistent

- Client opens a TCP connection for each request

HTTP 1.1: Persistent

- server leaves connection open after sending response
- subsequent HTTP messages between same client/server sent over open connection

HTTP 1.1: Pipelining



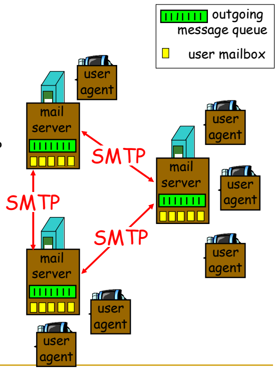
Electronic Mail

Three major components:

- user agents
- mail servers
- simple mail transfer protocol: SMTP

User Agent

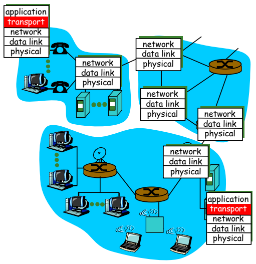
- a.k.a. "mail reader"
- composing, editing, reading mail messages
- e.g., Eudora, Outlook, elm, Netscape Messenger
- outgoing, incoming messages stored on server



What is the network layer?

- So far we have been treating the network "as a cloud" that "routes packets"
- The Network Layer

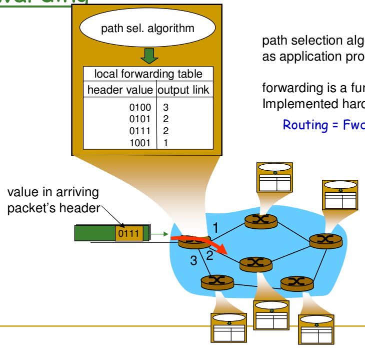
1. Chops transport layer messages into IP packets
2. Delivers them to the correct destination(s)
3. Reconstitutes packets into transport layer messages



Network Layer Functions

- Control Functions: Ensure that routers are configured to deliver packets correctly to the destination
  - Path Selection (called routing in the book)
  - Connection Setup: required in virtual circuit routing.
- Data Functions: Ensure that arriving packets are forwarded correctly within a router with minimum delay
  - Forwarding

Interplay between path selection and forwarding



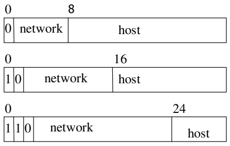
path selection algorithms run as application protocols

forwarding is a function mostly Implemented hardware

Routing = Fwding + Path Selection

Class-base Addressing

- Addressing reflects internet hierarchy
  - 32 bits divided into 2 parts:
  - Class A
  - Class B
  - Class C



~2 million nets
256 hosts

IP addressing: CIDR

CIDR: Classless InterDomain Routing

- net portion of address of arbitrary length: subnet
- address format: a.b.c.d/x, where x is # bits in subnet portion of address

subnet part ← → host part

11001000 00010111 00010000 00000000

200.23.16.0/23

CIDR: Example

- Example 128.5.10/23
  - Common prefix is 23 bits: 01000000 00000101 0000101
  - Number of addresses: $2^9 = 512$
- Prefix aggregation
  - Combine two address ranges
  - 128.5.10/24 and 128.5.11/24:
  - 01000000 00000101 00001010
    01000000 00000101 00001011

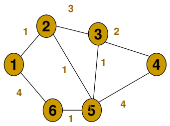
  gives 128.5.10/23
- Routers match to longest prefix

Link State Protocols

1. Every node learns the topology of the network
   - Flooding of Link State Packets (LSP)
2. An efficient shortest path algorithm computes routes to every other node
3. Node updates Forwarding Table

Route Computation: Dijkstra

- Every node knows the graph
  - All link weights are >= 0
- Goal at node 1: Find the shortest paths from 1 to all the other nodes.
- Each node computes the same shortest paths so they all agree on the routes
- Strategy at node 1: Find the shortest paths in order of increasing path length
  - List the nodes in increasing order of (shortest) distance
  - S(k): closest k nodes
  - Iteration k yields S(k) and a way to get there



S(1)={1}
S(2)={1,2}
S(3)={1,2,5}

Distance Vector Algorithms

- Nodes communicate distance estimates to their neighbors, not topology information
- Based on the Bellman Ford Equation:

Define D(x,y) to the shortest distance from x to y.

$$D(x,y) = \min_{v \in N(x)} \{c(x,v) + D(v,y)\}$$

where N(x) are the neighbors of node x.

- Why is this true?

Let D(x,v,y) be the shortest path from x to y where the first node after x is v.

Then $D(x,v,y) = c(x,v) + D(v,y)$.

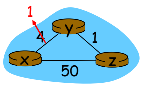
$D(x,y) = \min_v D(x,v,y)$

$= \min_v \{c(x,v) + D(v,y)\}$

Distance Vector: link cost changes

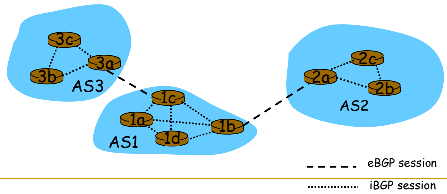
Link cost changes:

- node detects local link cost change
- updates routing info, recalculates distance vector
- Good news travels fast but Bad news can travel very slowly….Counting to infinity



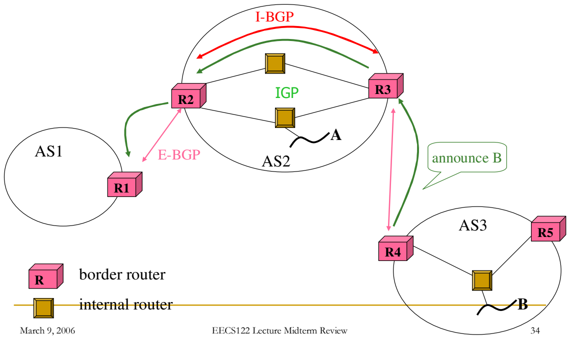
BGP: Path Vector Protocol

- Pairs of routers (BGP peers) exchange routing info over semi-permanent TCP connections: BGP sessions
  - BGP sessions need not correspond to physical links.
- When AS2 advertises a prefix to AS1, AS2 is *promising* it will forward any datagrams destined to that prefix towards the prefix.
  - AS2 can aggregate prefixes in its advertisement



Back

I-BGP and E-BGP



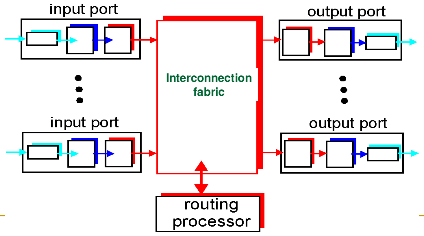
The Forwarding Decision Process

- Datagram Routing: Each packet is independently forwarded at each router
  - Must look up IP address ranges
  - Match Longest Prefix
- Virtual Circuit Routing:
  - call setup, teardown for each call *before* data can flow
  - each packet carries VC identifier (not destination host address)
  - *every* router on source-dest path maintains "state" for each passing connection
  - link, router resources (bandwidth, buffers) may be *allocated* to VC (dedicated resources = predictable service)

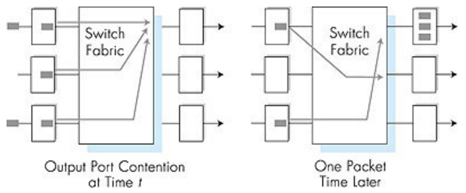
Router Architecture Overview

Two key router functions:

- run routing algorithms/protocol (RIP, OSPF, BGP)
- *forwarding* datagrams from incoming to outgoing link

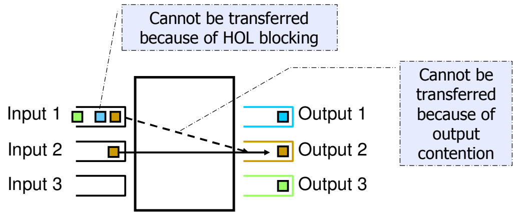


Output Queued Routers



Input Queues: Head-of-line Blocking

- The packet at the head of an input queue cannot be transferred, thus blocking the following packets



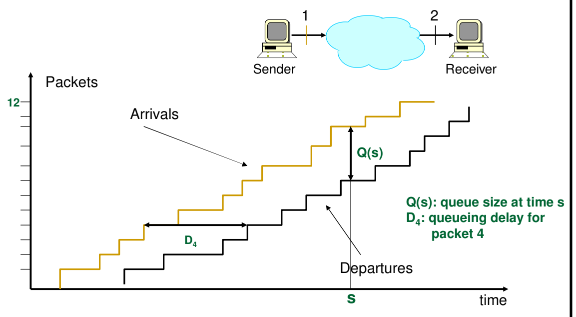
Wastes router capacity

Components of Per Hop Delay

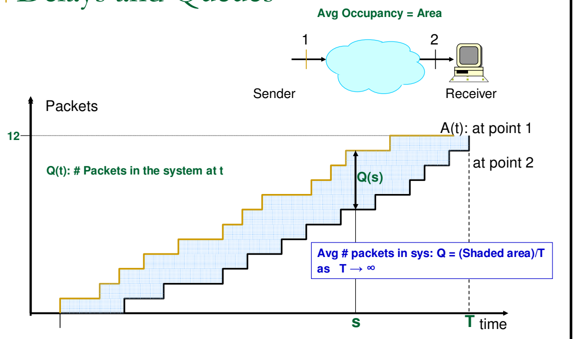
- **Propagation delay:** time it takes the signal to travel from source to destination
- **Packet transmission time:** time it takes the sender to transmit all bits of the packet
- **Queuing delay:** time the packet need to wait before being transmitted because the queue was not empty when it arrived
- **Processing Time:** time it takes a router/switch to process the packet header, manage memory, etc

Only random component

Delays and Queues

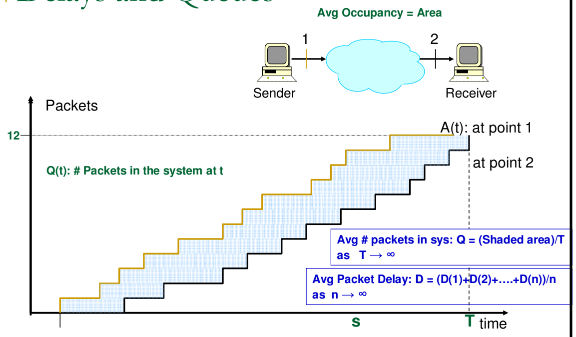


Delays and Queues



Avg # packets in sys: Q = (Shaded area)/T as T → ∞

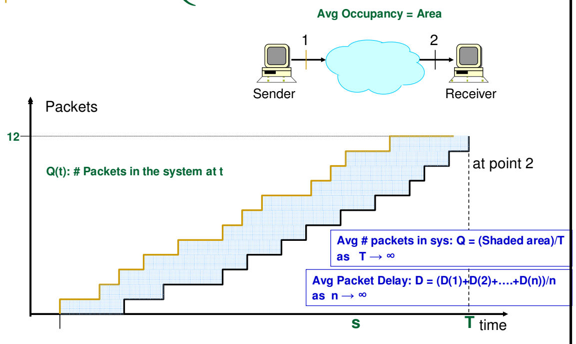
Delays and Queues



Avg # packets in sys: Q = (Shaded area)/T as T → ∞

Avg Packet Delay: D = (D(1)+D(2)+….+D(n))/n as n → ∞

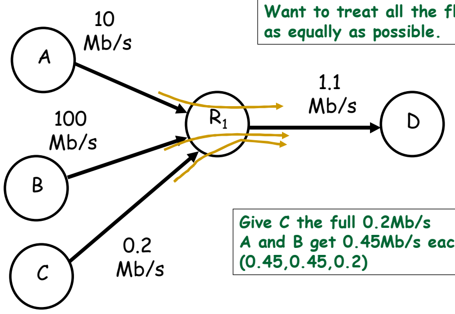
Little's Law: Q=Dα



Flows and Fairness

Max-Min Fair Allocation

Want to treat all the flows as equally as possible.



Give C the full 0.2Mb/s A and B get 0.45Mb/s each (0.45,0.45,0.2)

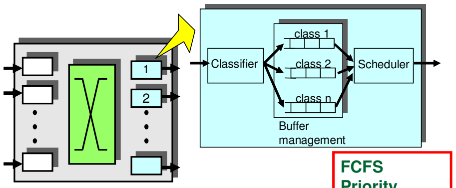
Mechanisms to Improve Best Effort

- Classification and Scheduling
- Drop Policies
- Call admission
- Policing

- Implementing even a subset of these can help!

Advanced Queuing Functions

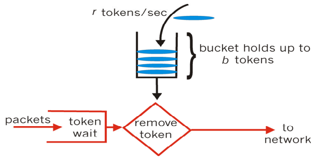
- Packet classification: map each packet to a predefined class
  - use to implement more sophisticated services (e.g., QoS)
- Flow: a subset of packets between any two endpoints in the network



FCFS
Priority
Round Robin
WRR

Policing Mechanisms

Token Bucket: limit input to specified Burst Size and Average Rate.



- bucket can hold b tokens
- tokens generated at rate *r token/sec* unless bucket full
- *over interval of length t: number of packets admitted less than or equal to (r t + b).*

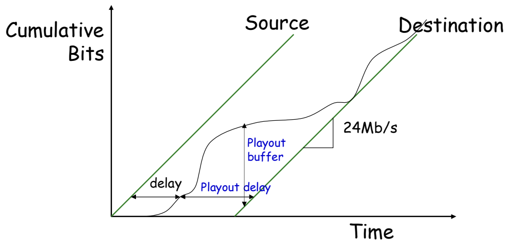
Intserv/Diffserv Comparison

	Best-Effort	Diffserv	Intserv
Service	Connectivity No isolation No guarantees	Per aggregate isolation Per aggregate guarantee	Per flow isolation Per flow guarantee
Service scope	End-to-end	Domain	End-to-end
Complexity	No setup	Long term setup	Per flow steup
Scalability	Highly scalable (nodes maintain only routing state)	Scalable (edge routers maintains per aggregate state; core routers per class state)	Not scalable (each router maintains per flow state)

Internet Multimedia: bag of tricks

- use UDP to avoid TCP congestion control (delays) for time-sensitive traffic
- client-side adaptive playout delay: to compensate for delay
- server side matches stream bandwidth to available client-to-server path bandwidth
  - chose among pre-encoded stream rates
  - dynamic server encoding rate
- error recovery (on top of UDP)
  - FEC, interleaving
  - retransmissions, time permitting
  - conceal errors: repeat nearby data
- Has to be VERY application specific…

Smooth out non-interactive traffic with a playback buffer



Playback Point