RTP Payload format for Application and Desktop Sharing

Omer Boyaci & Henning Schulzrinne

November 18, 2008

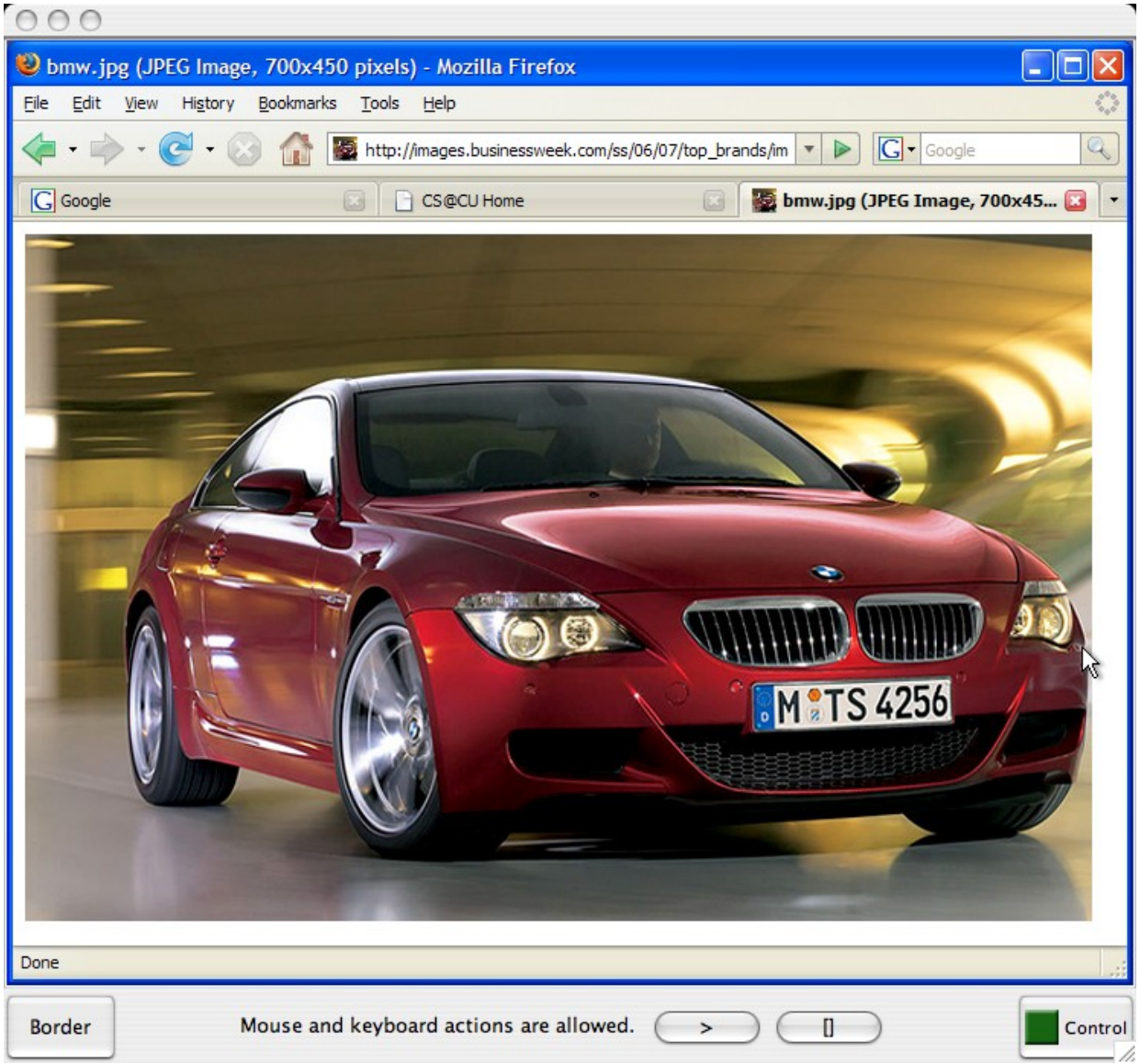
Application Sharing

- Sharing an application with multiple users
- There is only one copy of the application
- Participants do not need application itself
- Briefly, participants
  - receive screen updates
  - send keyboard and mouse events

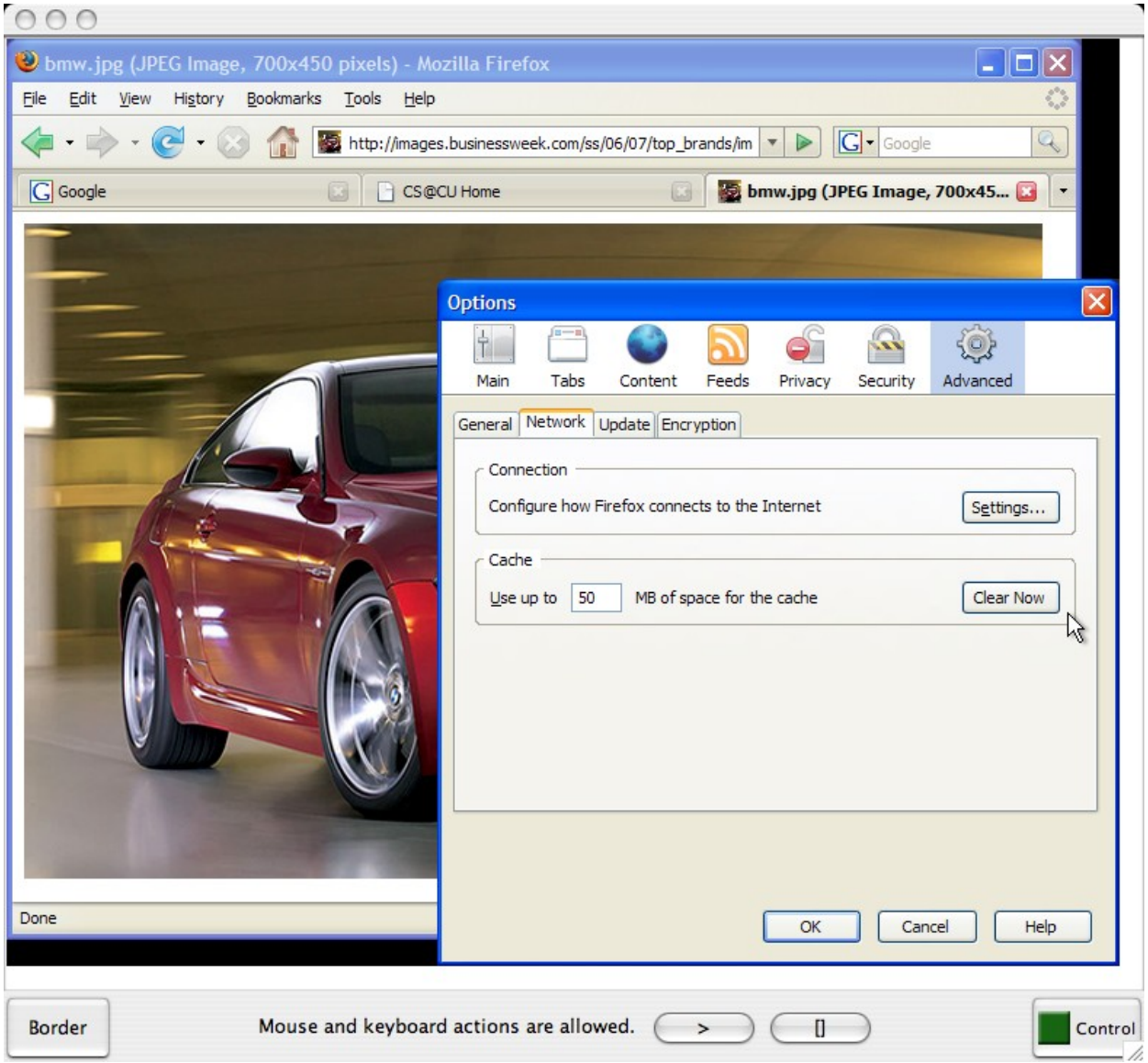
Terminology

- Application Host (AH)
- Participant
- Remoting protocol
- Human Interface Protocol (HIP)
- Different modes of remote access
  - Remote Desktop Connection
  - (Full or partial)Screen Sharing
  - Application Sharing

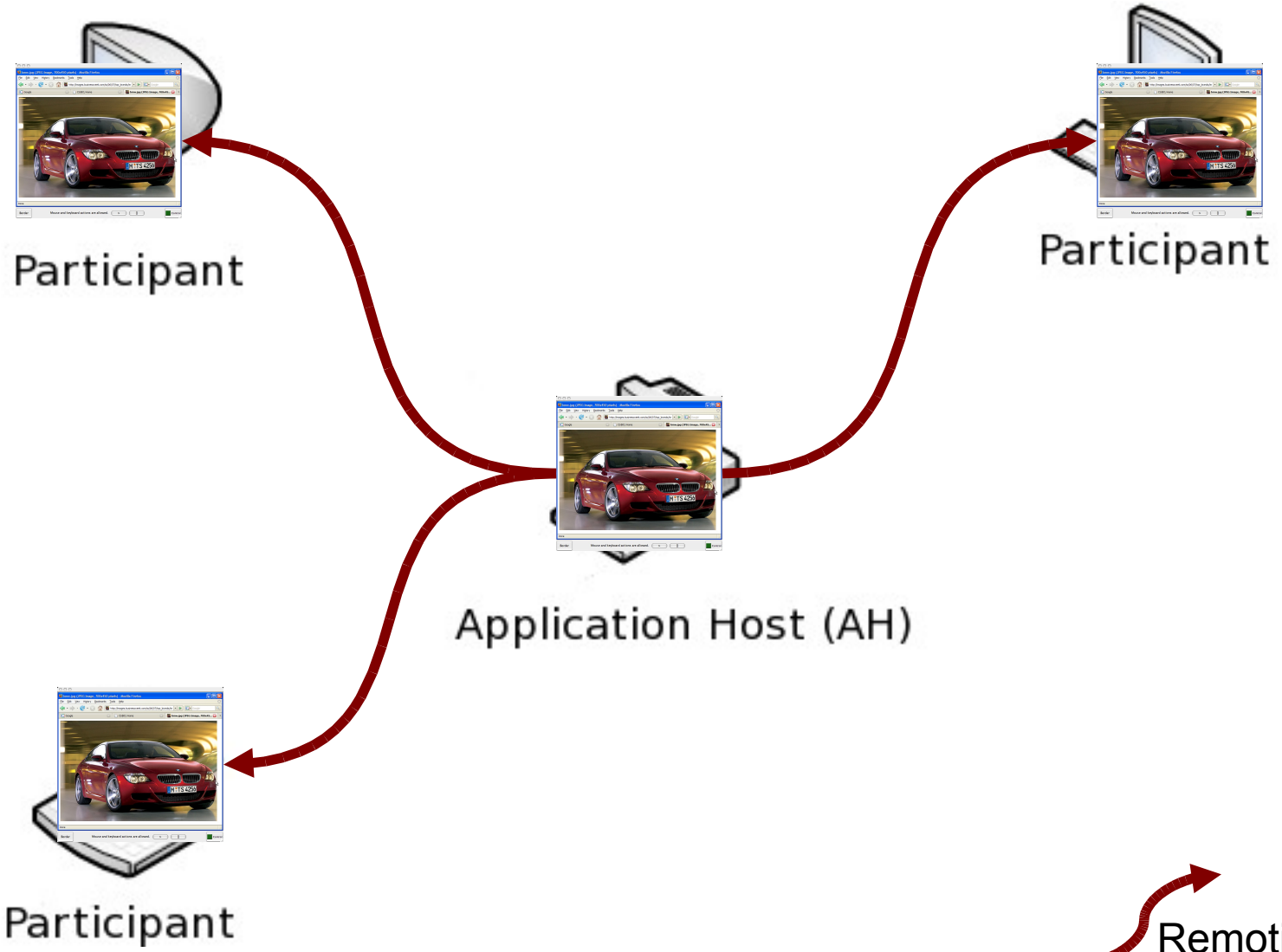
Screenshot


bmw.jpg (JPEG Image, 700x450 pixels) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://images.businessweek.com/ss/06/07/top_brands/im
Google
Google
CS@CU Home
bmw.jpg (JPEG Image, 700x45...
M TS 4256
Done
Border
Mouse and keyboard actions are allowed.
>
[]
Control

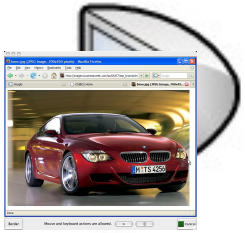
Screenshot (2)



Architecture



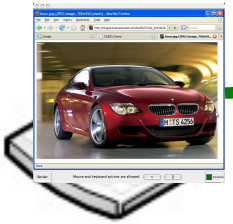
Architecture



Participant

Participant

Application Host (AH)



Participant

Human Interface Protocol (HIP)

Architecture

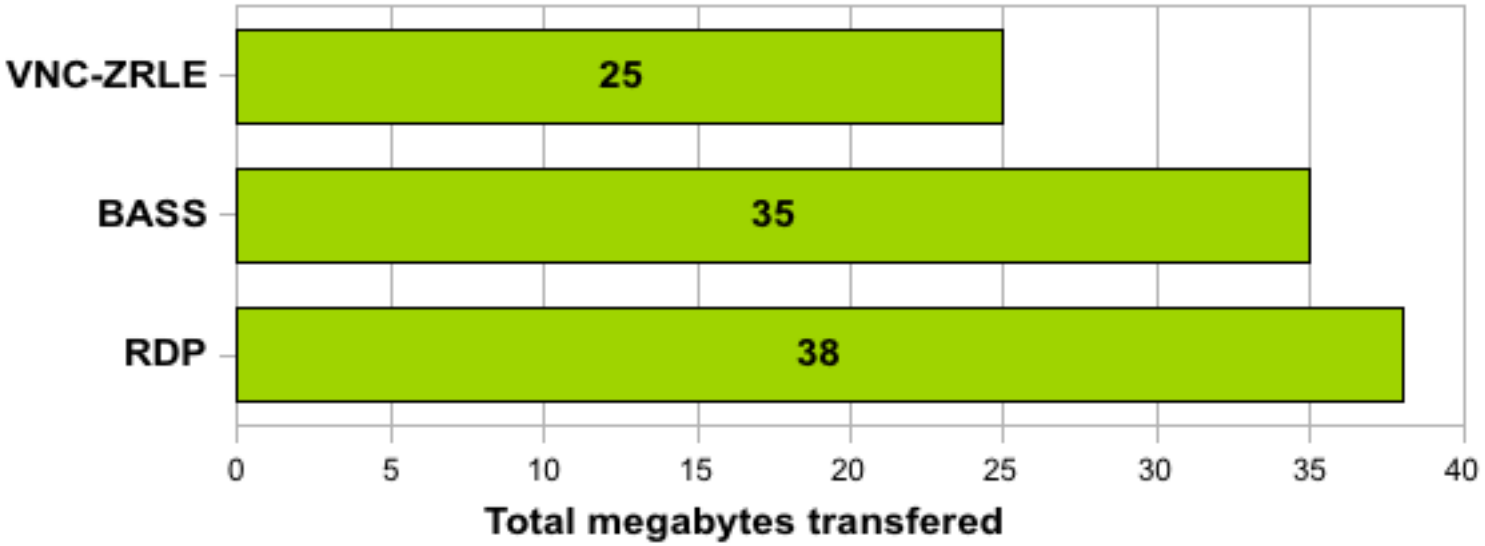
Binary Floor Control Protocol (BFCP) manages the ownership of AH side human interface devices.

Multimedia Support (Movies)

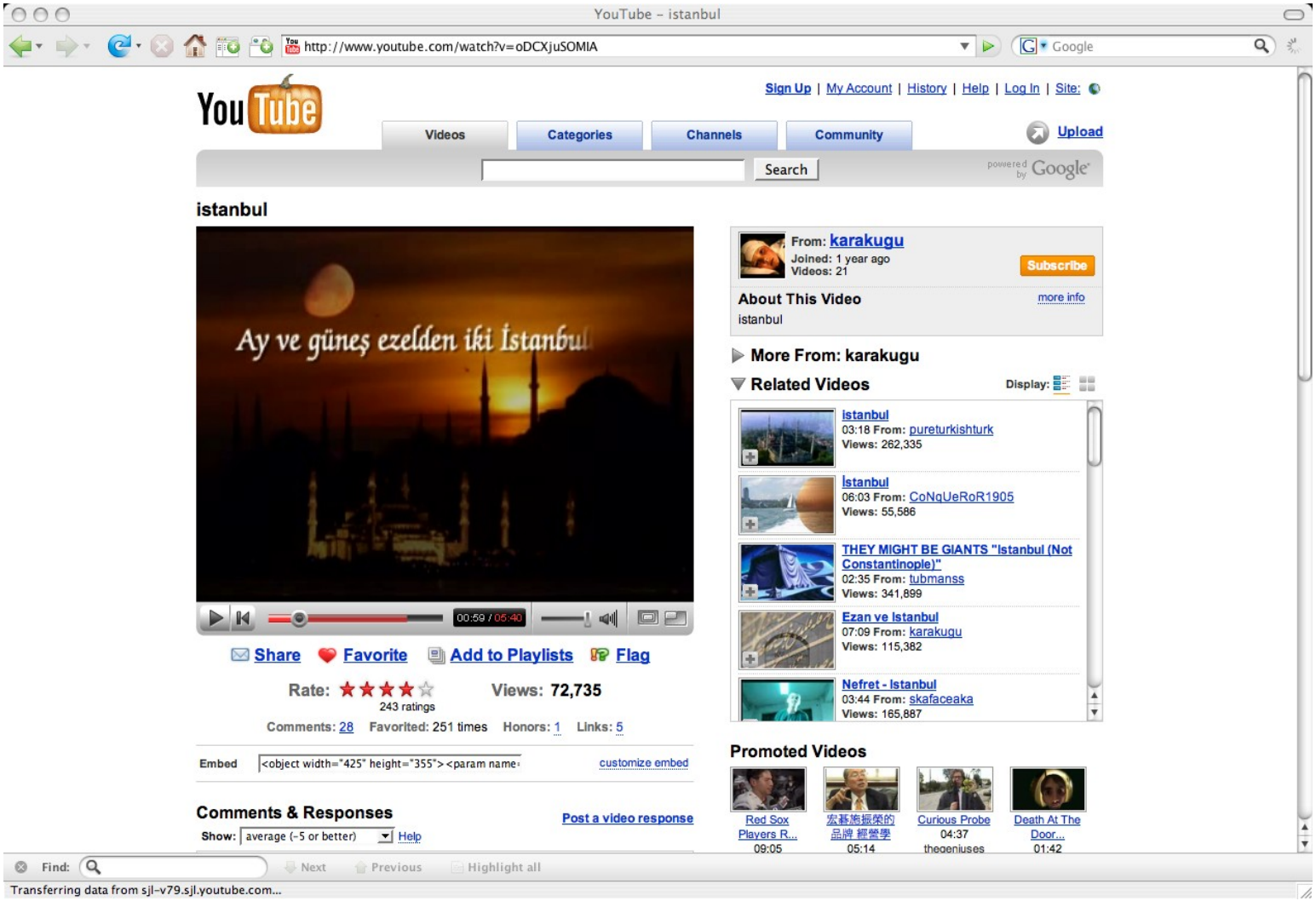
- Composite image comparing JPEG and PNG: notice artifacts in JPEG versus solid PNG background.

draft-boyaci-avt-png-00

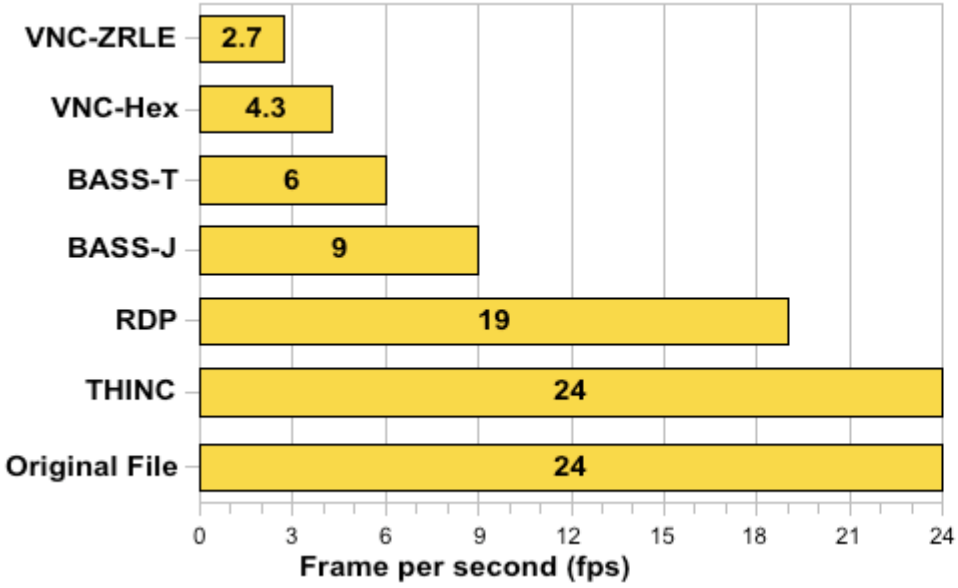
Comparison of Sharing Systems in terms of web page visiting performance

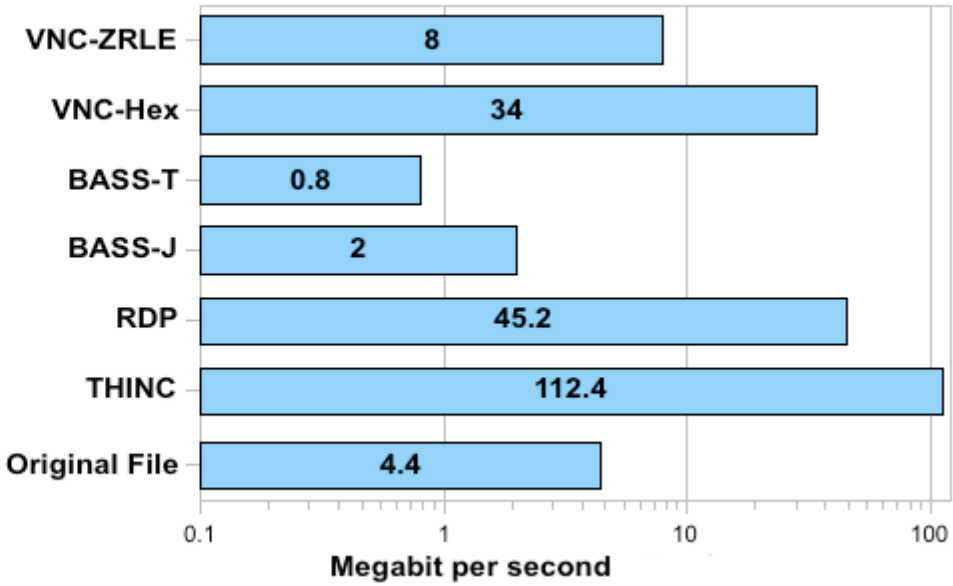


Multimedia Support (Movies)



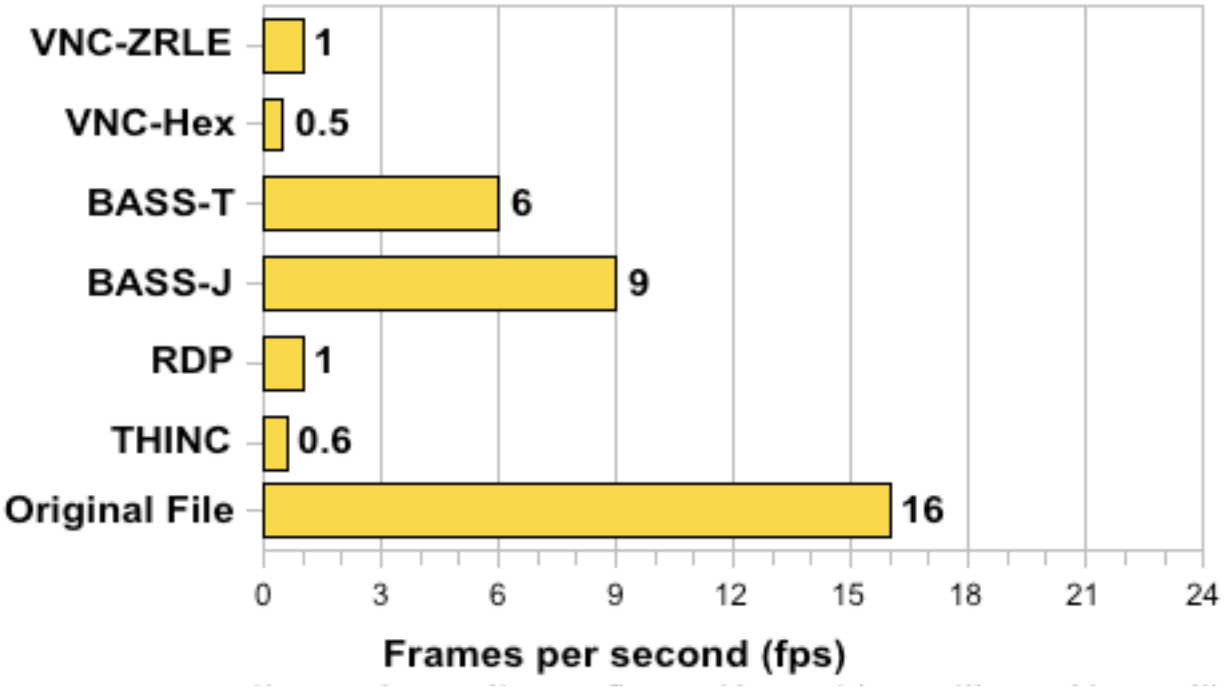
Comparison of Sharing Systems in terms of multimedia performance





Unlimited Bandwidth

Comparison of Sharing Systems in terms of multimedia performance



3Mb/s Bandwidth

Remoting Messages

- Application host (AH) to participants
  - WindowStateInfo
  - RegionUpdate
  - MoveRectangle
  - MousePointerInfo
- Participants to AH (NACK messages)
  - PLI (Picture Loss Indication)
  - NACK request

HIP Messages

- MousePressed
- MouseReleased
- MouseMoved
- MouseWheelMoved
- KeyPressed
- KeyReleased
- KeyTyped

VNC problems

- Client-pull based
- No multicast support
- Same encoding for all updates
- CPU usage increases for each new user

TeleTeachingTool (VNC based)

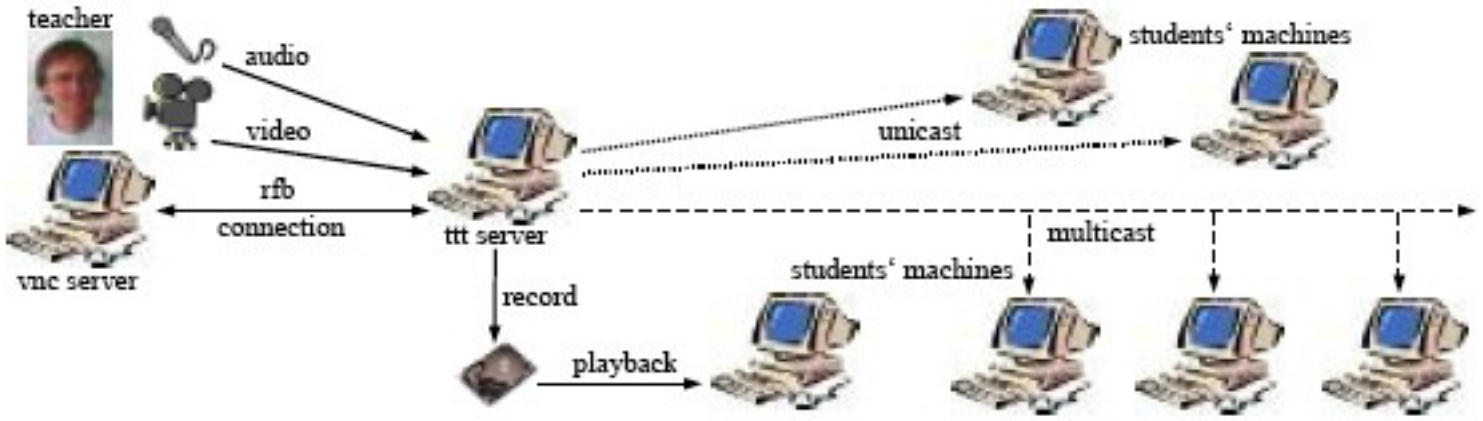


Figure 2: TeleTeachingTool environment overview

Problems

Modified VNC protocol and clients (TTT server is pushing updates)

No compression (only hextile encoding) because packets can get lost

Open Issues

- Transport Protocol
  - RTP
  - MSRP
  - Custom Made

Open Issues

- How to do retransmission requests for UDP clients
- **Current proposal**
  - NACK-based solution
  - NACK suppression to prevent floods
  - RTP-library level retransmissions

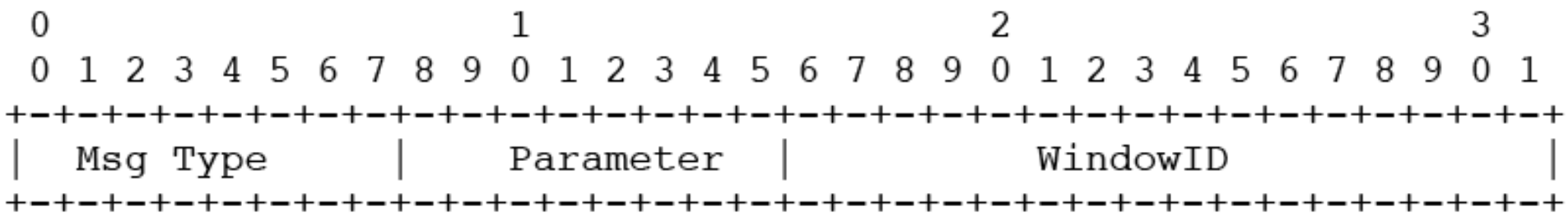
Current Internet Drafts

draft-boyaci-avt-app-sharing-00
draft-boyaci-avt-png-00

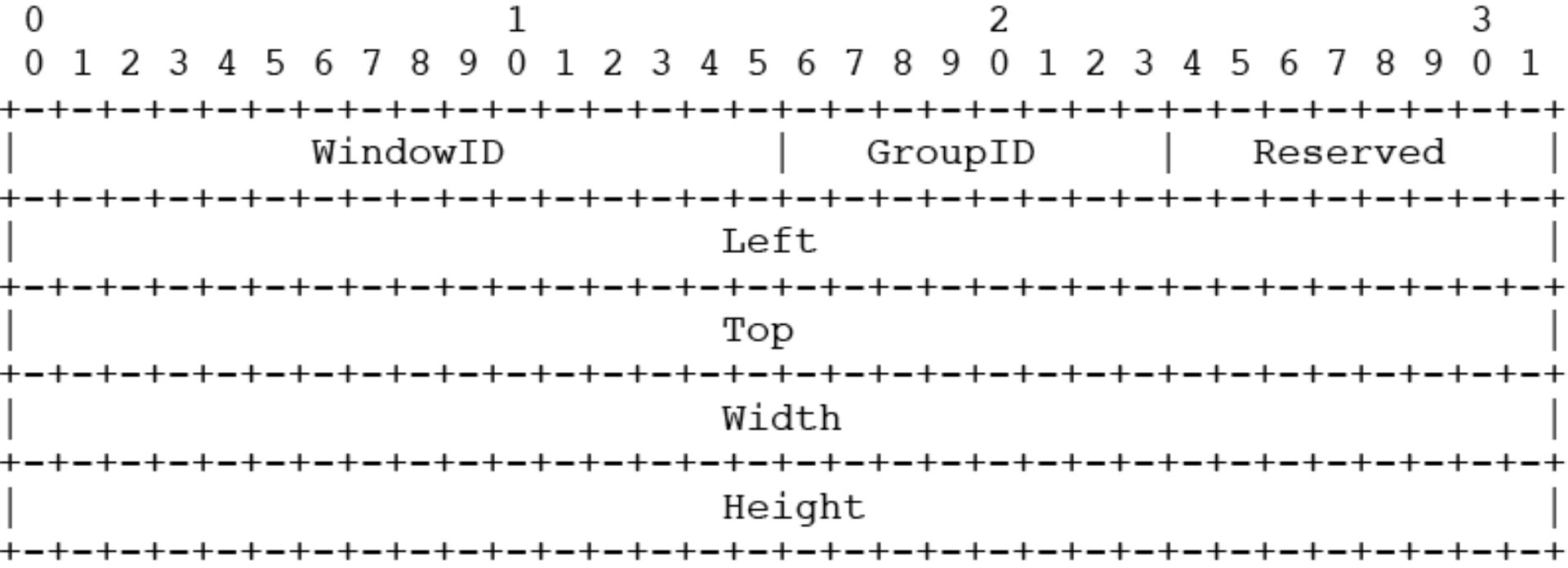
Backup Slides

Protocol Message Structures

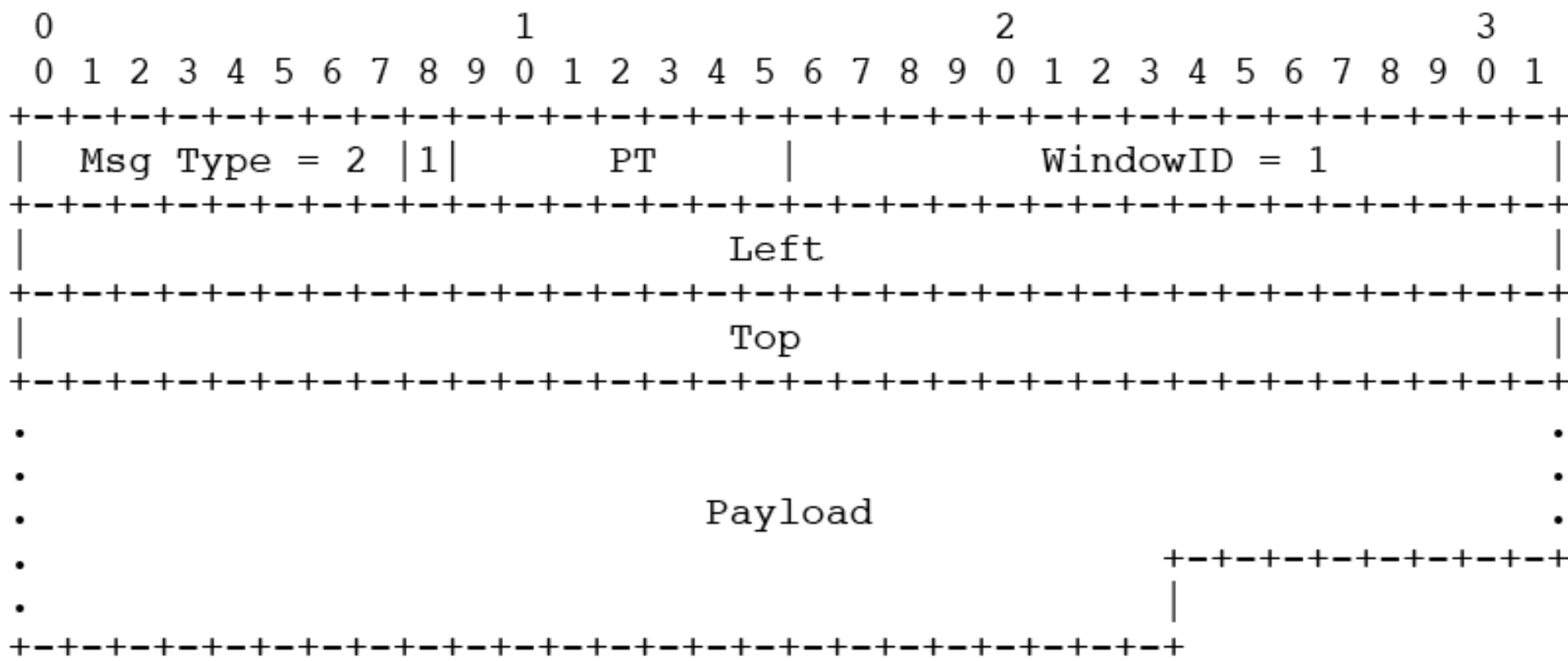
Common remoting/hip Header



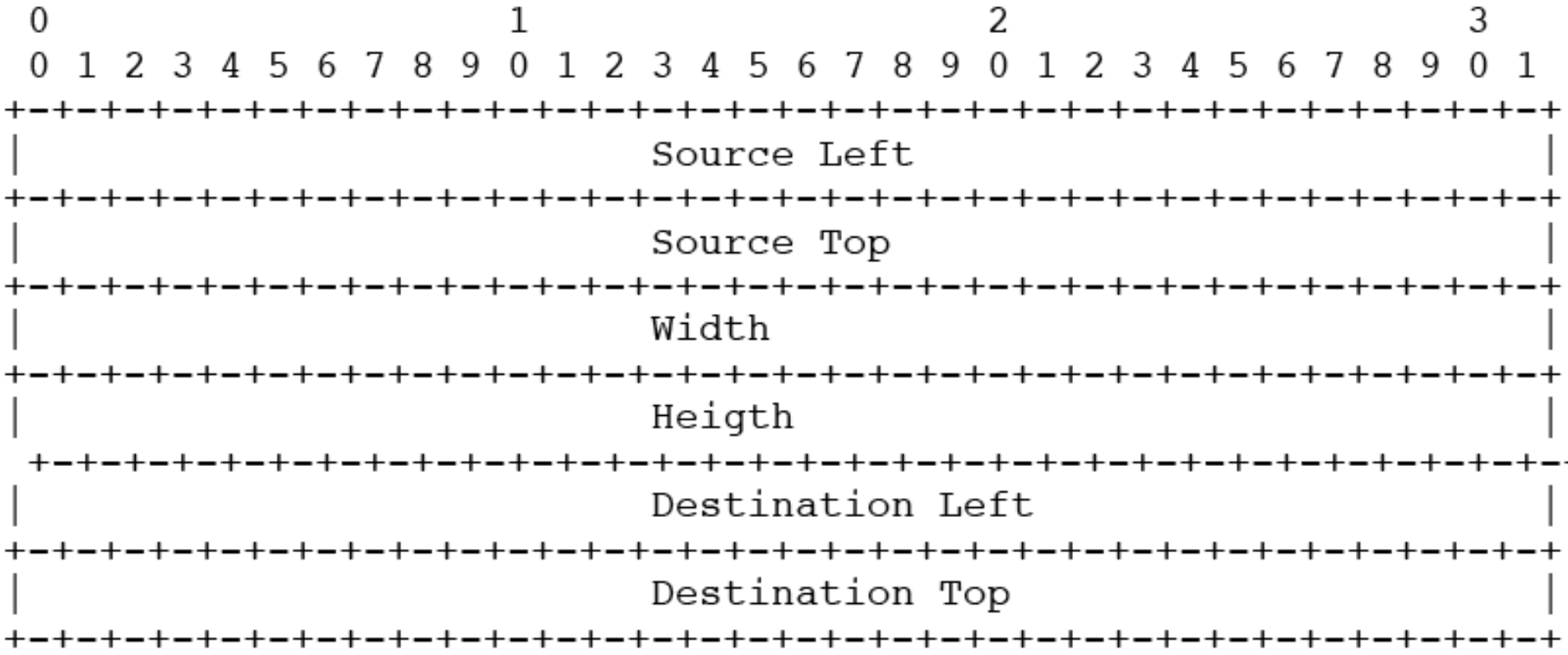
A Window Record



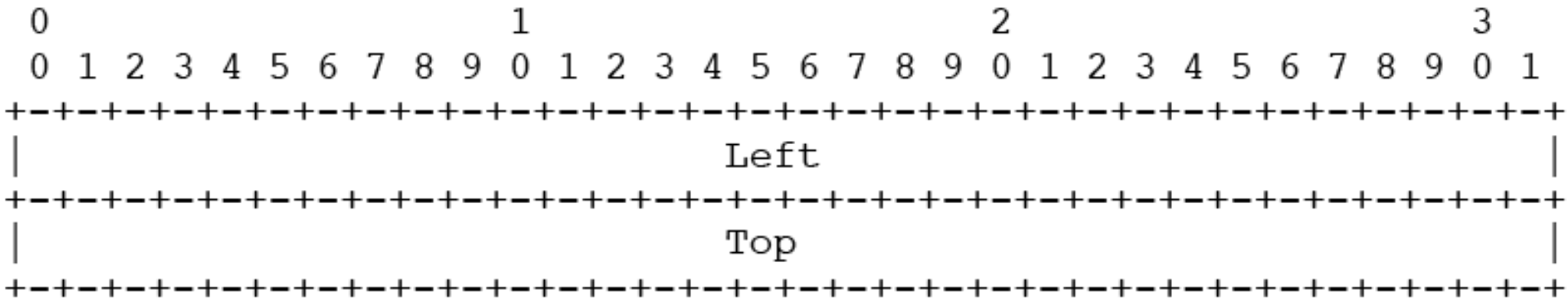
A Region Update Message



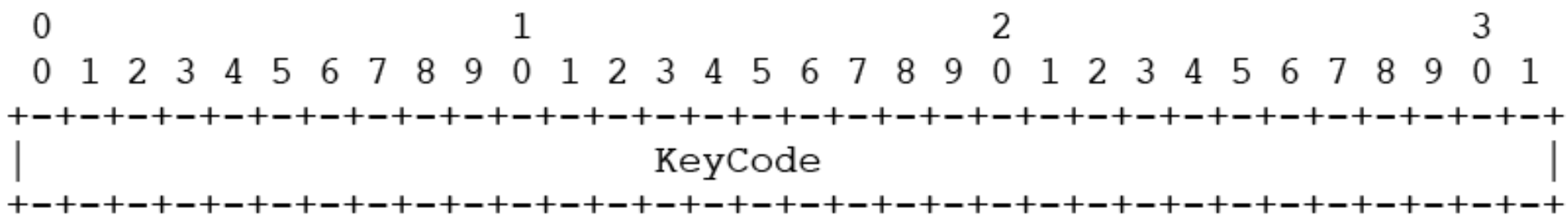
The MoveRectangle Message



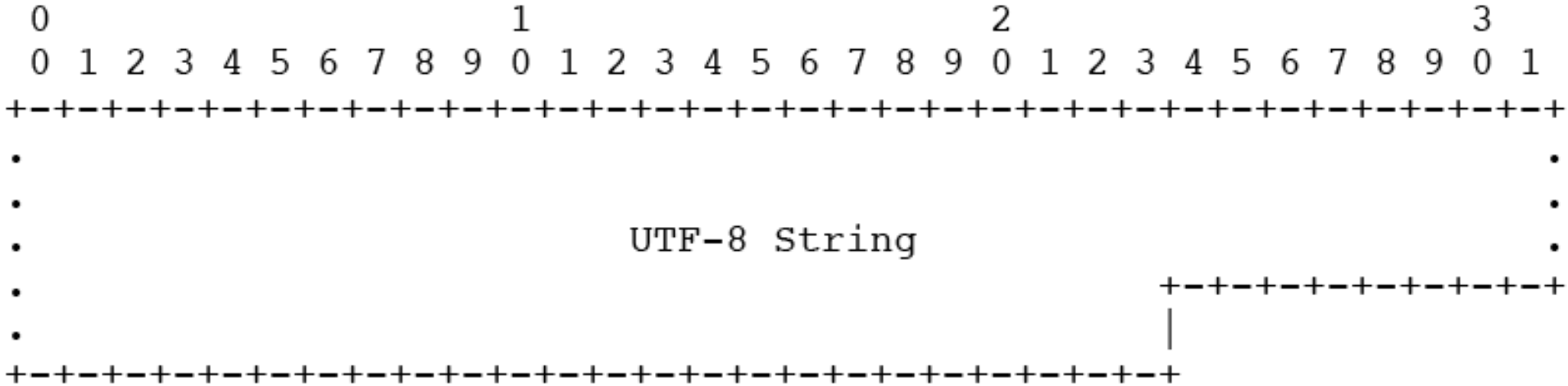
The MousePressed Message



The KeyPressed Message

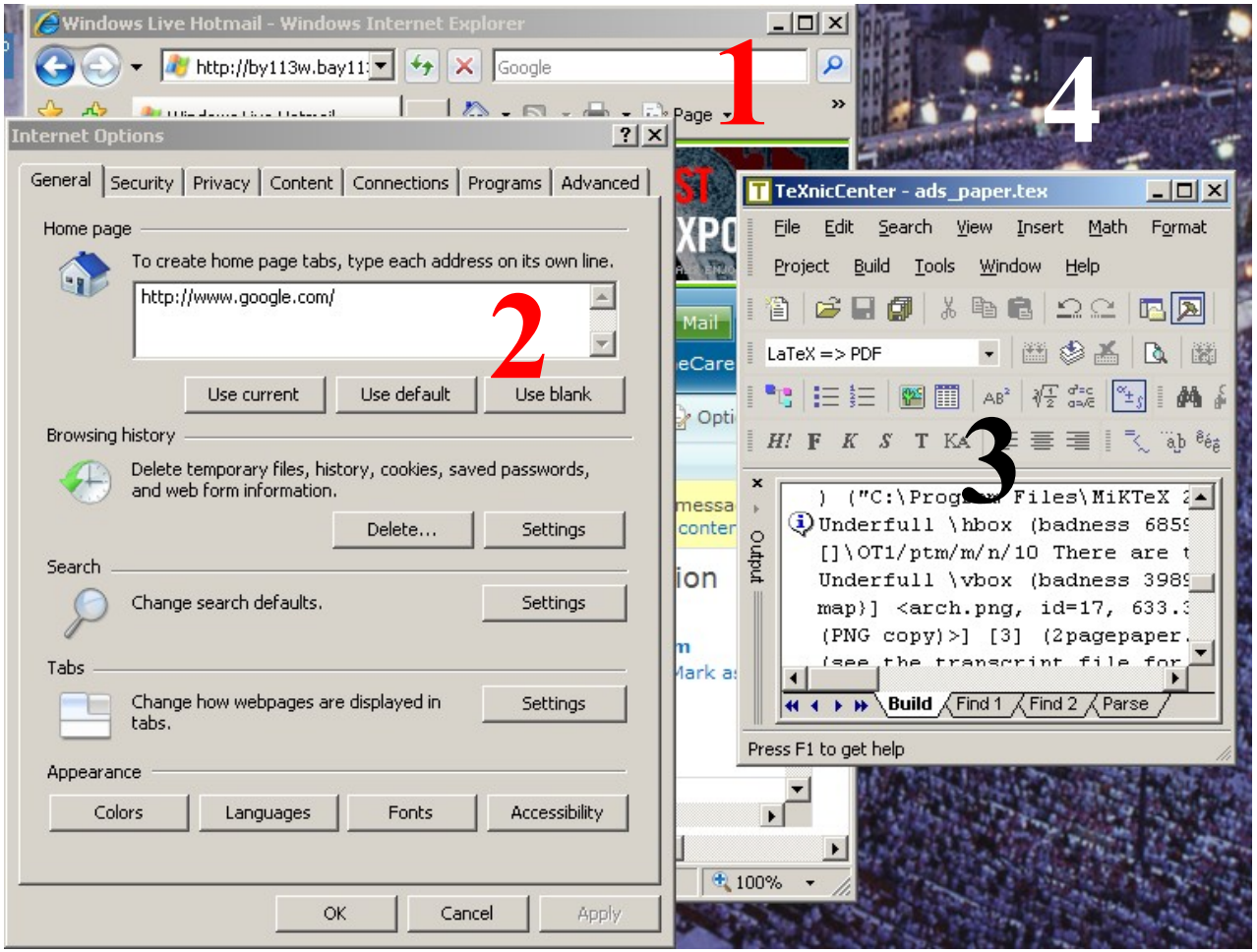


The KeyTyped Message

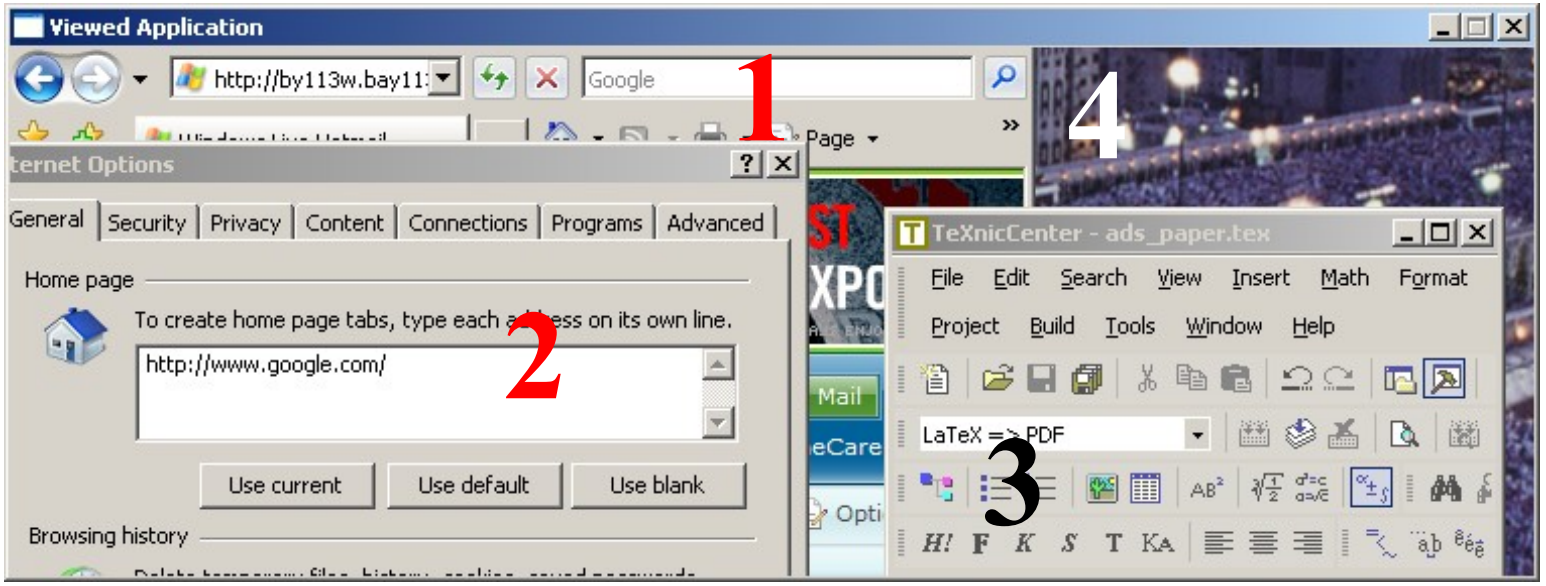


Comparison of Sharing Systems

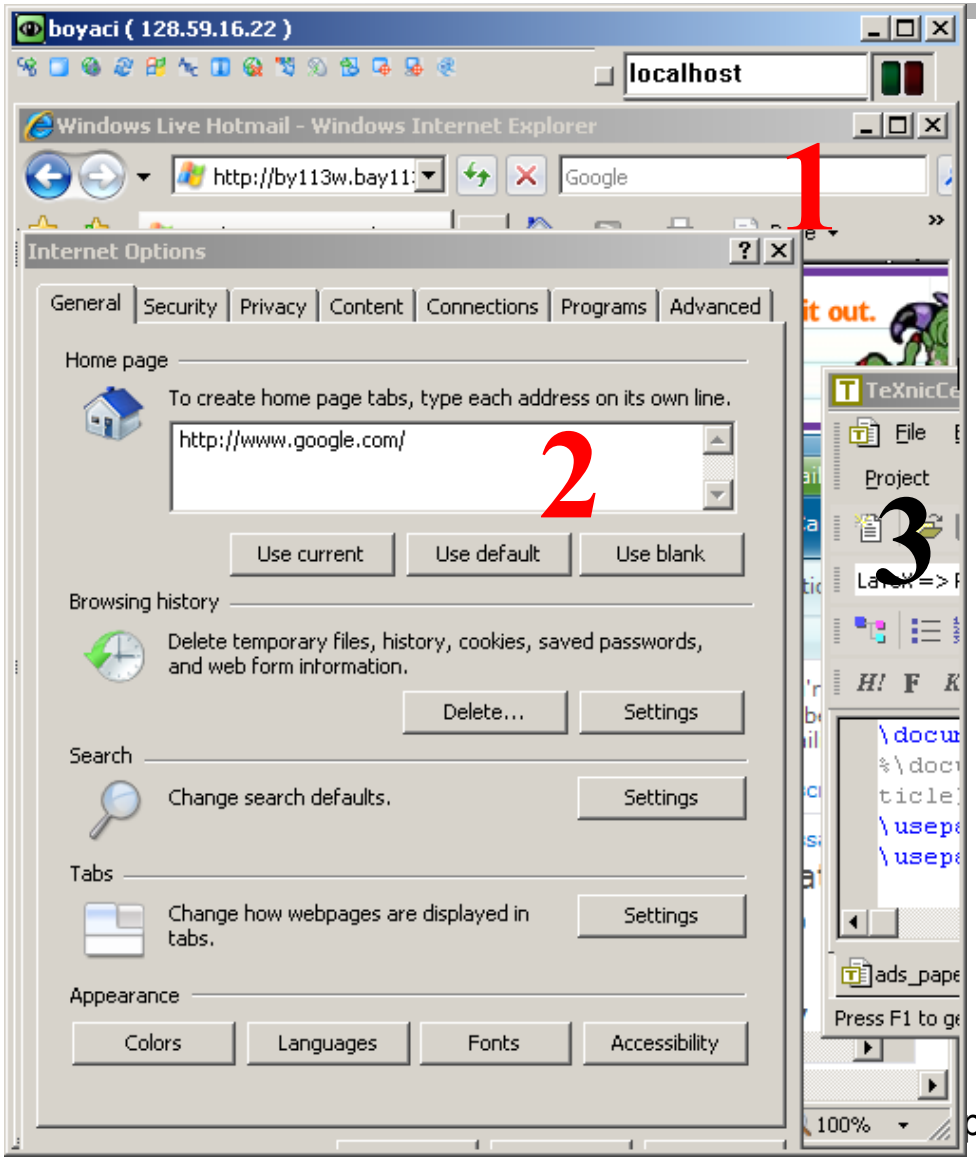
Screenshot (Overlapped Windows)



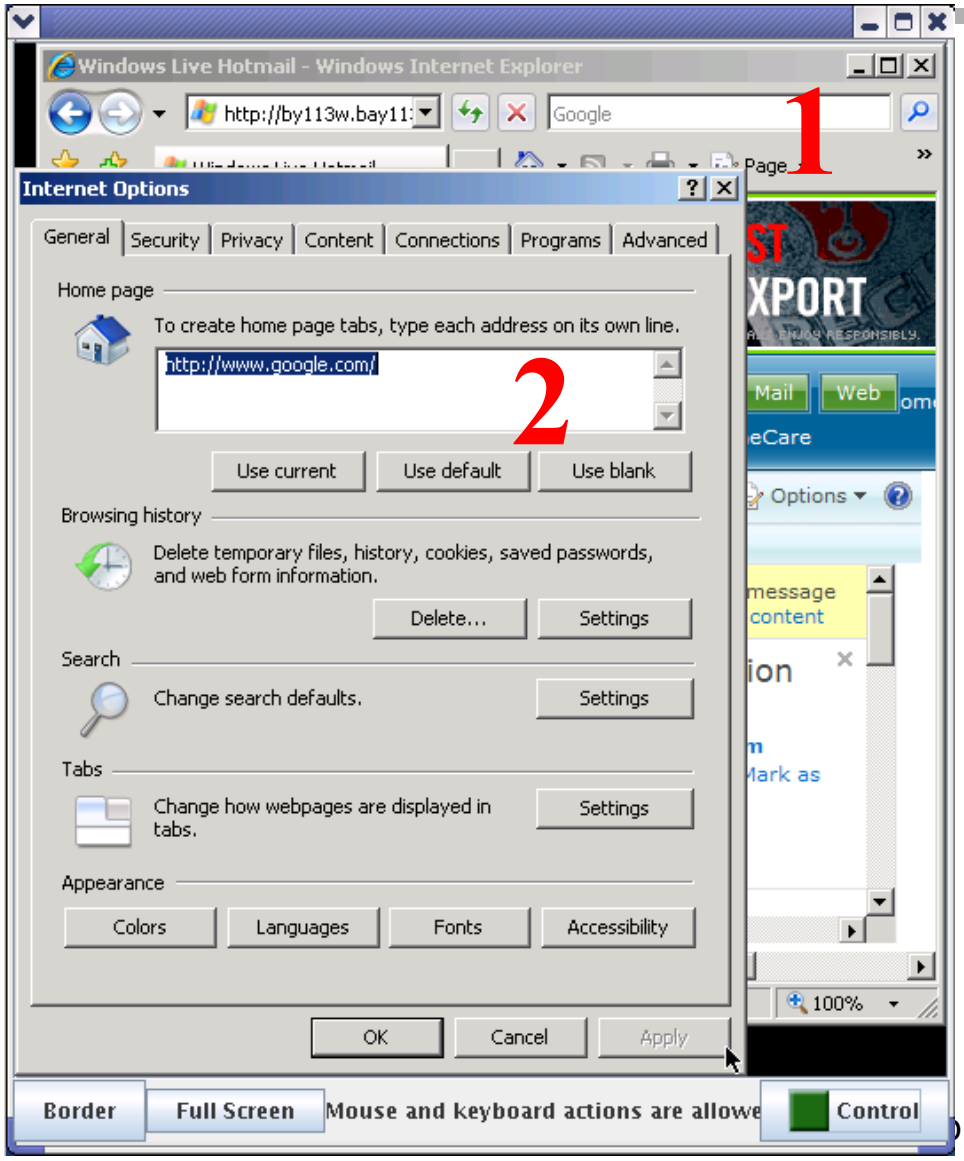
Screenshot (Multicast App. Tool)



Screenshot (Ultra VNC)



Screenshot (BASS)



Comparison of Sharing Systems

System	Pull/Push	Technique	Multicast	App Sharing	Movies
VNC	Pull	Mirror Driver		Region	
TTT: TeleTeachingTool(VNC)	Pull+Push	Mirror Driver		Region	
SharedAppVNC	Pull	Polling		Region	
MAST	Push	Polling		Region	
RDP	Push	Mirror Driver			High B/W
THINC	Push	Mirror Driver			High B/W
Distributed Workspace	Push	Mirror Driver			High B/W

VNC problems

- Client-pull based
- No multicast support
- Same encoding for all updates
- CPU usage increases for each new user

TeleTeachingTool (VNC based)

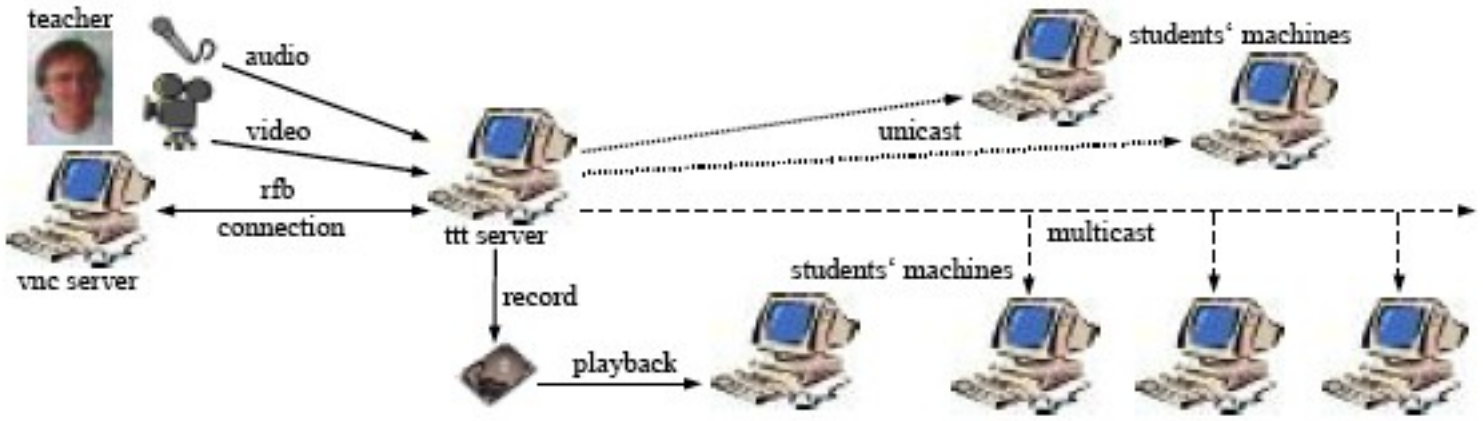


Figure 2: TeleTeachingTool environment overview

Information about BASS

Supported Platforms/OS

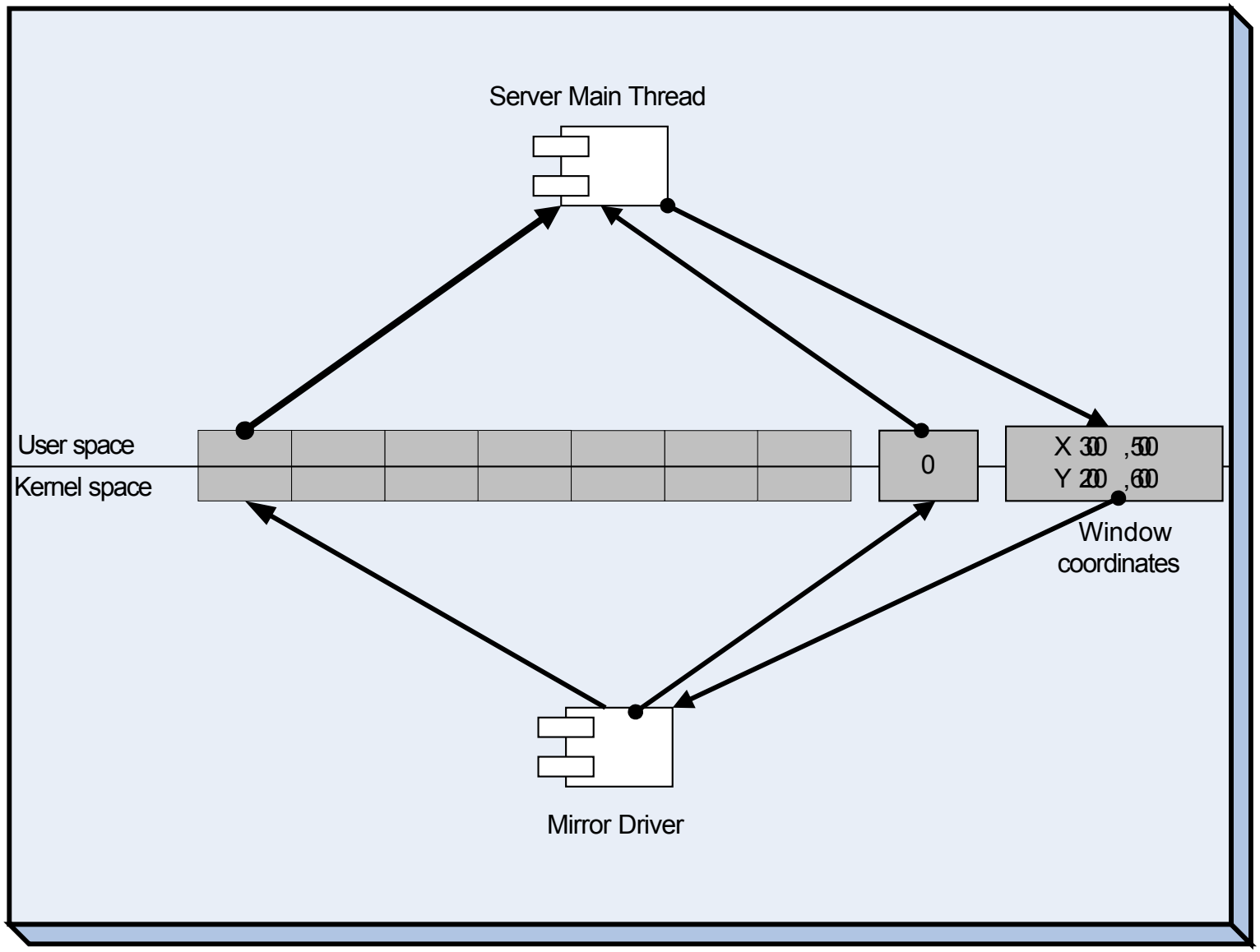
	Server	Client*
Windows 	+	+
*nix 	-+	+
Mac OS X 	-+	+

*Client is Java based.

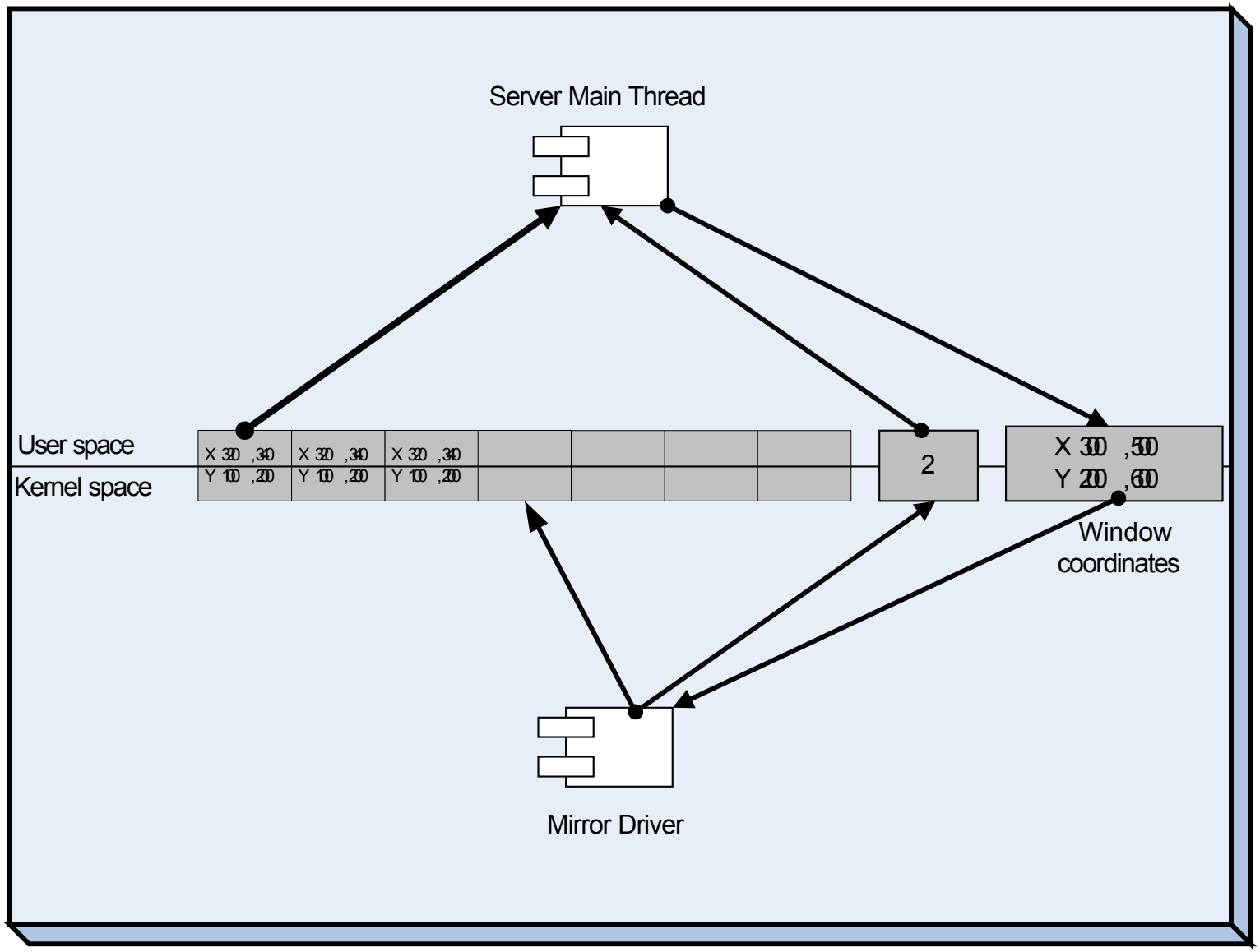
Client (Viewer) Architecture

- Client can
  - Connect to server
  - Wait for incoming connections
- Client supports
  - TCP
  - UDP (+Multicast)

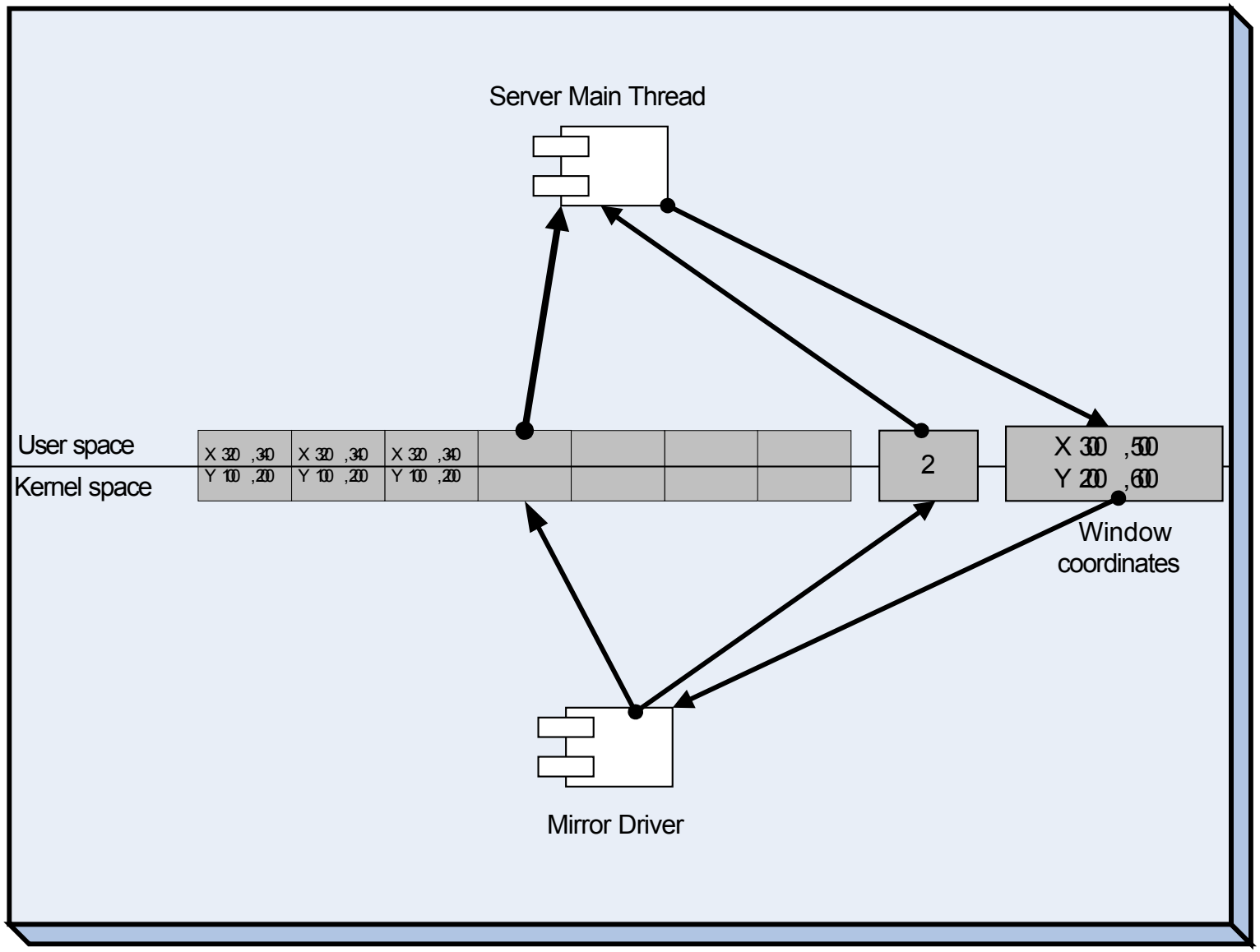
BASS Windows Server Architecture



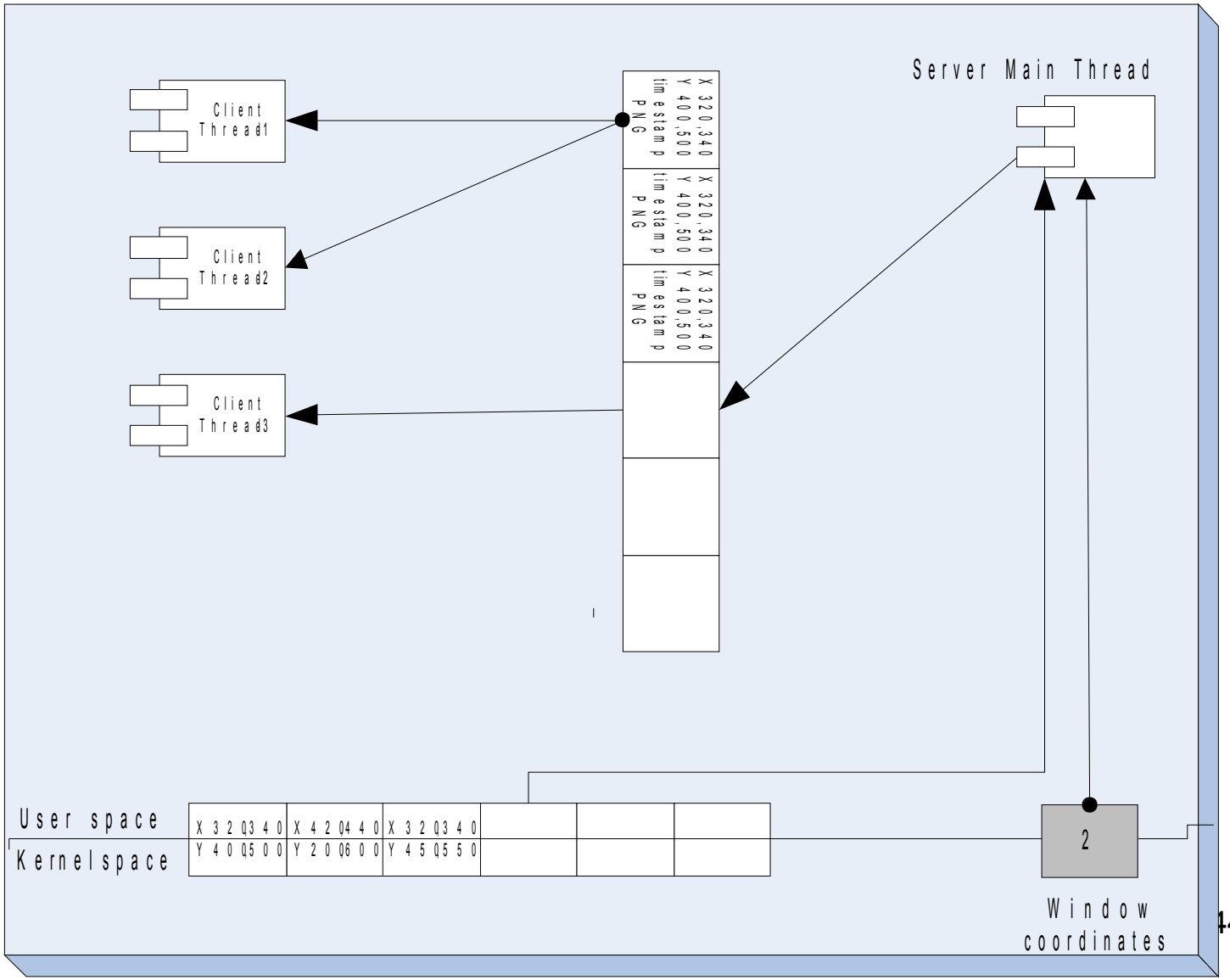
BASS Windows Server Architecture



BASS Windows Server Architecture



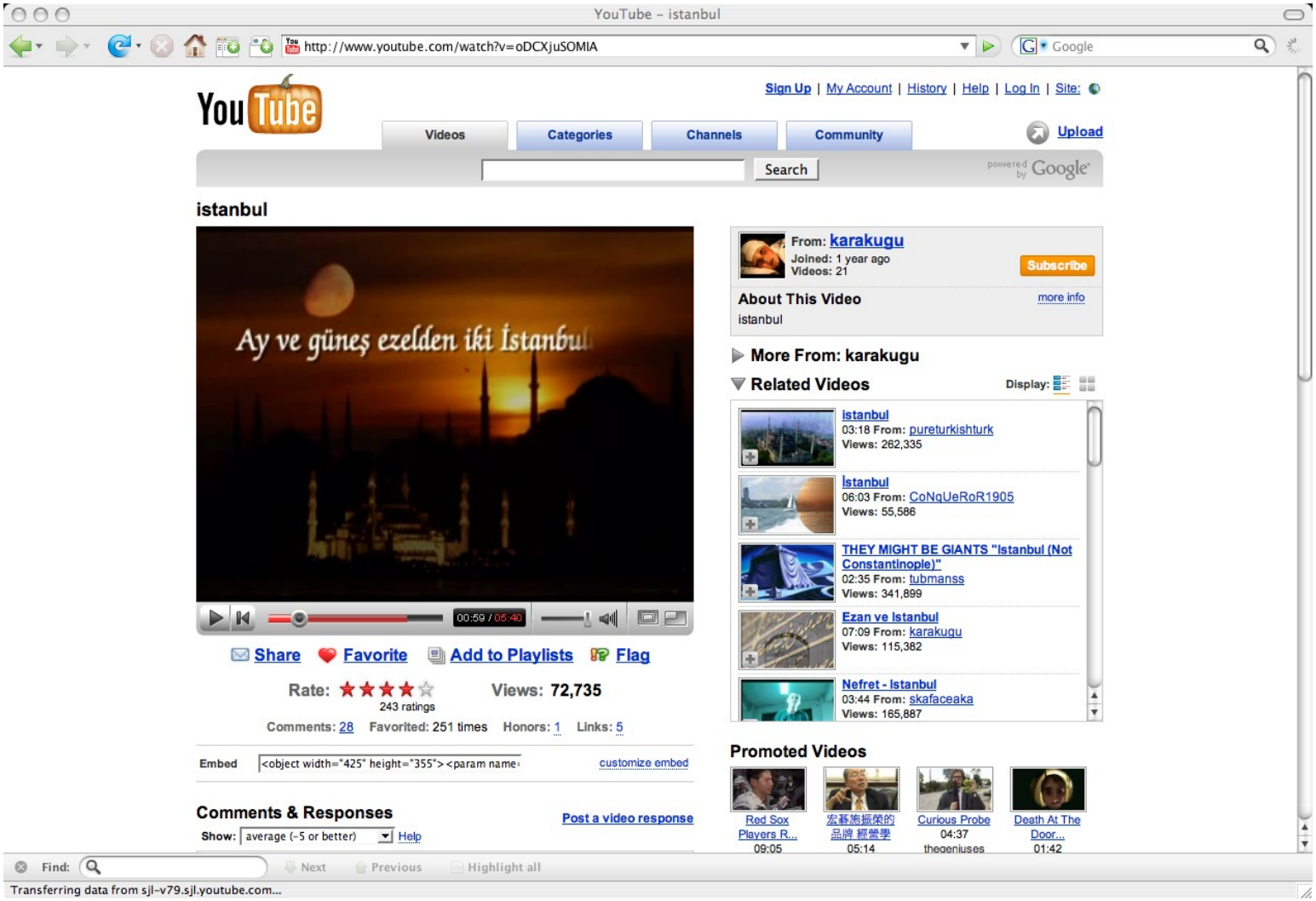
BASS Windows Server Architecture



Challenges

- Different client bandwidths/speeds
- Late Joiner
- The effects of packet loss
- Reliable multicast

Multimedia Support (Movies)



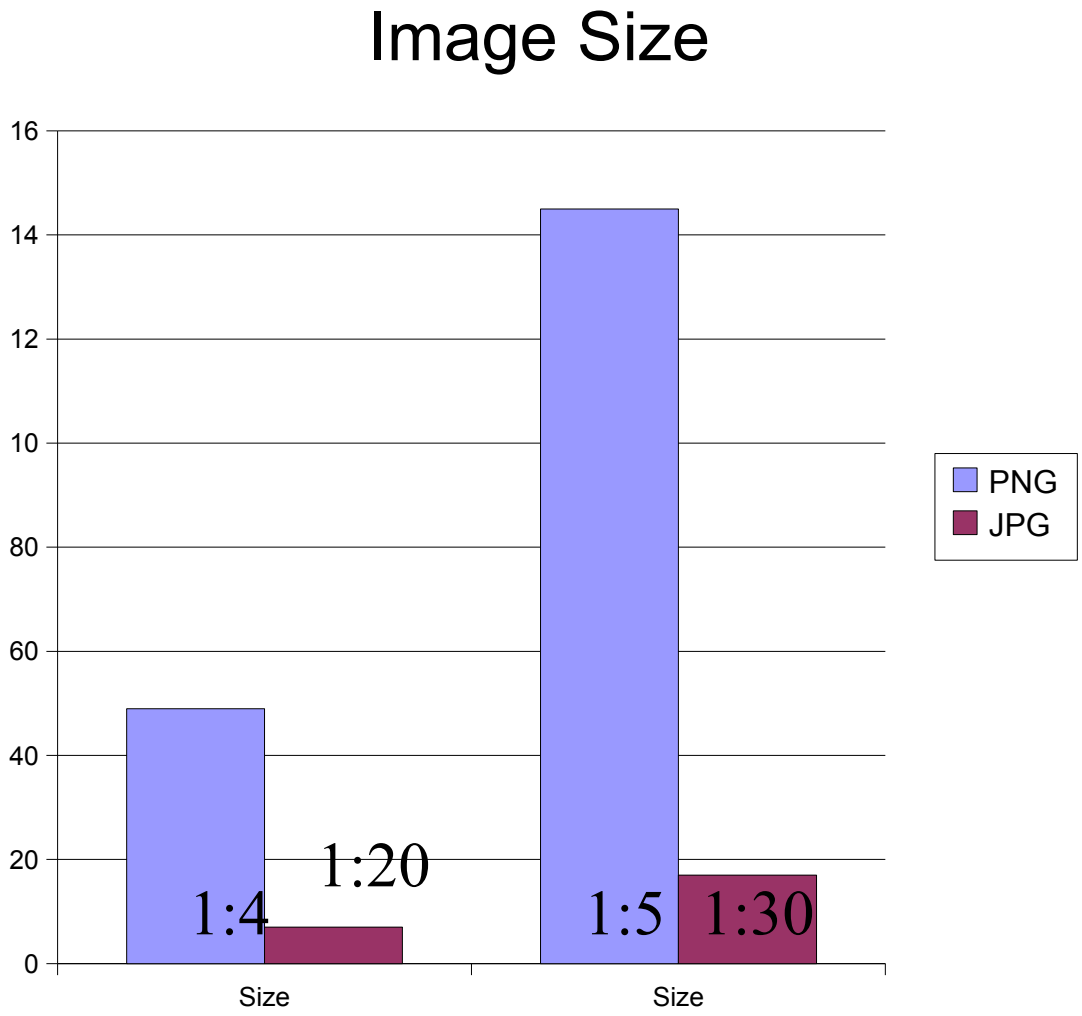
Multimedia Support (Movies)

- Our system uses PNG to compress and transmit the region updates
- PNG is lossless and effective for computer generated images but ineffective for real world captures like pictures or movies
- JPG is more suitable for photographic images
- However, JPG is lossy and not effective for computer generated images (text, line, shapes,...)
- Our system should use both

Multimedia Support (Movies)

- Composite image comparing JPEG and PNG: notice artifacts in JPEG versus solid PNG background.

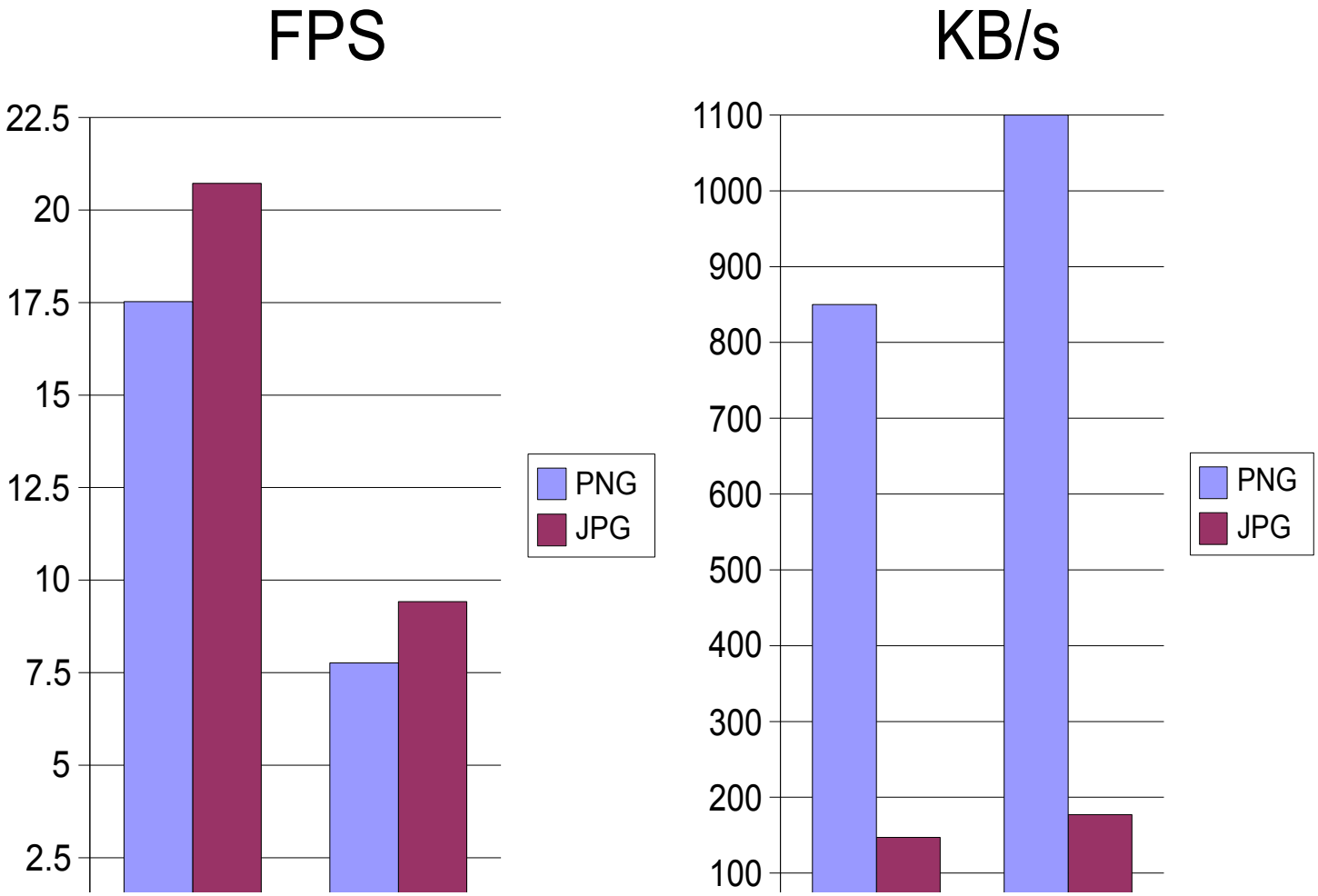
Multimedia Support (PNG vs JPG)



Size x
360x150
162K

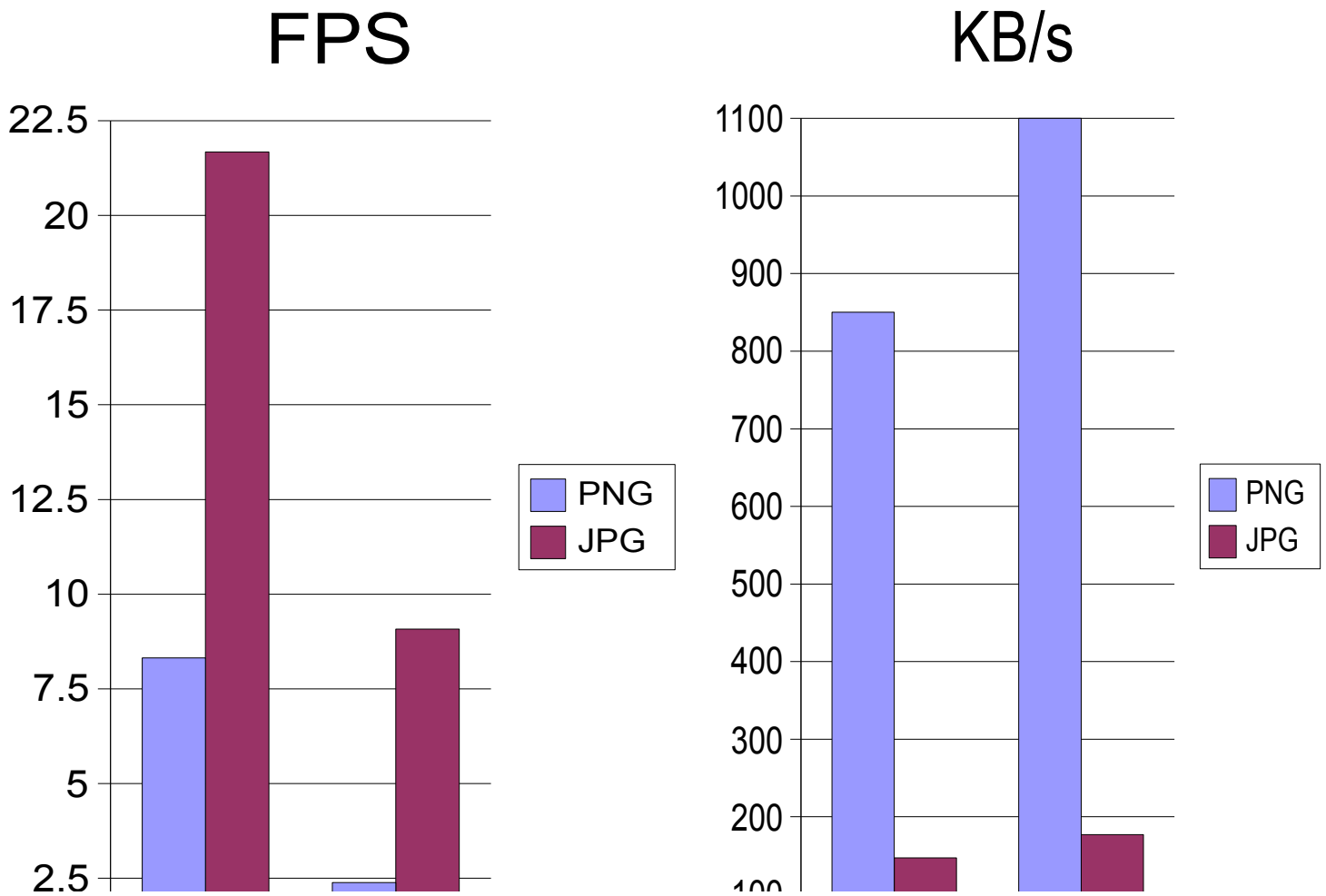
Size 4x
720x300
648K

Multimedia Support (PNG vs JPG)



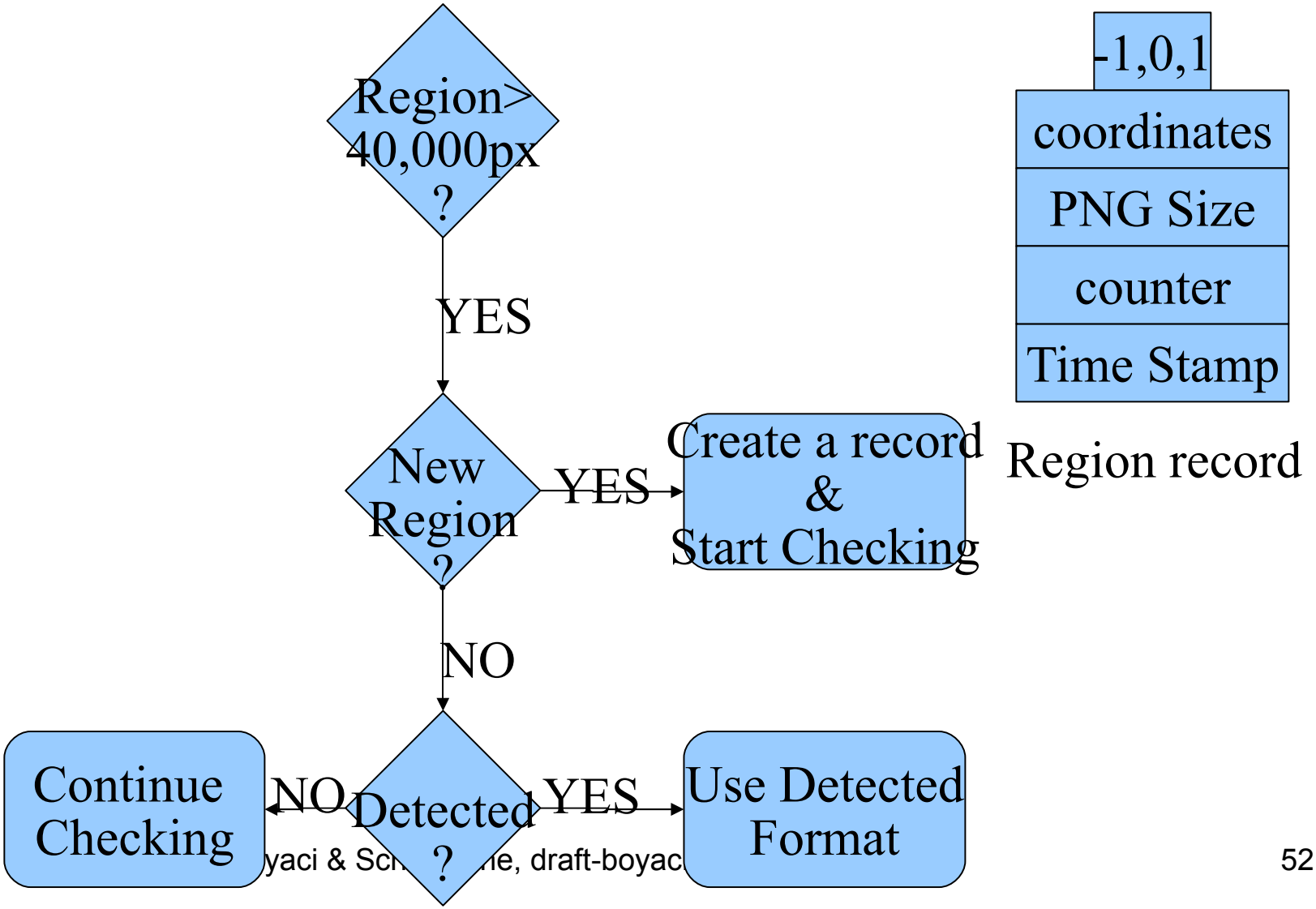
Ethernet (60Mb/s)

Multimedia Support (PNG vs JPG)



Wireless (4Mb/s)

PNG/JPG Detection Algorithm

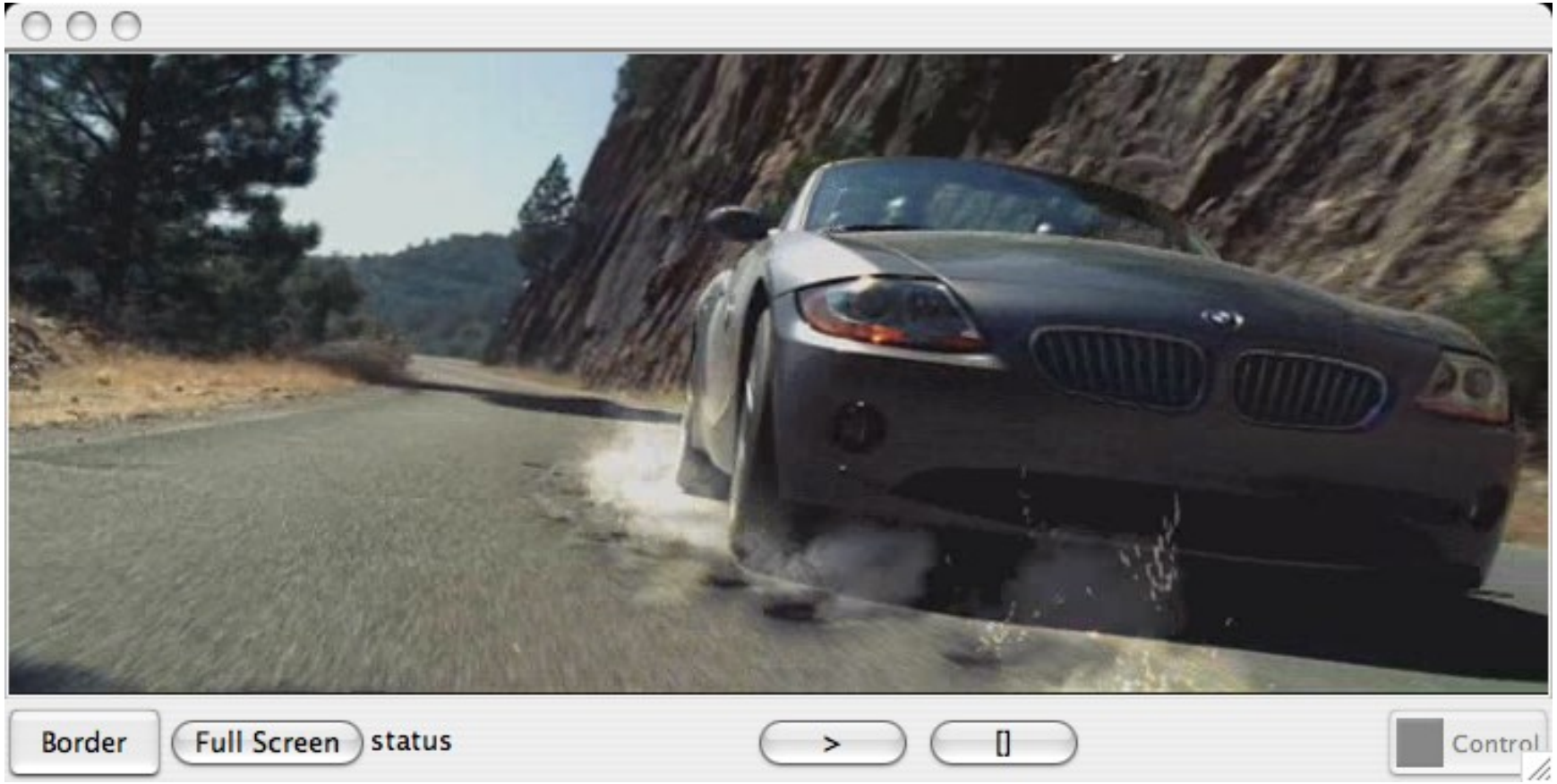


Sharing a Movie (Media Player)

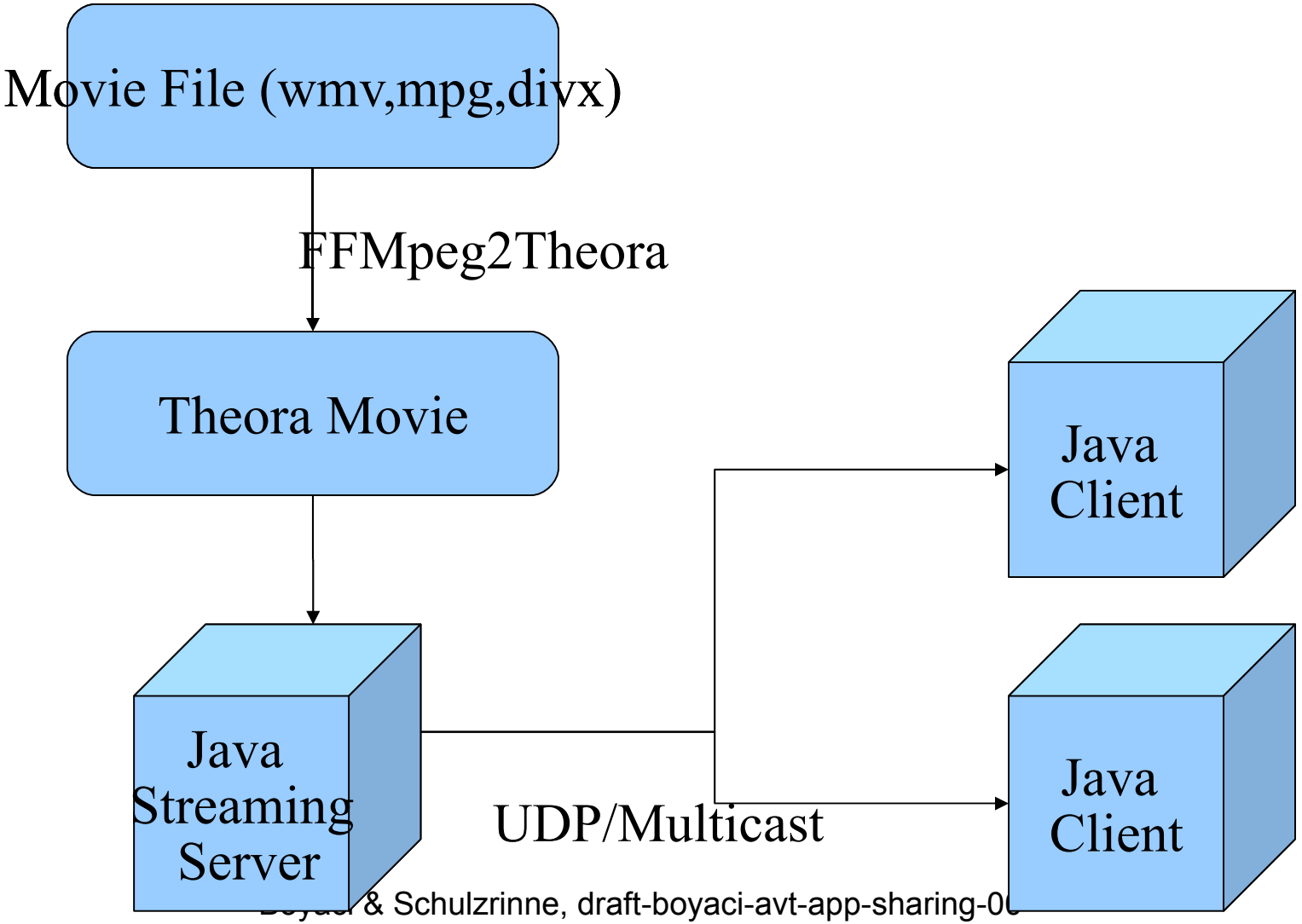
Sharing a Movie File

- Capturing from the Frame Buffer is expensive.
- Instead
  - Transcode the movie to Theora beforehand
- Then stream the theora directly to participants
  - Java client supports theora playback
- Negligible CPU usage during playback on the server
  -

Sharing a Movie (Our Method)



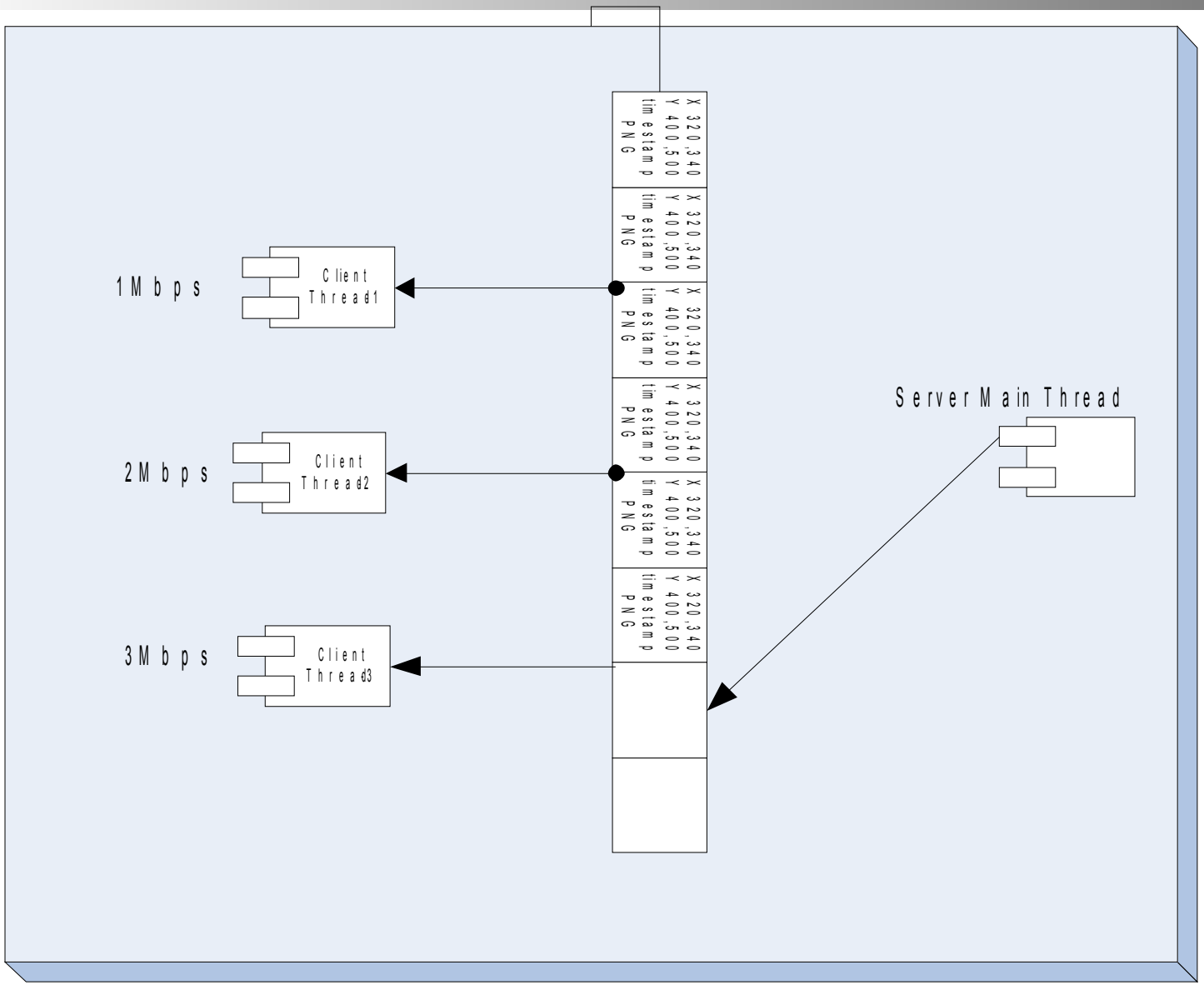
Sharing a Movie File



Challenges

- Different client bandwidths/speeds
- Late Joiner
- The effects of packet loss
- Reliable multicast

Different Client Bandwidths/Speeds



Different Client Bandwidths/Speeds

- Possible Solutions
  - Slowest one
  - Average speed
  - Fastest one

Different Client Bandwidths/Speeds

- Possible Solutions
  - Slowest one
    - Problem: Penalize everybody except the slowest
  - Average speed
  - Fastest one

Different Client Bandwidths/Speeds

- Possible Solutions
  - Slowest one
    - Problem: Penalize everybody except the slowest
  - Average speed
    - Possible solution
  - Fastest one

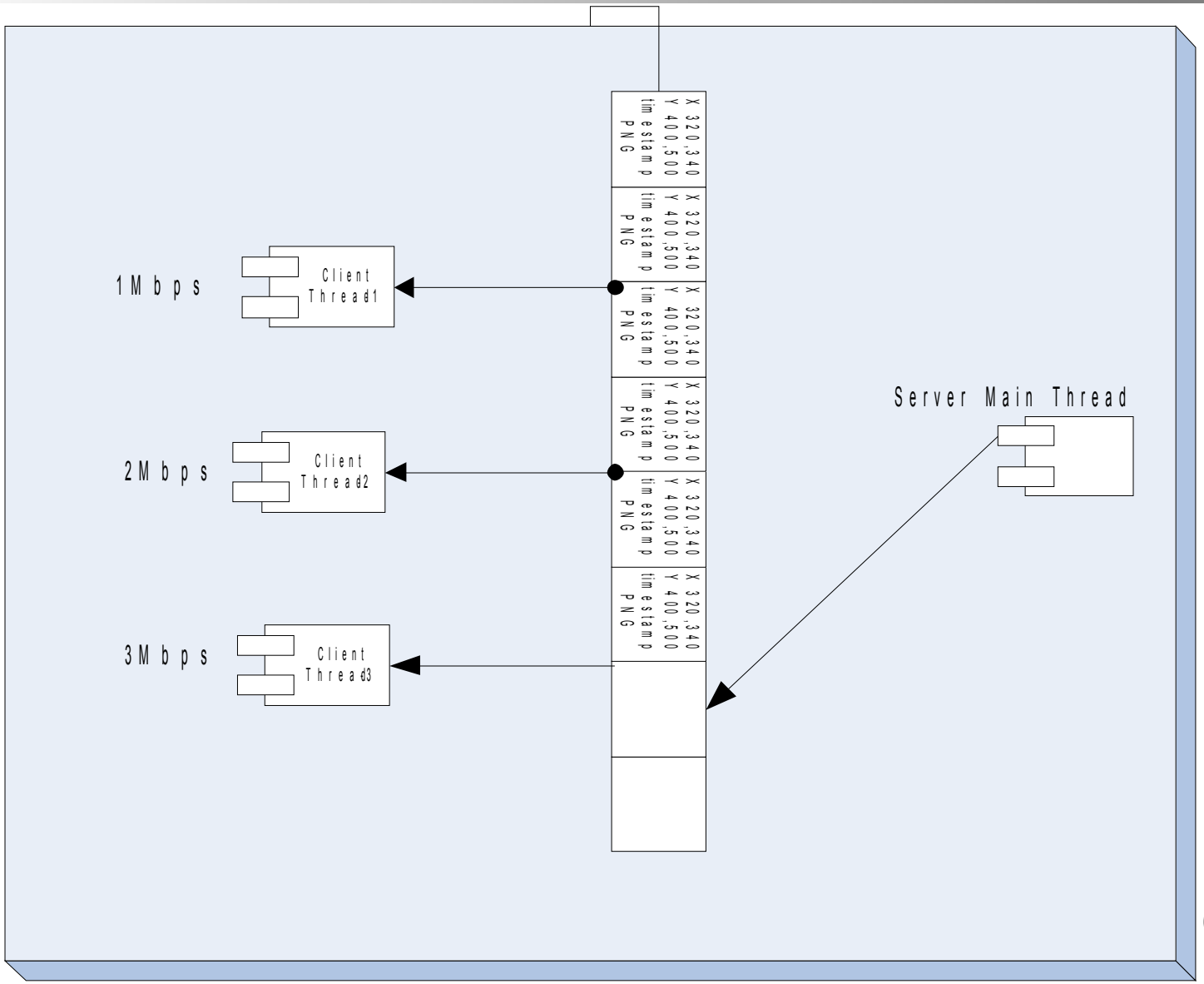
Different Client Bandwidths/Speeds

- Possible Solutions
  - Slowest one
    - Problem: Penalize everybody except the slowest
  - Average speed
    - Possible solution (Can we do better?)
  - Fastest one

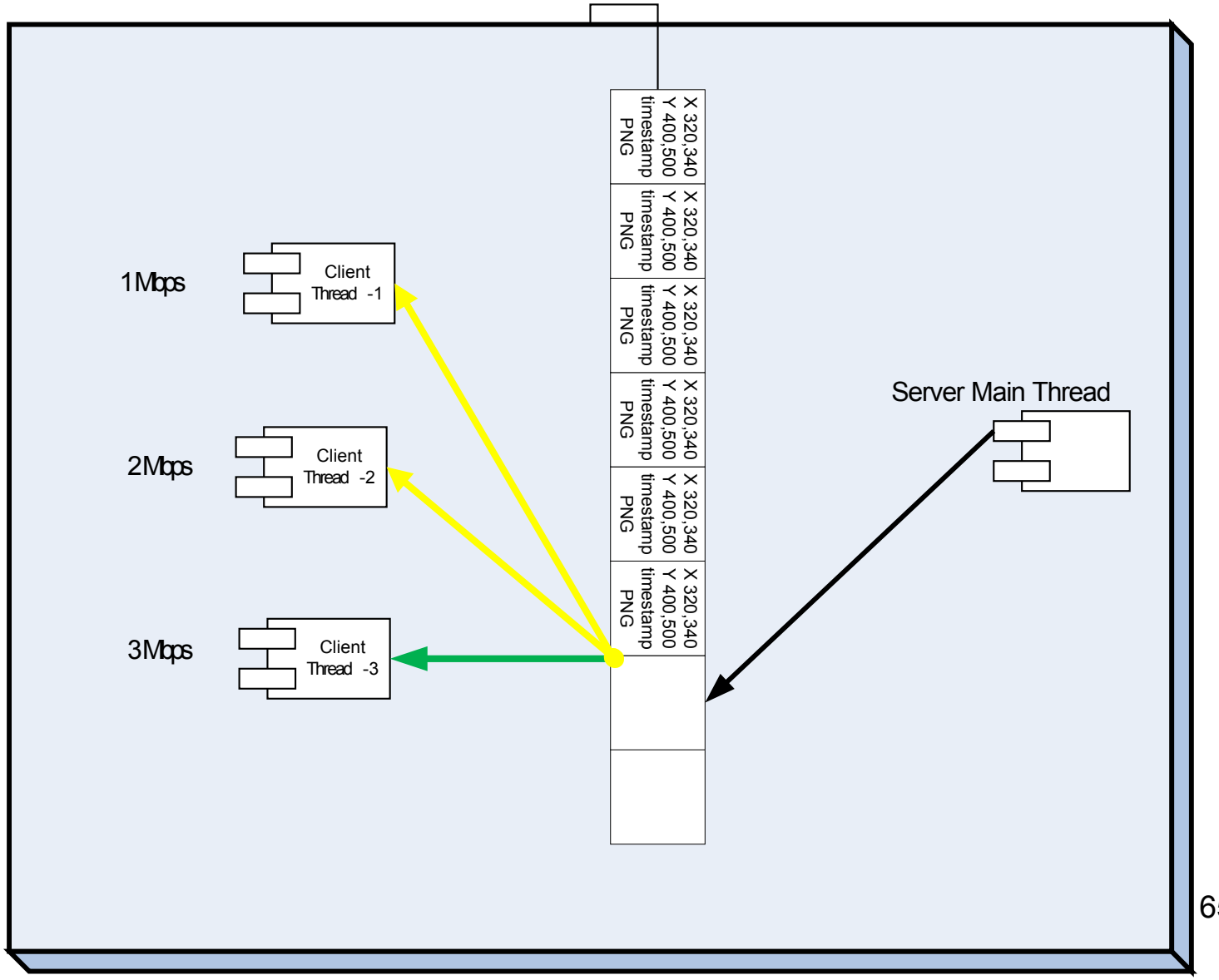
Different Client Bandwidths/Speeds

- Possible Solutions
  - Slowest one
    - Problem: Penalize everybody except the slowest
  - Average speed
    - Possible solution (Can we do better?)
  - Fastest one
    - The best solution
    - Client bandwidths are fully utilized

Different Client Bandwidths/Speeds



Different Client Bandwidths/Speeds



Challenges

- Different client bandwidths/speeds
- **Late Joiner**
- The effects of packet loss
- Reliable multicast

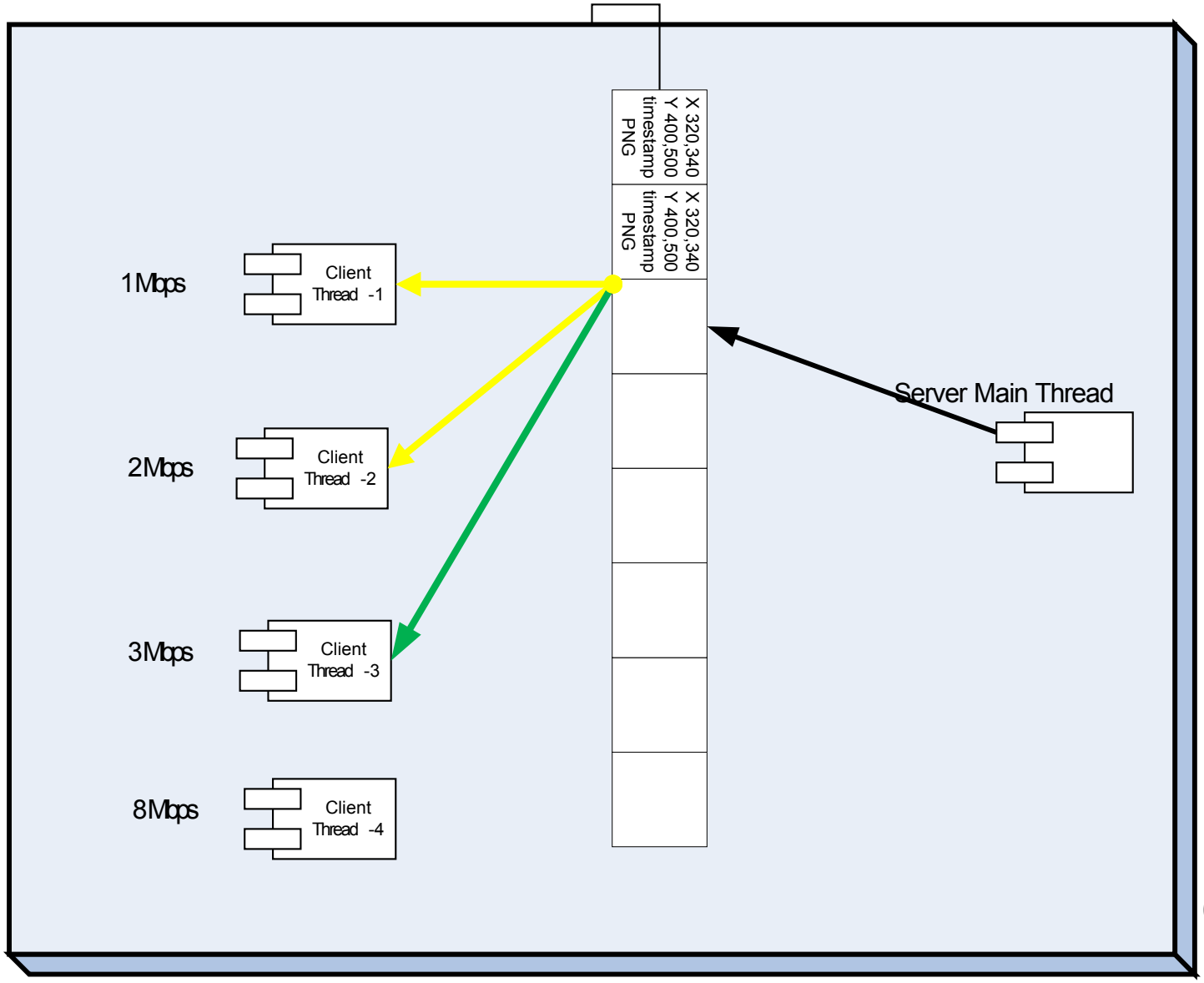
Late Joiner

- Force server to generate full screen update

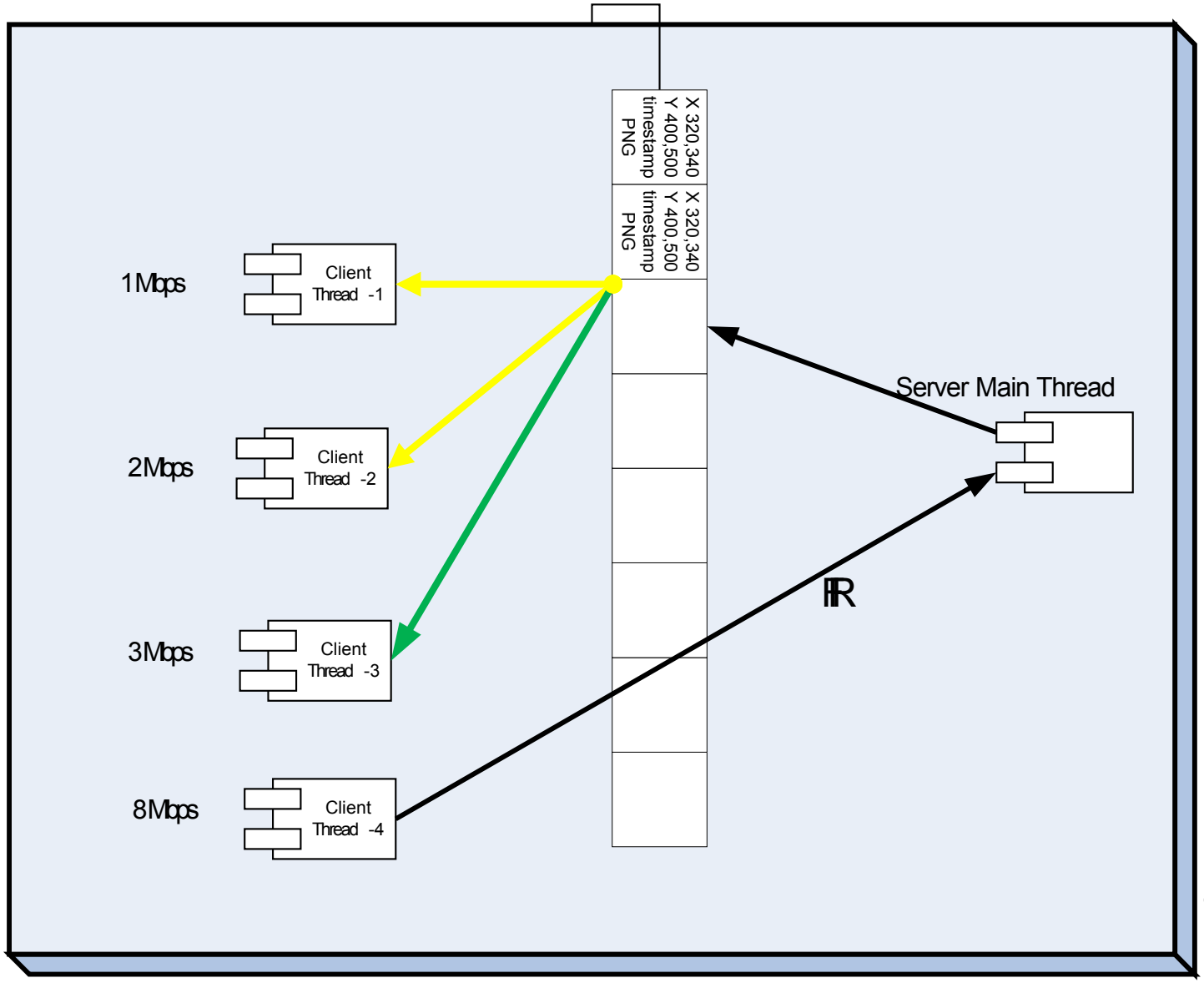
Late Joiner

- Force server to generate full screen update
  - Problems
    - Misbehaving clients can degrade performance
    - If Join/Leave rate is high, too much burden on server
  - Solution
    - Generate full screen updates if really necessary
    - Otherwise start the new client from last full screen update

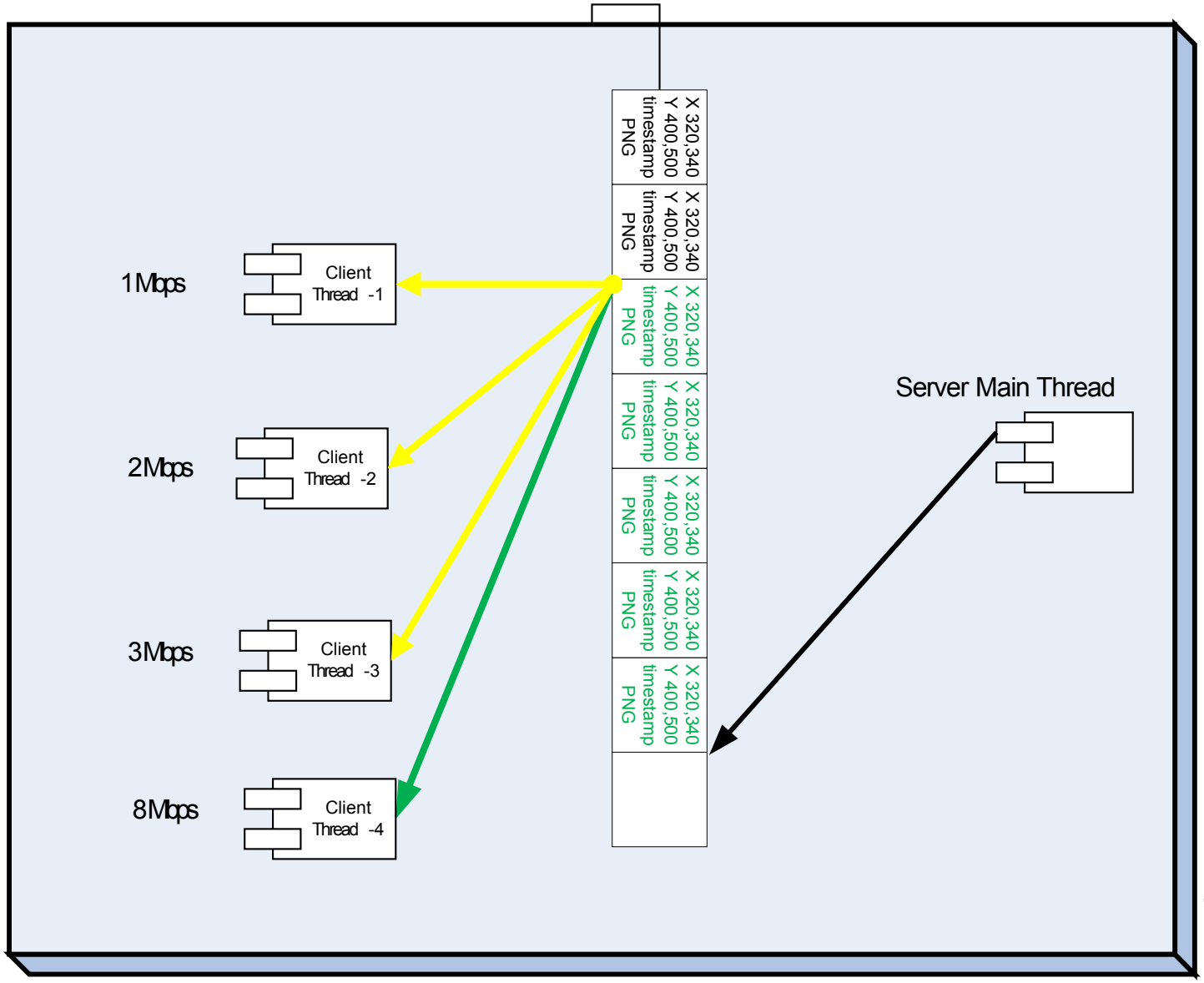
Different Client Bandwidths/Speeds



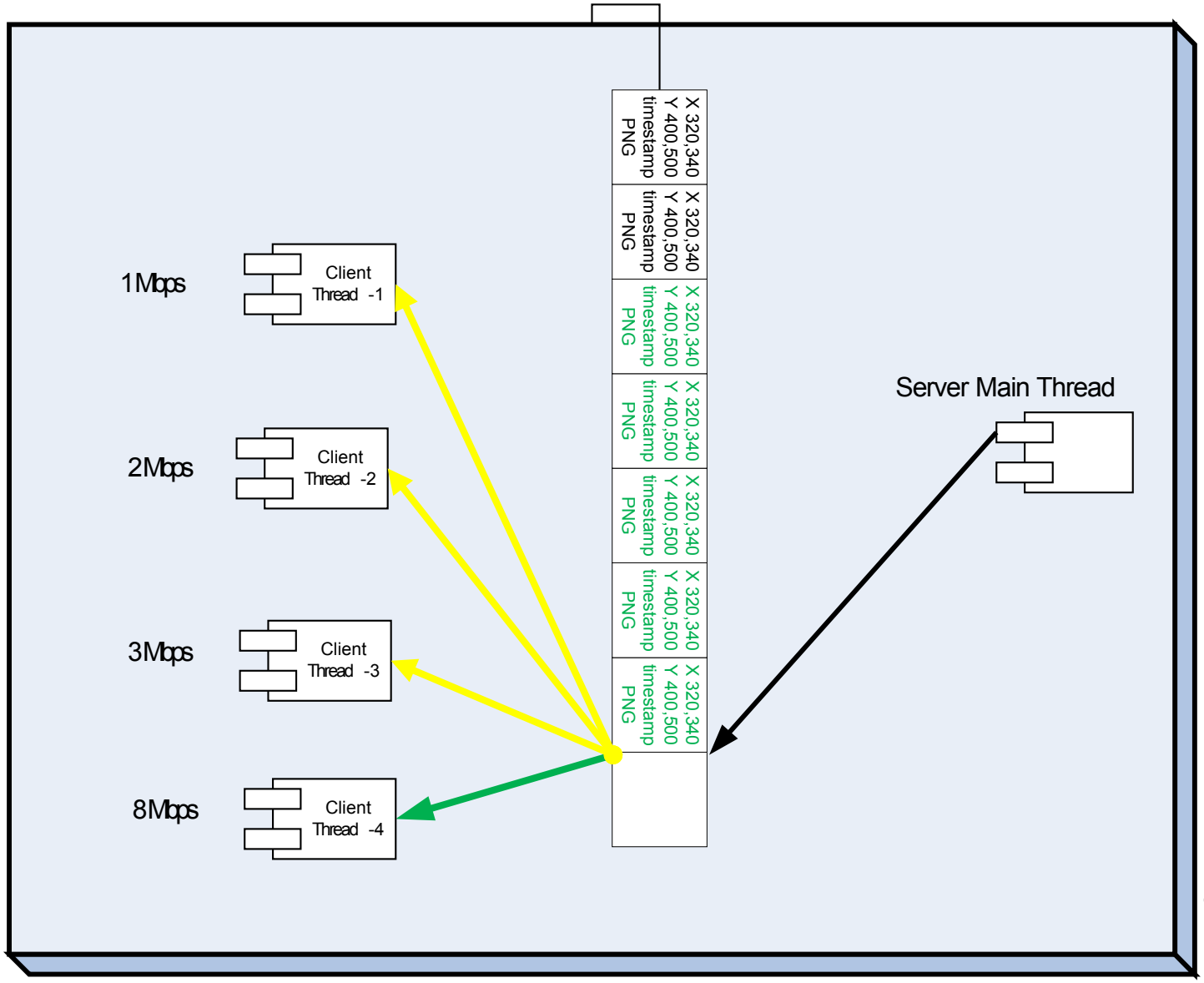
Different Client Bandwidths/Speeds



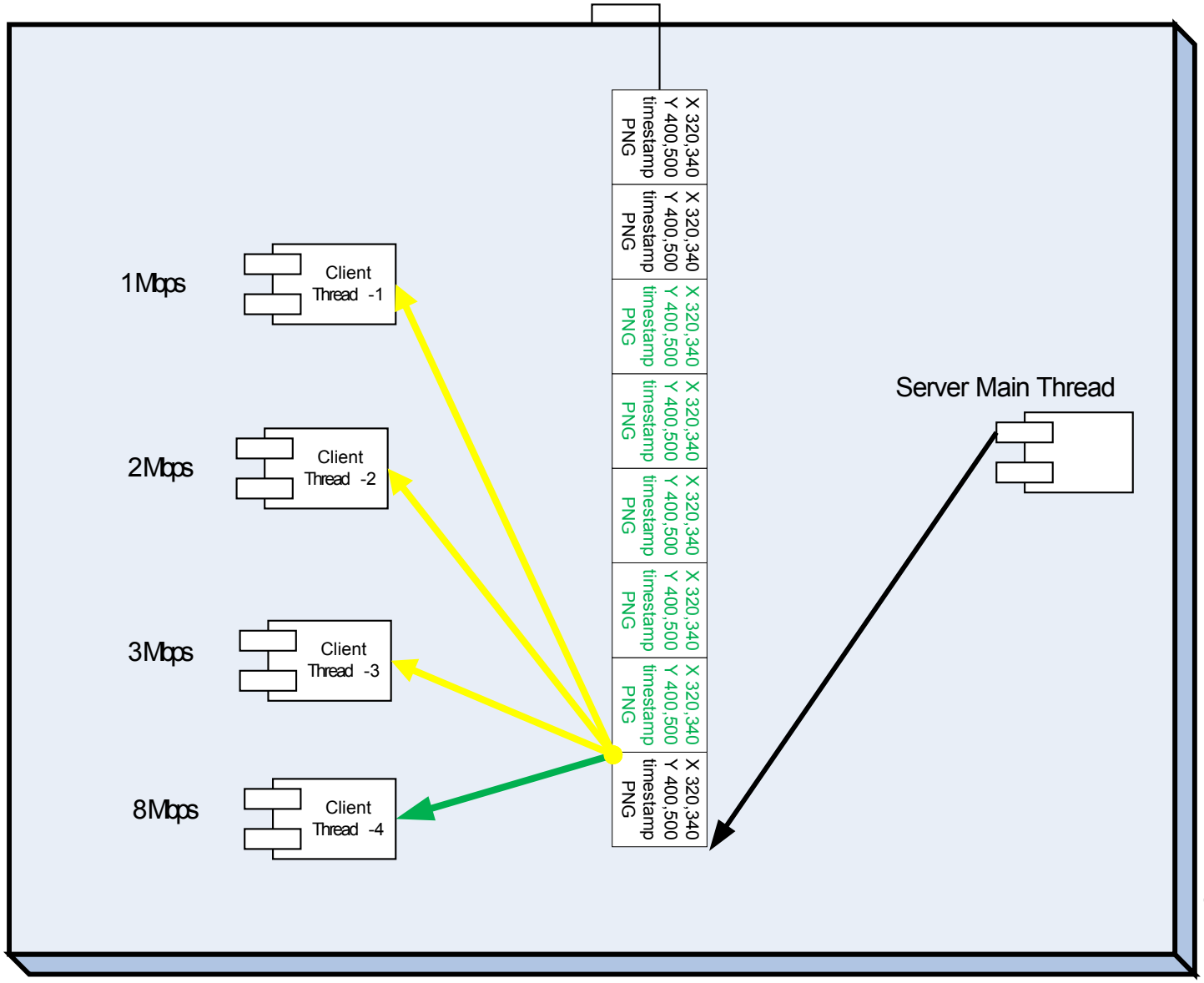
Different Client Bandwidths/Speeds



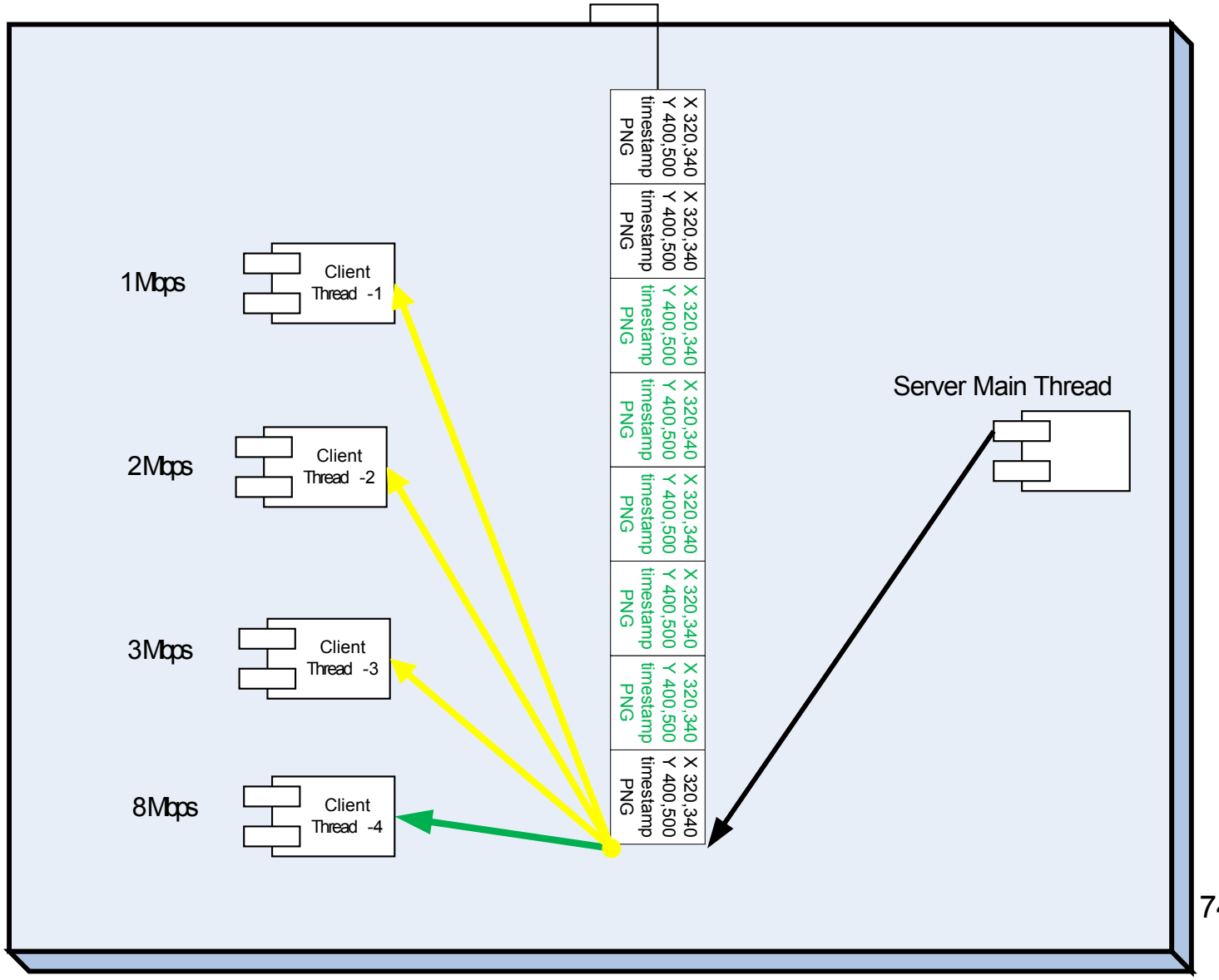
Different Client Bandwidths/Speeds



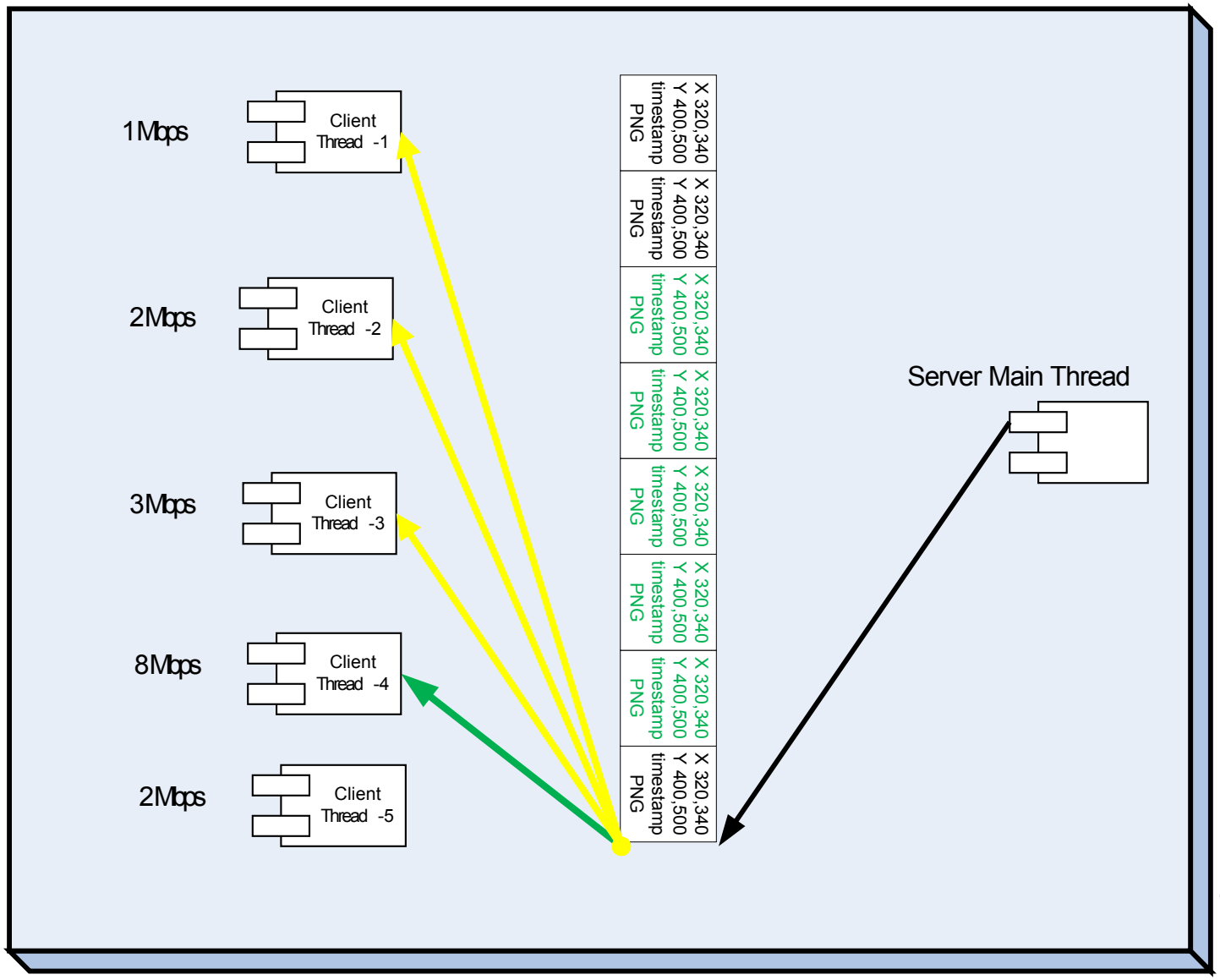
Different Client Bandwidths/Speeds



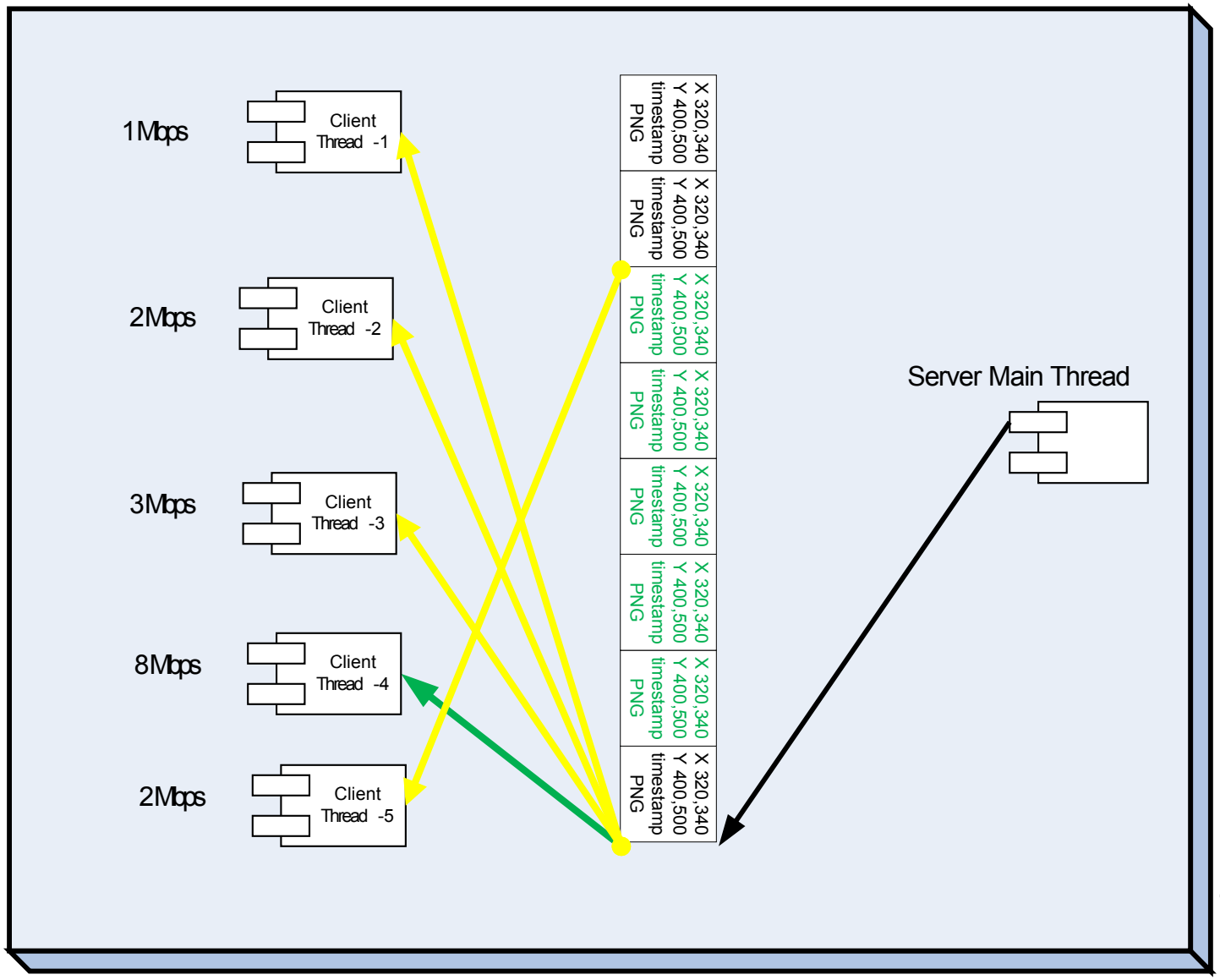
Different Client Bandwidths/Speeds



Different Client Bandwidths/Speeds



Different Client Bandwidths/Speeds



Challenges

- Different client bandwidths/speeds
- Late Joiner
- **The effects of packet loss**
- Reliable multicast

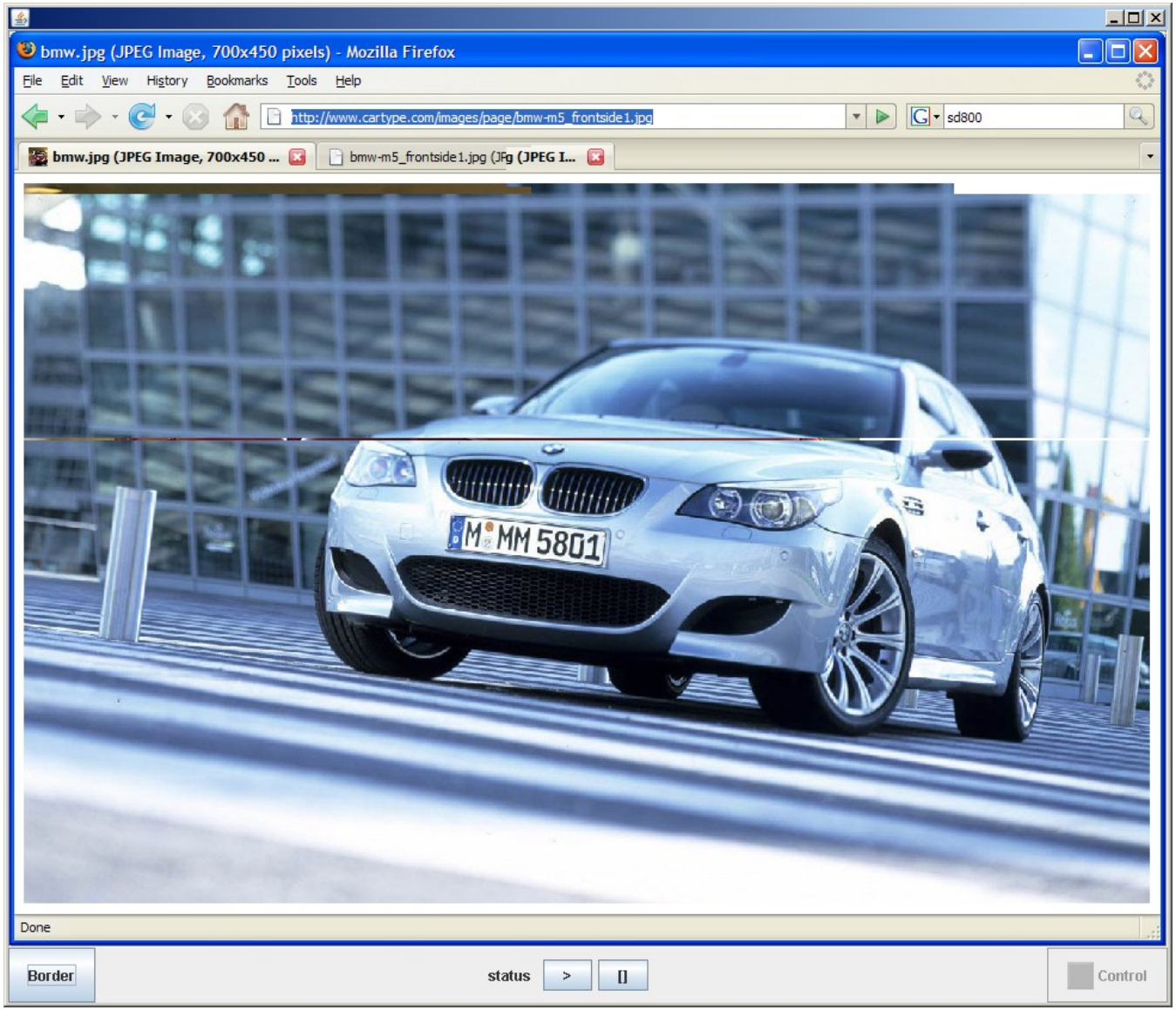
The effects of Packet Loss

- This problem applies to
  - Multicast
  - UDP
- The PNG images can be large
  - Regular desktop can be ~900KB
  - ~600 Ethernet packets
  - One packet loss wastes all PNG image

The effects of Packet Loss

- Solution
  - Small PNG images
    - Around ~1500 bytes
    - Consist of a few scanlines
  - Disadvantages
    - Increased CPU usage (client&server)
    - Lower compression ratio (%20 lower)
  - Advantages
    - One packet loss = no update for a few scanlines

The effects of Packet Loss



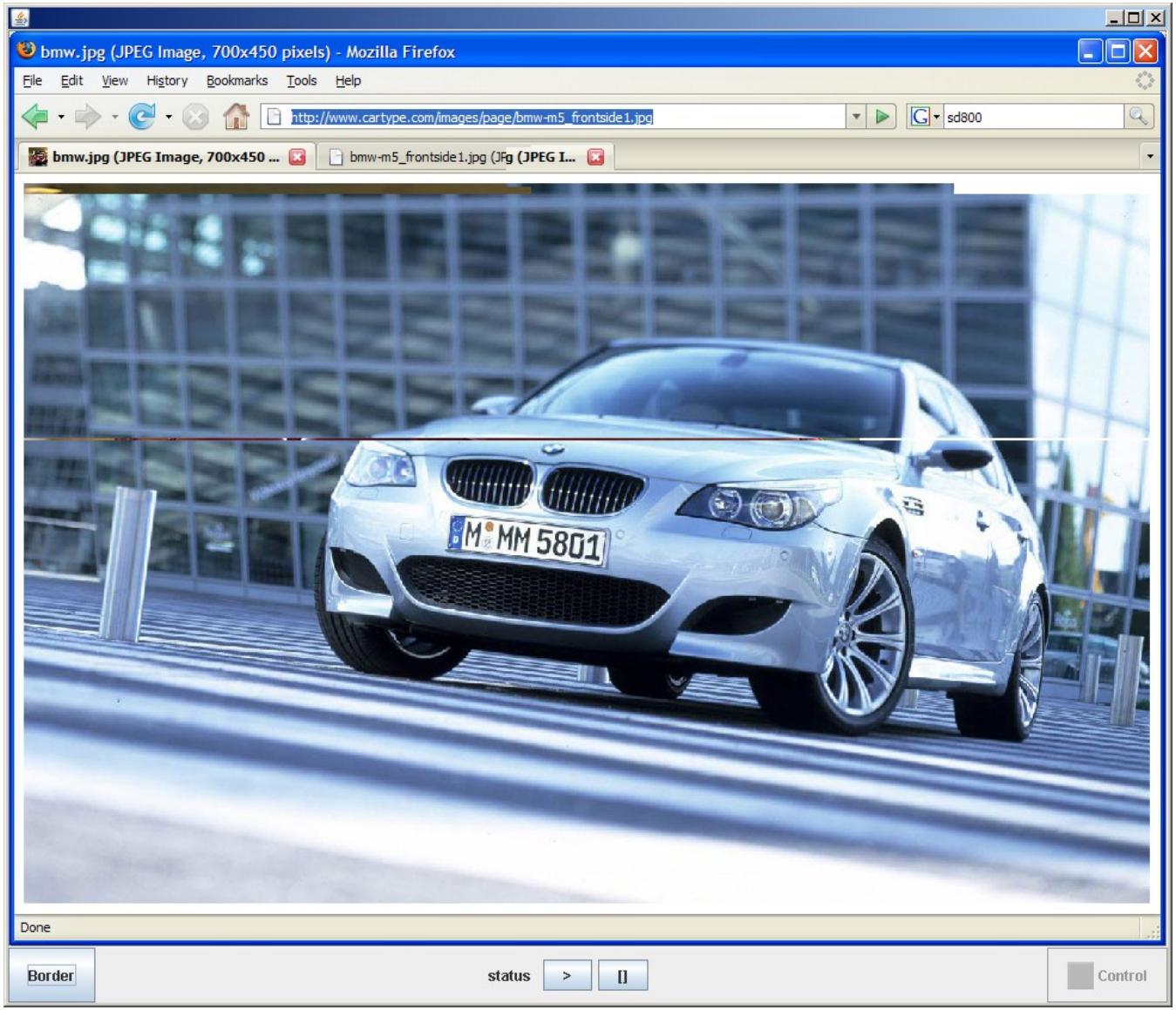
Challenges

- Different client bandwidths/speeds
- Late Joiner
- The effects of packet loss
- **Reliable multicast**

Reliable Multicast

- RTP Library stores last N rtp packets
- Clients send NACK for lost packets
- RTP Library resend the requested packets

The effects of Packet Loss



Overview

- Introduction
- Demo
- Architecture
- Challenges
- **Features**
- Conclusion

Recording

- Clients can record the whole/part session
- Anybody can play these files locally
- These files can be streamed to receivers via streaming server
- Streaming server supports multiple receivers
  - Also late joiners

Listening Client

- Client waits for incoming connections
- It can display windows from multiple user
- Can be used for RGB cable replacement

Conclusion

- Application sharing allows users to share a single application with multiple participants.
- Participants don't need the application.
- It is not specific to a single application.
- Extra features like recording is added.