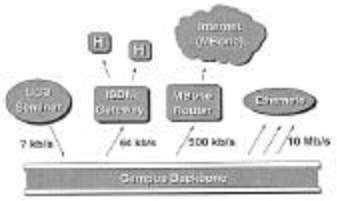
Receiver-driven Layered Multicast

Steven McCanne, Van Jacobson and Martin Vetterli

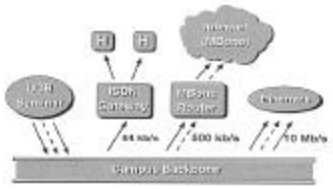
ACM SIGCOMM, Stanford California
August 1996

Problem



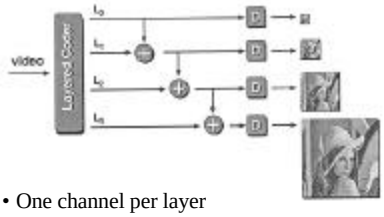
- Network heterogeneity
- One output to multiple users with varied capabilities
- How to decide the rate?
  - Minimum? Maximum?
- How to determine network capacity?

Solution?



- Multiple levels of quality across multiple network channels
- Receivers decide their own rates of reception
- Note, requires layered media streams

Layered Video Stream

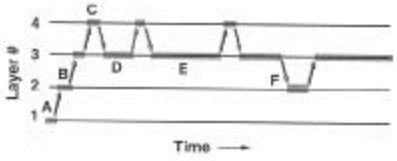


- One channel per layer
- Layers are additive
- Adding more channels gives better quality
- Adding more channels requires more bandwidth

The RLM Protocol

- High level abstraction
  - on congestion, drop a layer
  - on spare capacity, add a layer
- Q: How does the receiver decide?
  - detection time
  - capacity inference

Determining Capacity

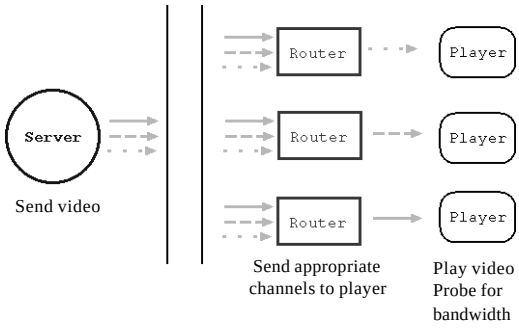


- At a "well-chosen" time conduct a join experiment
- If congestion is experienced, leave the new group
- If no congestion for awhile, try to join next higher group

Project 2: Mini-RLM

- Three programs
  - Server: send video on all channels
  - Router: receive video, 'route' appropriate channels to client
  - Player: receive video, probe for capacity, play video

Mini RLM



Mini Video

Taking a Walk	Taking a Walk	Taking a Walk
0	0	0
|	|	|
|	/ \	|

- Text-based frames
- One frame per second on each channel
  - sleep! alarm! setitimer!

Video Scaling

- Receiver

```
Channel 1:     1  4  7    Time →
Channel 2:      2  5  8
Channel 3:       3  6  9
```

- Base case, channel 1 only:

```
Channel 1:     1  4  7 ...
```

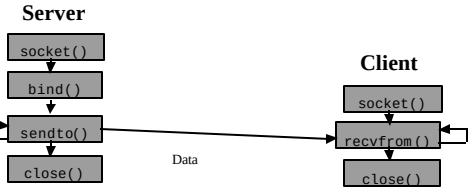
- If more bandwidth:

```
Channels 1-2:  12 45 78 ...
```

- Full quality:

```
Channels 1-3:  123456789 ...
```

IP Multicast Client-Server



-Same as UDP client server!
  -Multicast requires special address (reserved)
  -A few socket options
-No two-way communication

IP Multicast

- Server
  - Send to 239.0.0.1 to 239.255.255.255

```
addr.sin_addr.s_addr = inet_addr(239.0.0.1);
```

  - Port
- Receiver

```
struct ip_mreq mreq;
mreq.imr_multiaddr.s_addr = inet_addr(239.0.0.1);
mreq.imr_interface.s_addr = htonl(INADDR_ANY);
setsockopt(sock, IPPROTO_IP, IP_ADD_MEMBERSHIP,
  &mreq, sizeof(mreq))
```