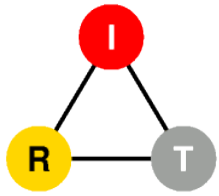


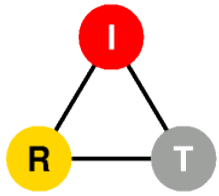
Application Sharing

Omer Boyaci
September 11, 2007



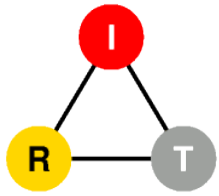
Overview

- Introduction
- Architecture
- Challenges
- Features
- Conclusion



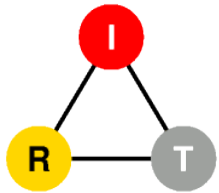
Overview

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



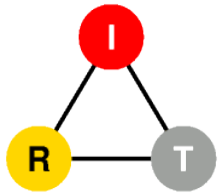
Application Sharing Models

- Application specific
 - + Efficient
 - Participants need application
 - Application has to be modified
- Generic
 - Inefficient (sometimes)
 - + Participants don't need application
 - + All applications are supported



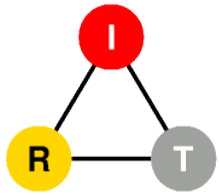
Use Cases

- Multiple Party working on the same document
- Classroom
 - Show something to remote or local users
 - Request them to repeat the same thing
 - All participants can watch and participate



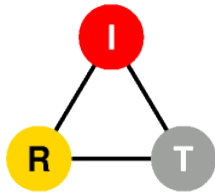
Application Sharing Models

- Application specific
 - + Efficient
 - Participants need application
 - Application has to be modified
- **Generic**
 - **Inefficient (sometimes)**
 - + **Participants don't need application**
 - + **All applications are supported**

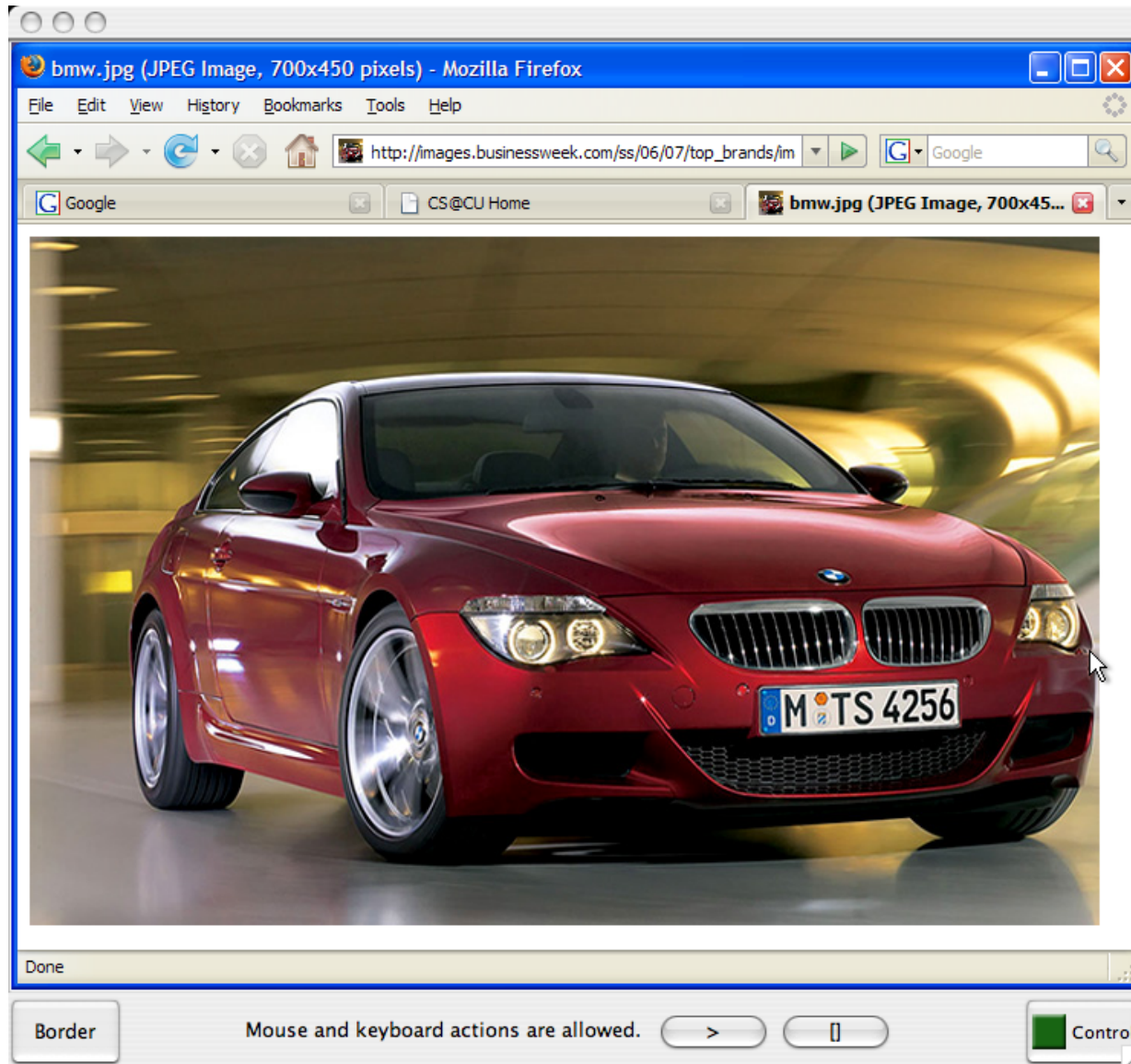


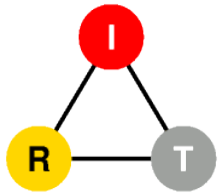
Application Sharing

- ♦ 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
- ♦ Desktop sharing is also supported.

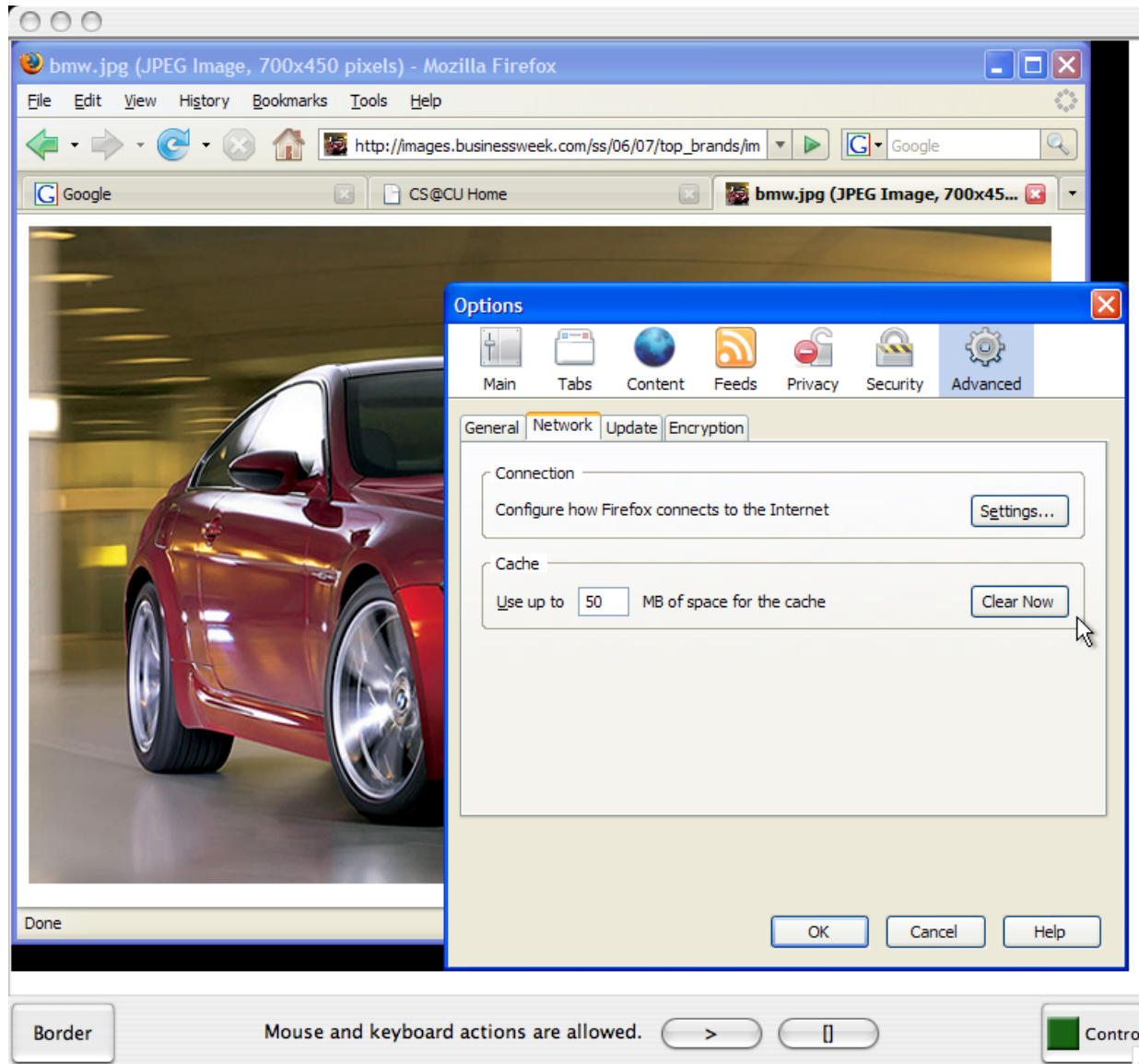


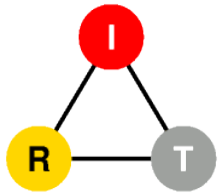
Screenshot








Screenshot (2)

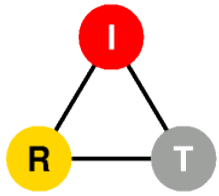




Supported Platforms/OS

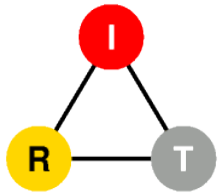
	Server	Client*
Windows 	+	+
*nix 	- +	+
Mac OS X 	-	+

*Client is Java based.



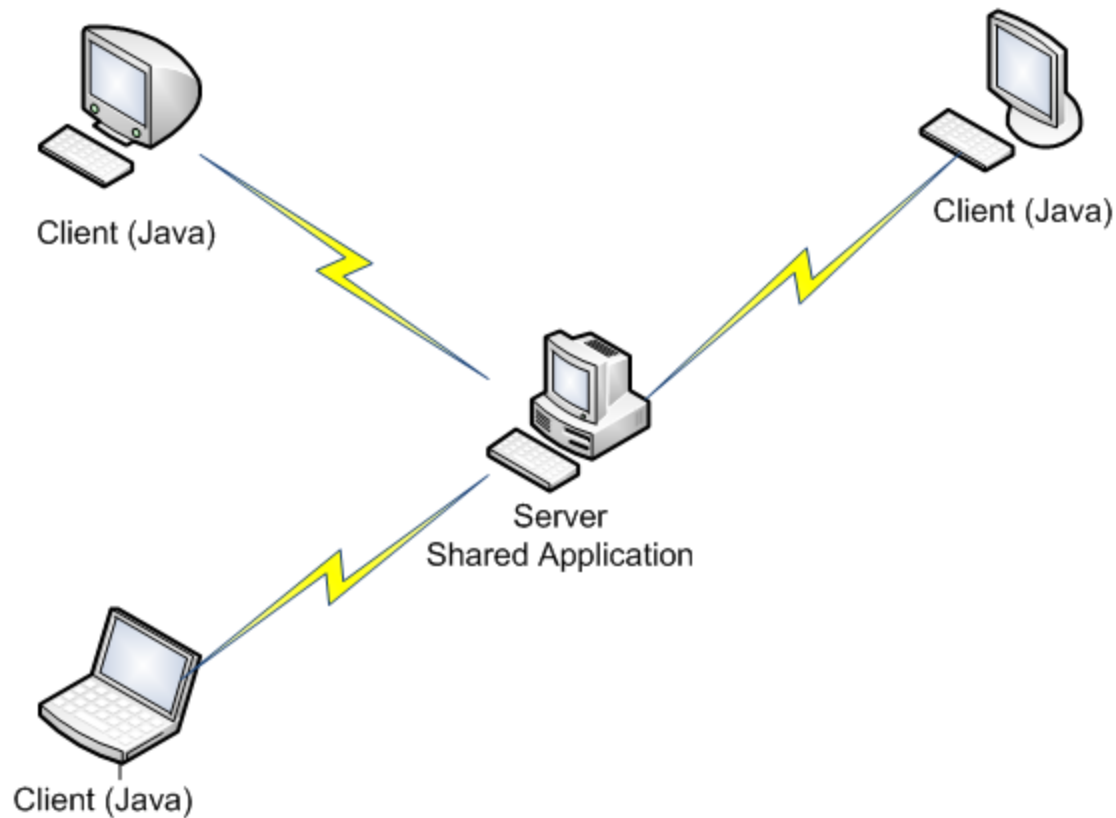
Overview

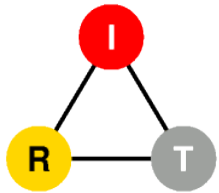
- Introduction
- **Architecture**
- Challenges
- Features
- Conclusion



System Architecture

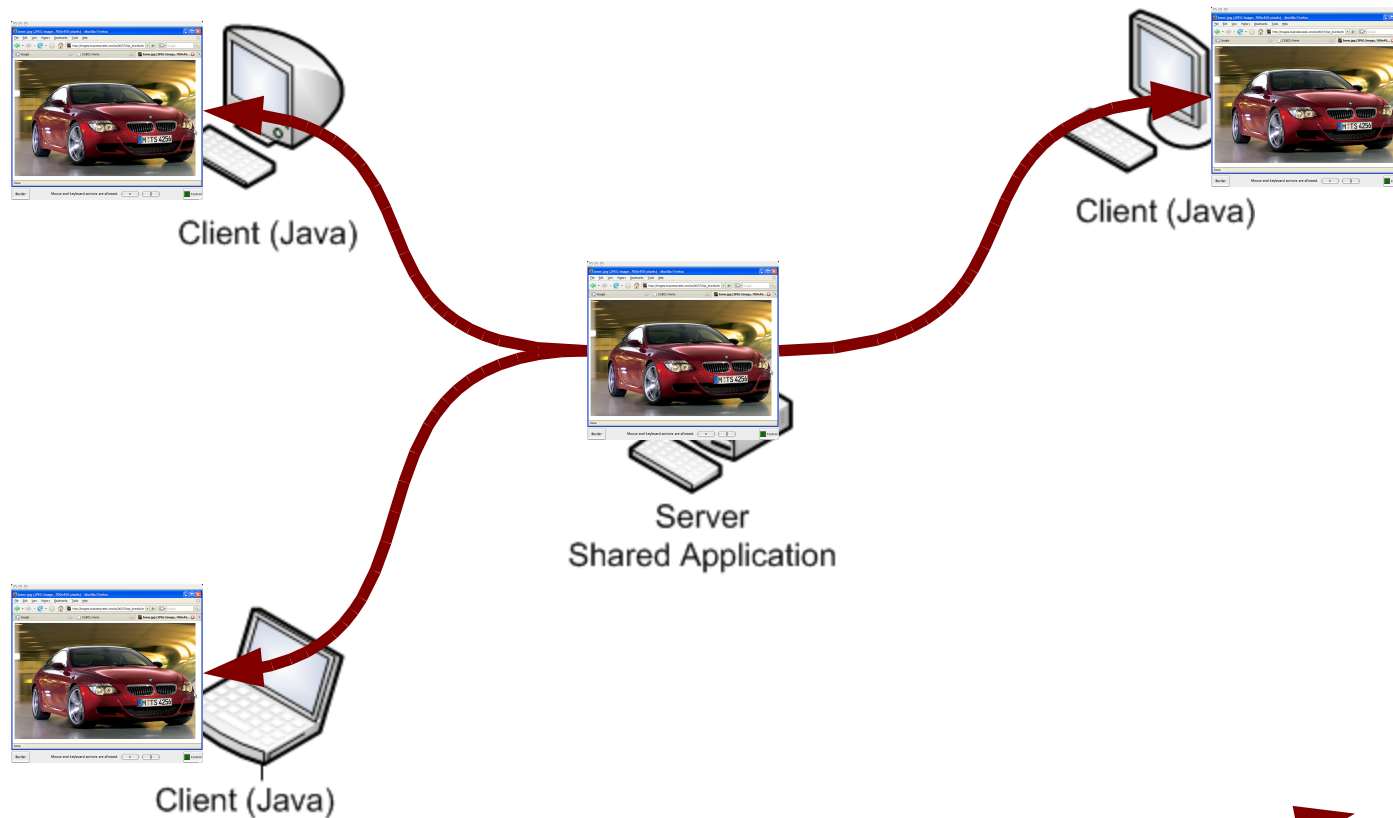
- Client/Server Software Architecture



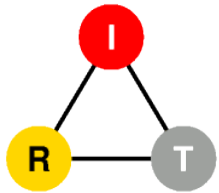


System Architecture

- Client/Server Software Architecture

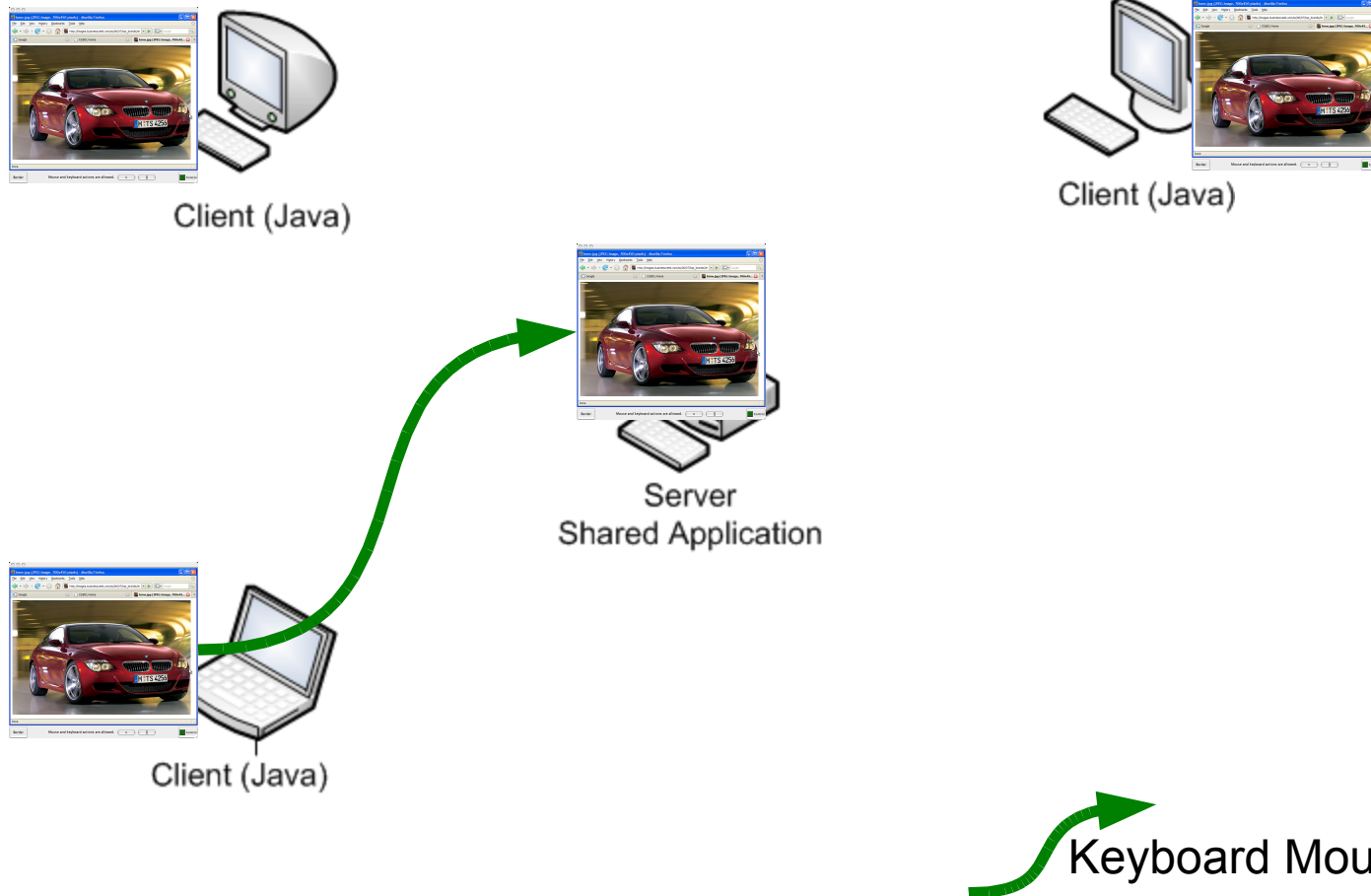


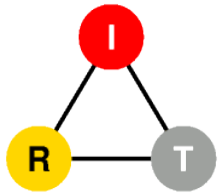
 Screen Updates



System Architecture

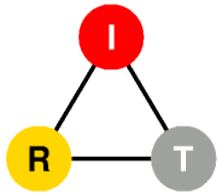
- Client/Server Software Architecture





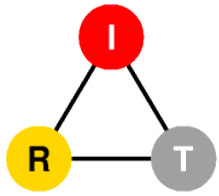
Client (Viewer) Architecture

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



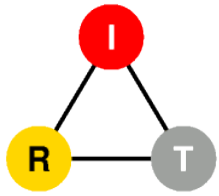
Client (Viewer) Architecture

- Client receives these commands
 - Open new window
 - Window size changed
 - Pixel update
 - Close window
- Client sends
 - BFCP (Binary Floor Control Protocol) commands
 - Keyboard and mouse events



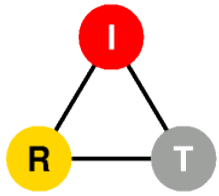
Overview

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



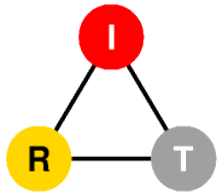
Challenges

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



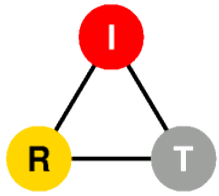
Overview

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



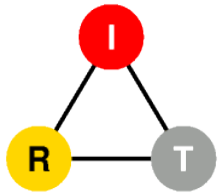
Recording

- 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



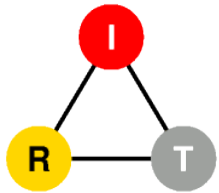
Listening Client

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



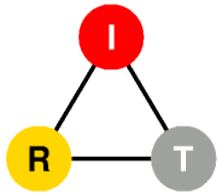
Overview

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



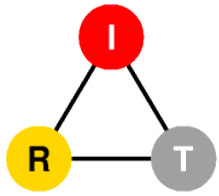
Conclusion

- 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, ...



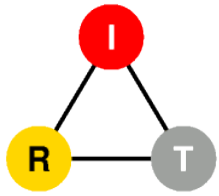
Thanks

- Questions?



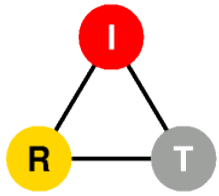
Overview

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



Challenges

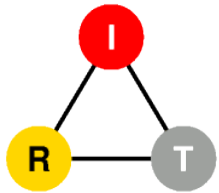
- Different client bandwidths/speeds
- Late Joiner
- The effects of packet loss
- Reliable multicast



Challenges

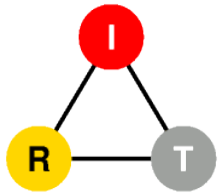
- Different client bandwidths/speeds
- Late Joiner
- The effects of packet loss
- Reliable multicast





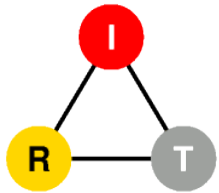
Different Client Bandwidths/Speeds

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



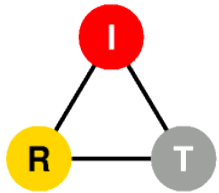
Different Client Bandwidths/Speeds

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



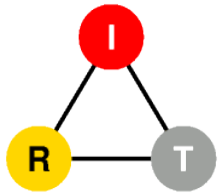
Different Client Bandwidths/Speeds

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



Different Client Bandwidths/Speeds

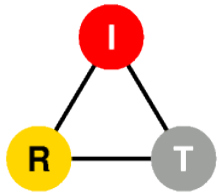
- Possible Solutions
 - Slowest one
 - Problem: Penalize everybody except the slowest
 - Average speed
 - Possible solution (Can we do better?)
 - Fastest one



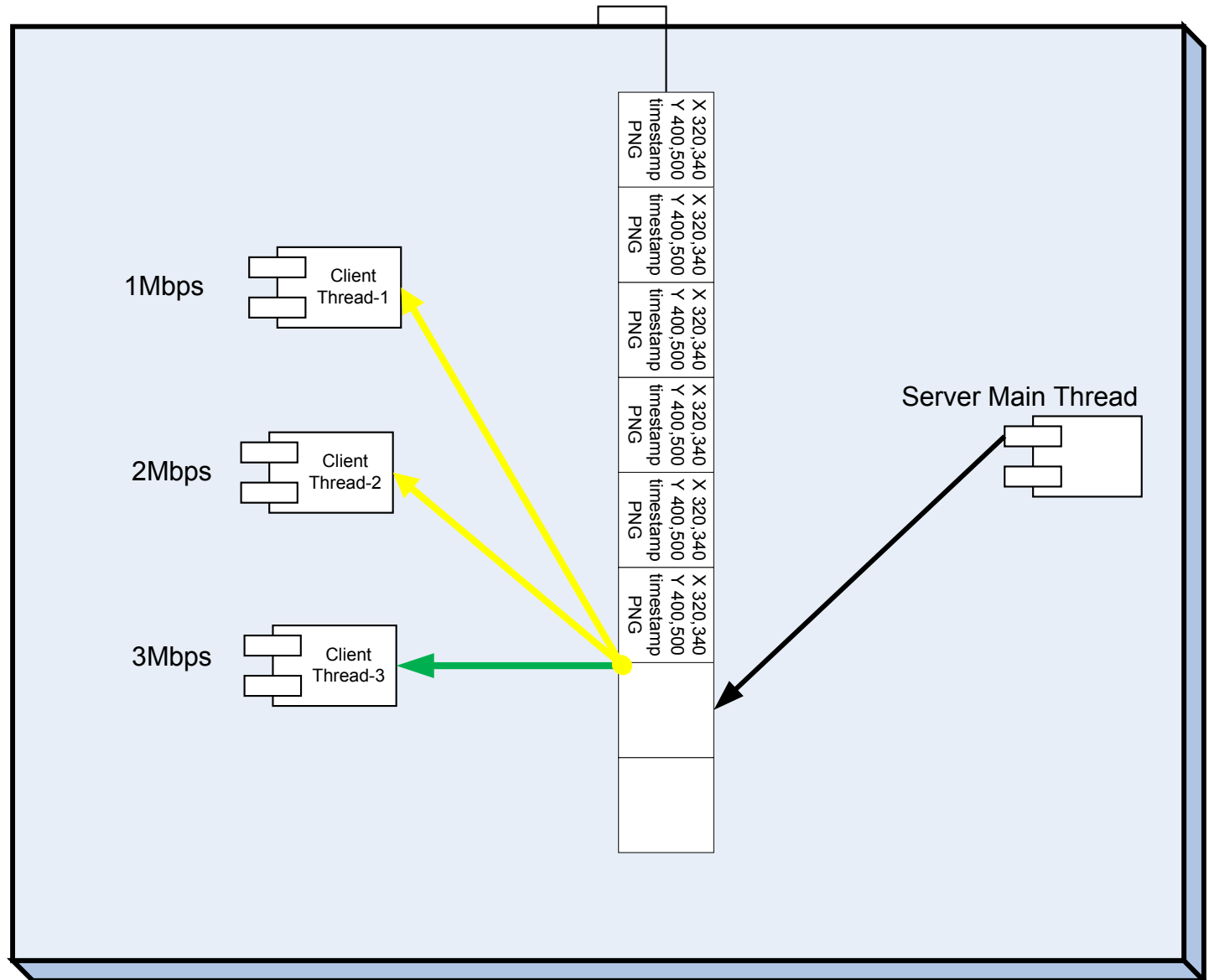
Different Client Bandwidths/Speeds

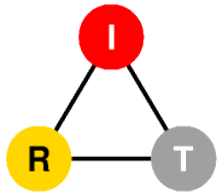
- Possible Solutions
 - Slowest one
 - Problem: Penalize everybody except the slowest
 - Average speed
 - Possible solution (Can we do better?)
 - Fastest one
 - The best solution
 - Client bandwidths are fully utilized





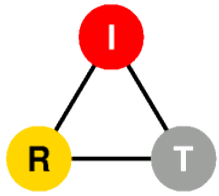
Different Client Bandwidths/Speeds





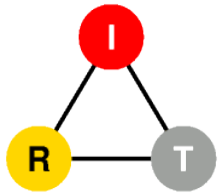
Challenges

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



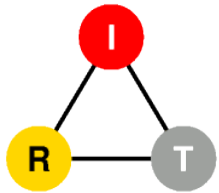
Late Joiner

- Force server to generate full screen update

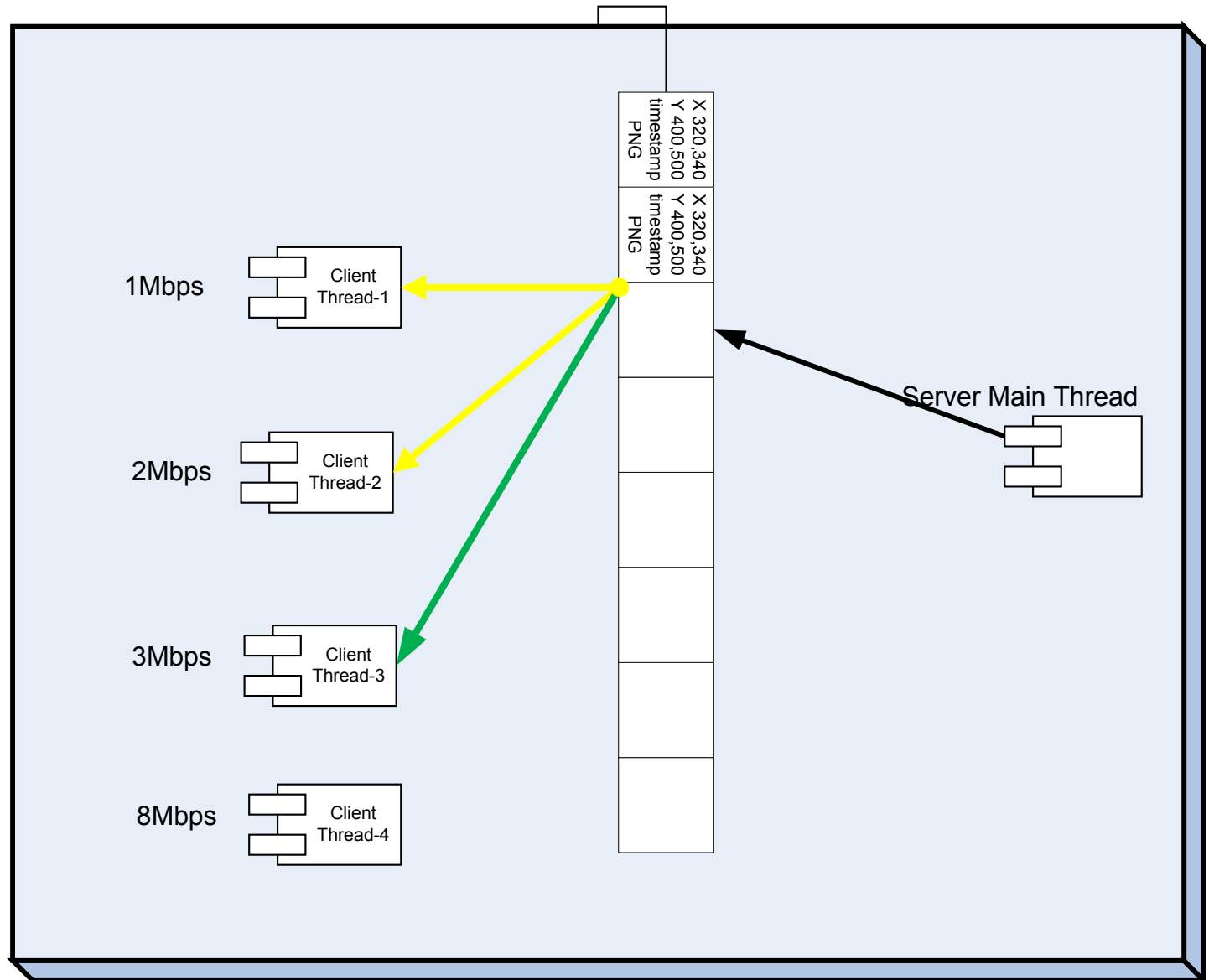


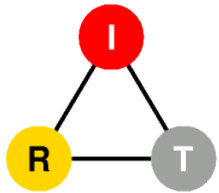
Late Joiner

- 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

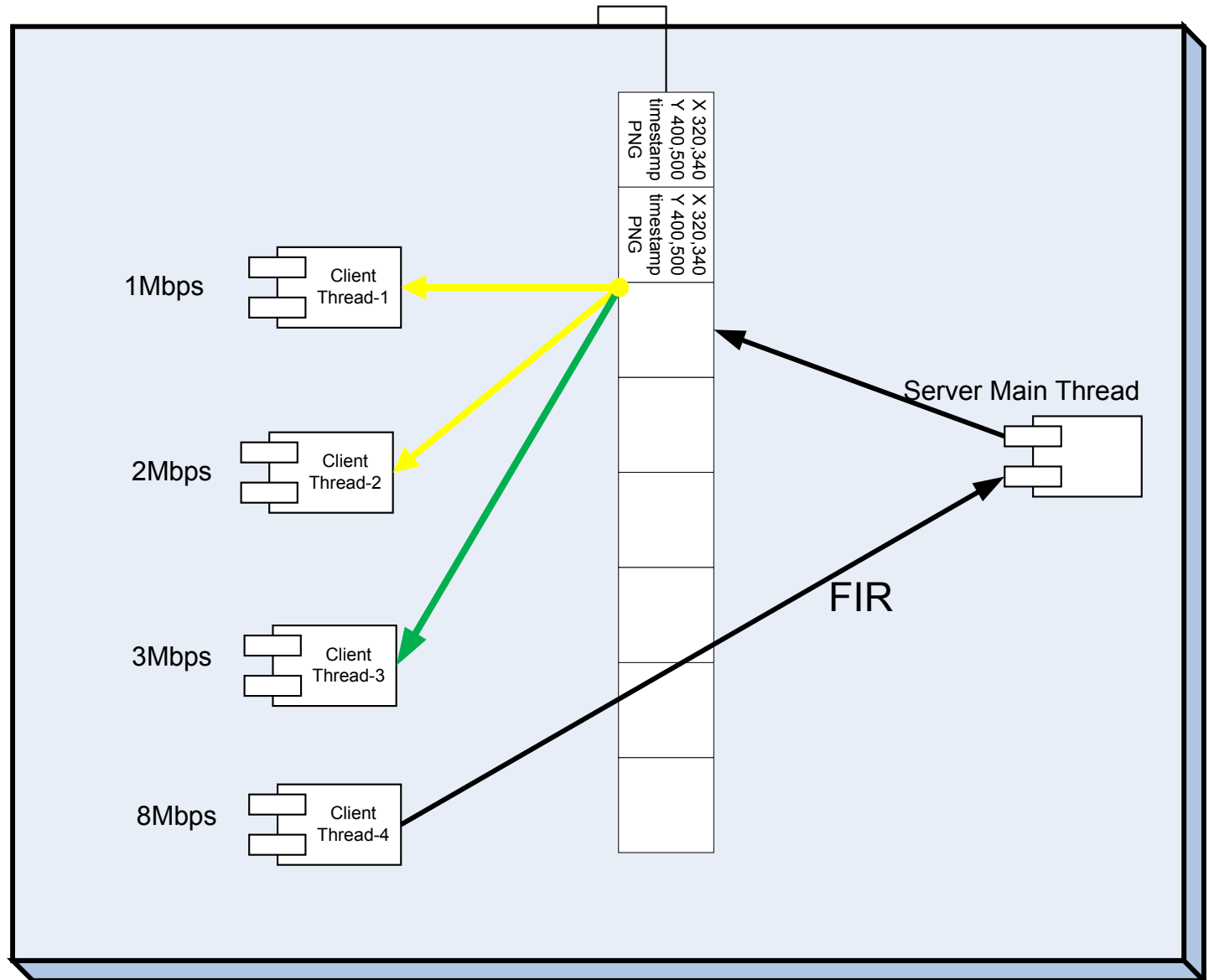


Different Client Bandwidths/Speeds

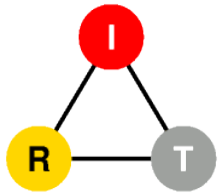




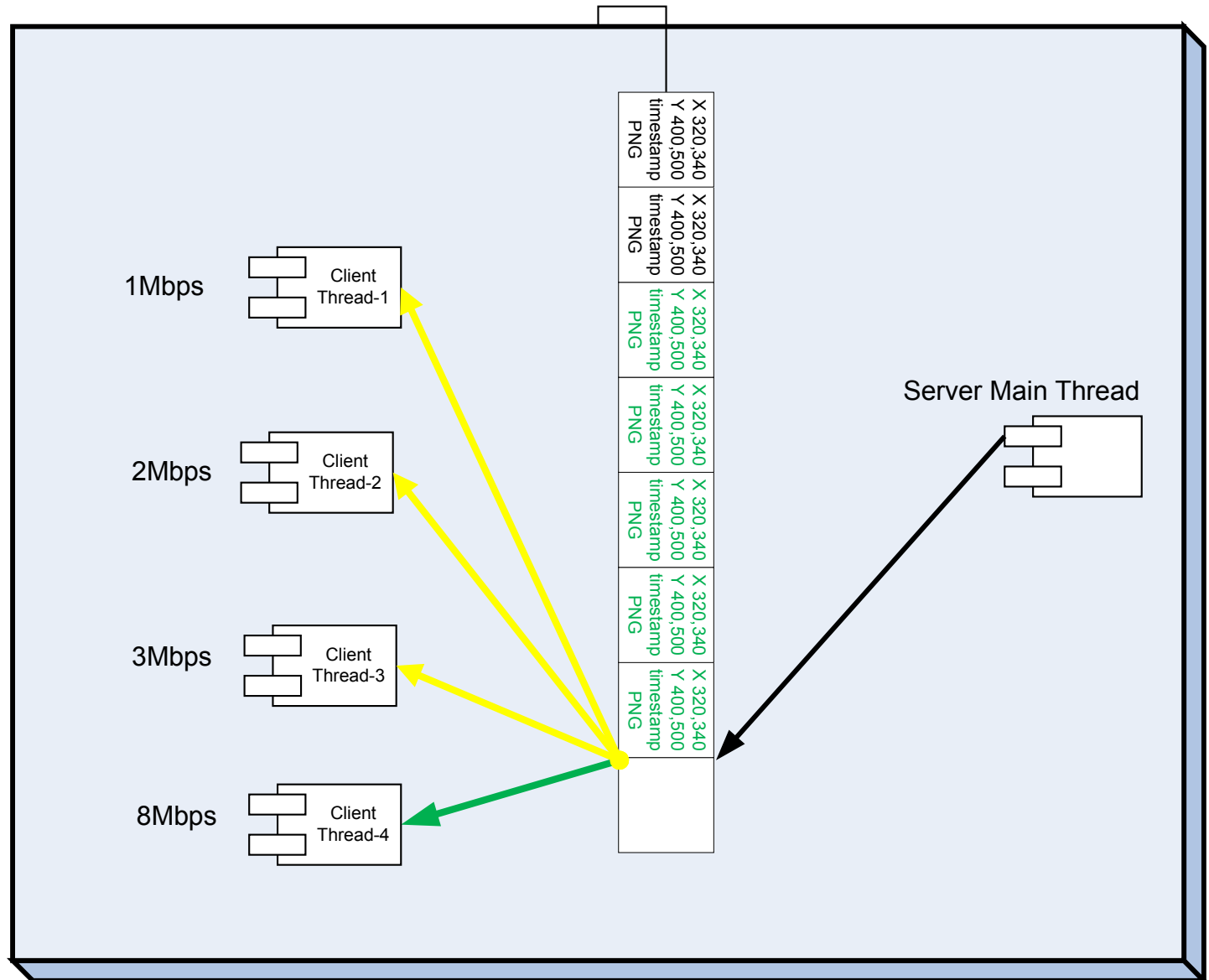
Different Client Bandwidths/Speeds

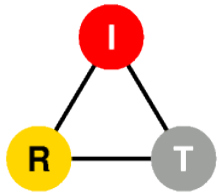




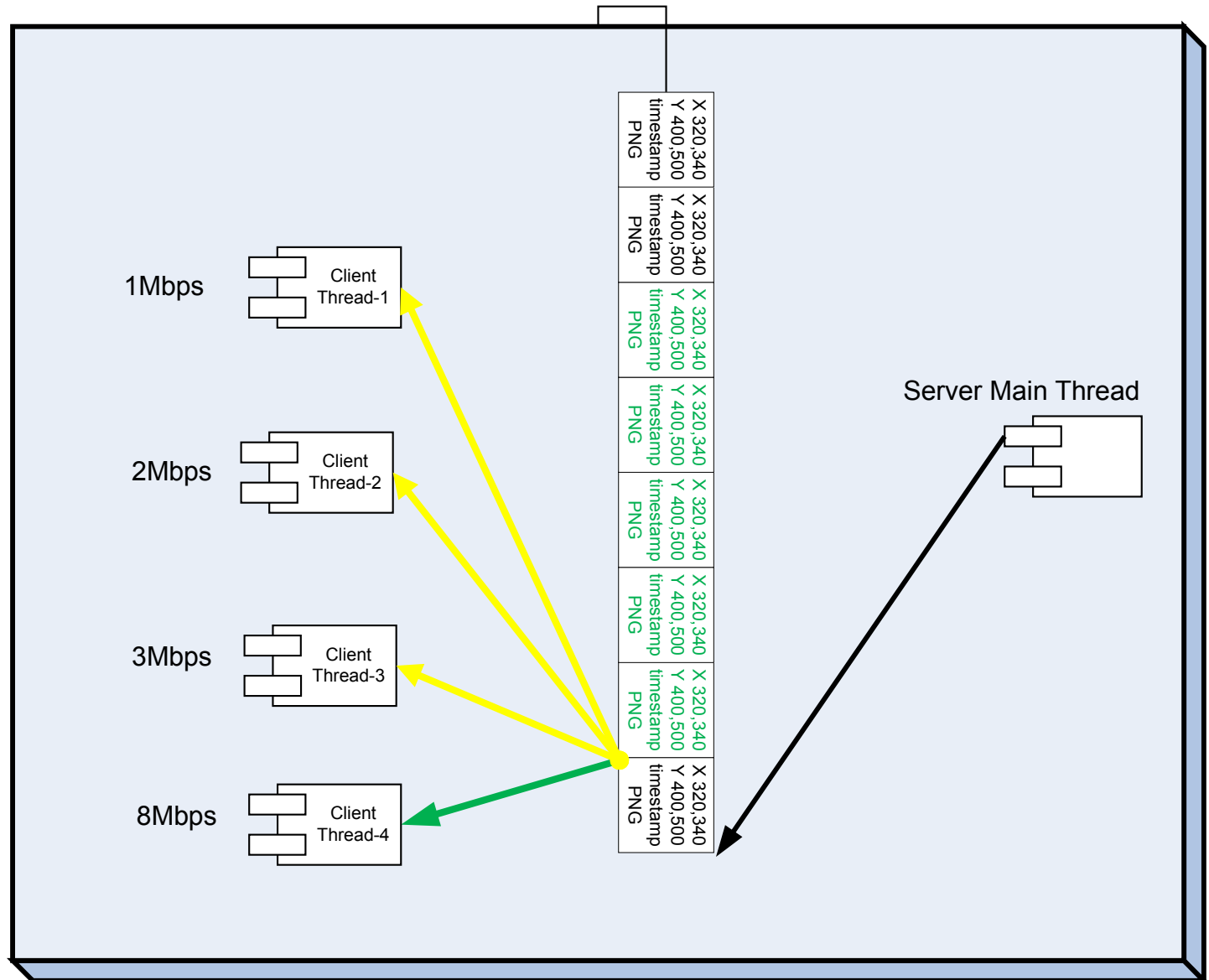


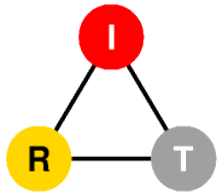
Different Client Bandwidths/Speeds



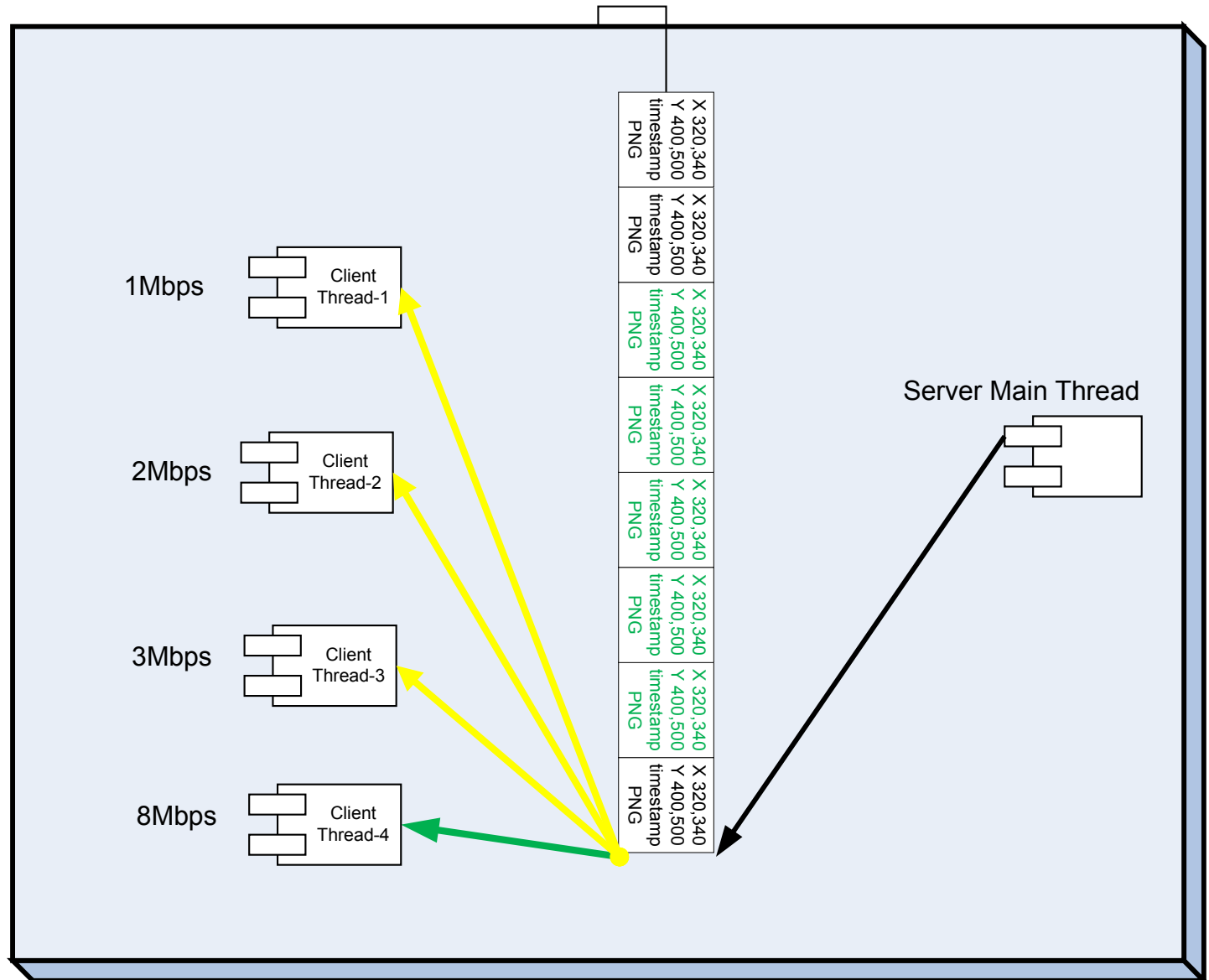


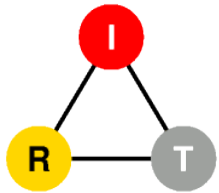
Different Client Bandwidths/Speeds



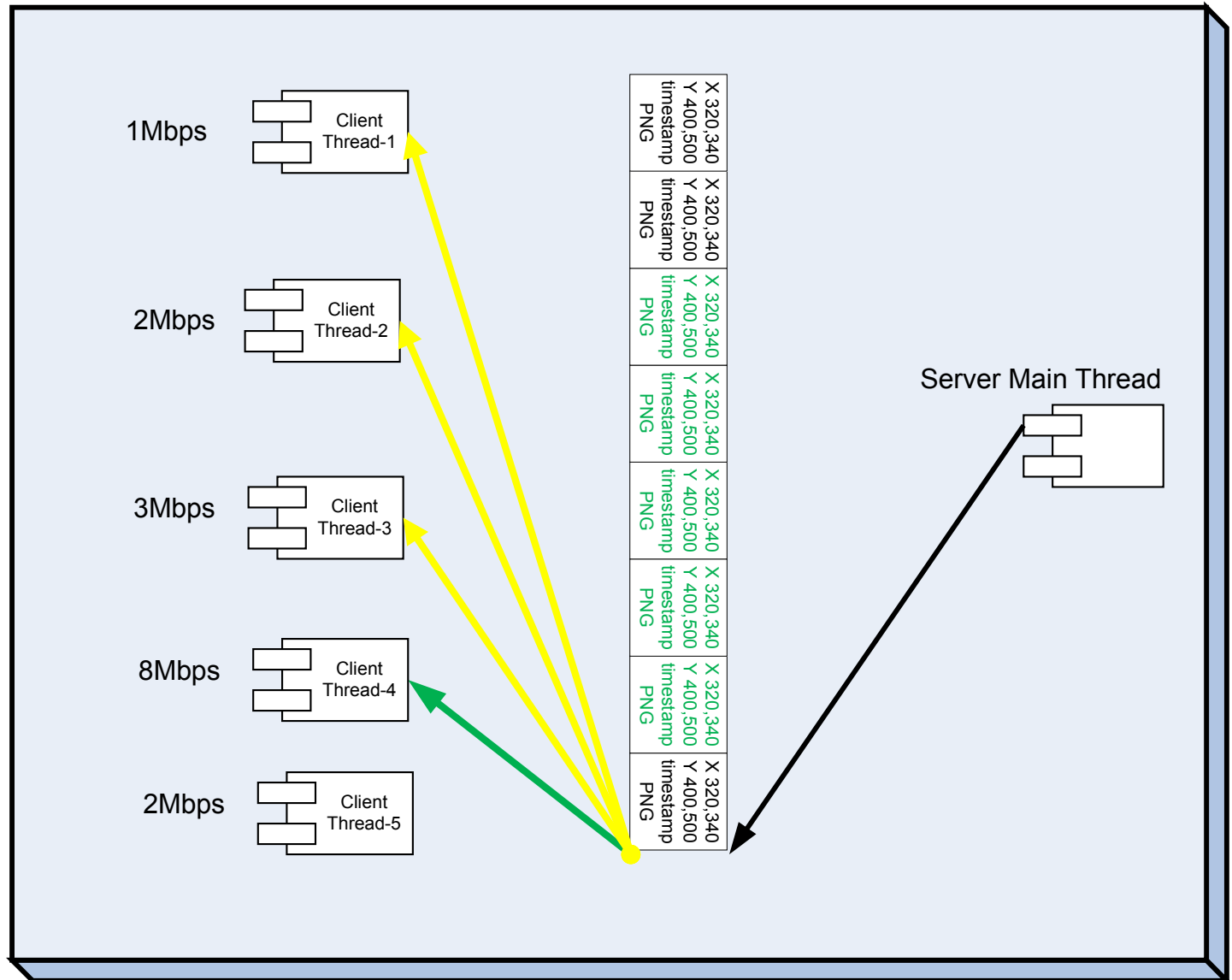


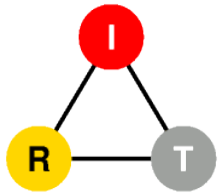
Different Client Bandwidths/Speeds



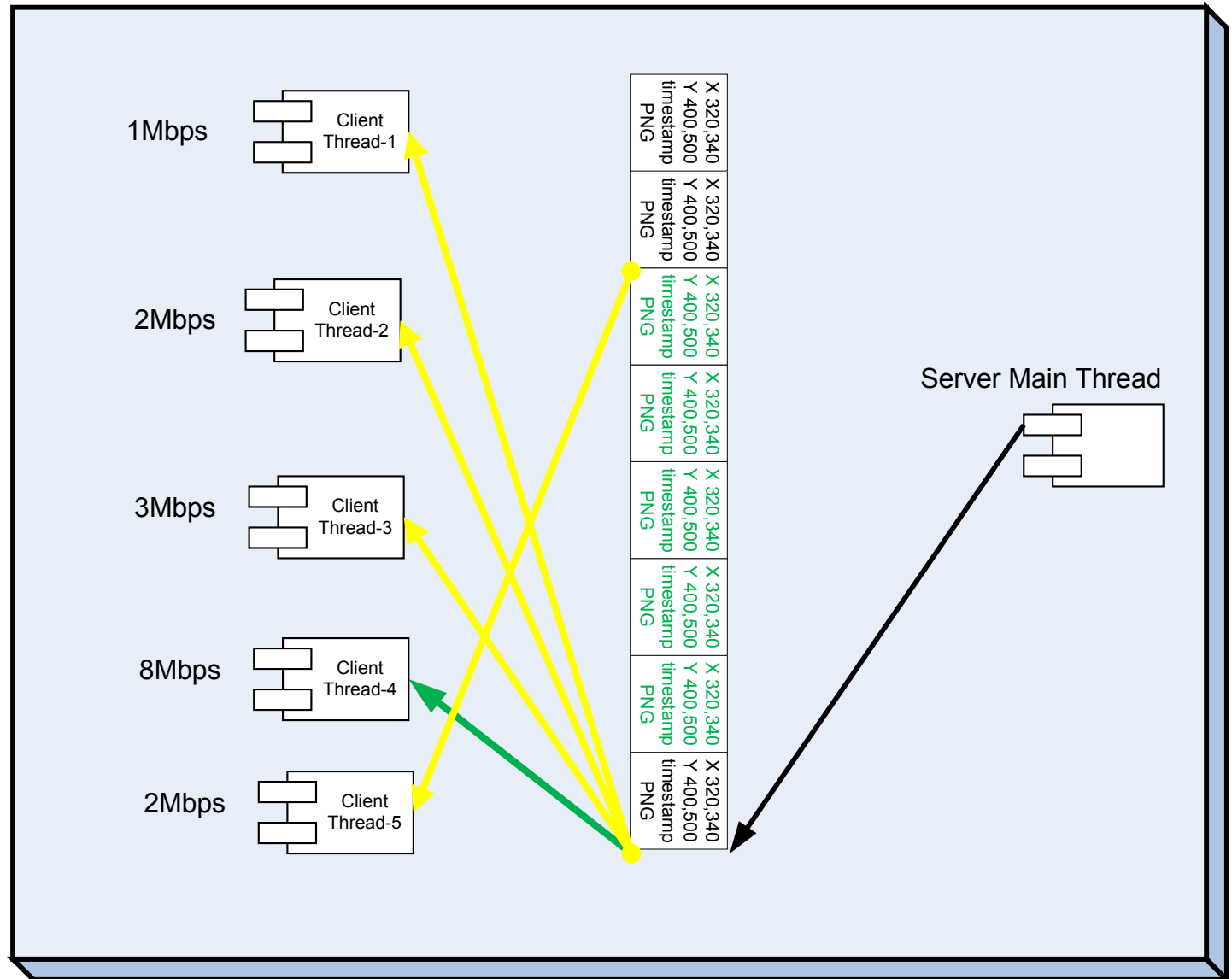


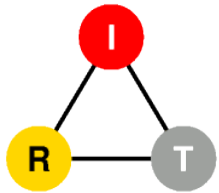
Different Client Bandwidths/Speeds





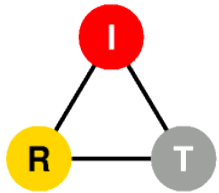
Different Client Bandwidths/Speeds





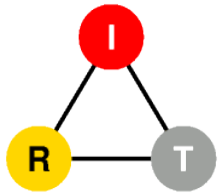
Challenges

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



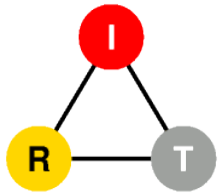
The effects of Packet Loss

- This problem applies to
 - Multicast
 - UDP
- The PNG images can be large
 - Regular desktop can be ~1MB
 - ~600 Ethernet packets
 - One packet loss wastes the whole PNG image

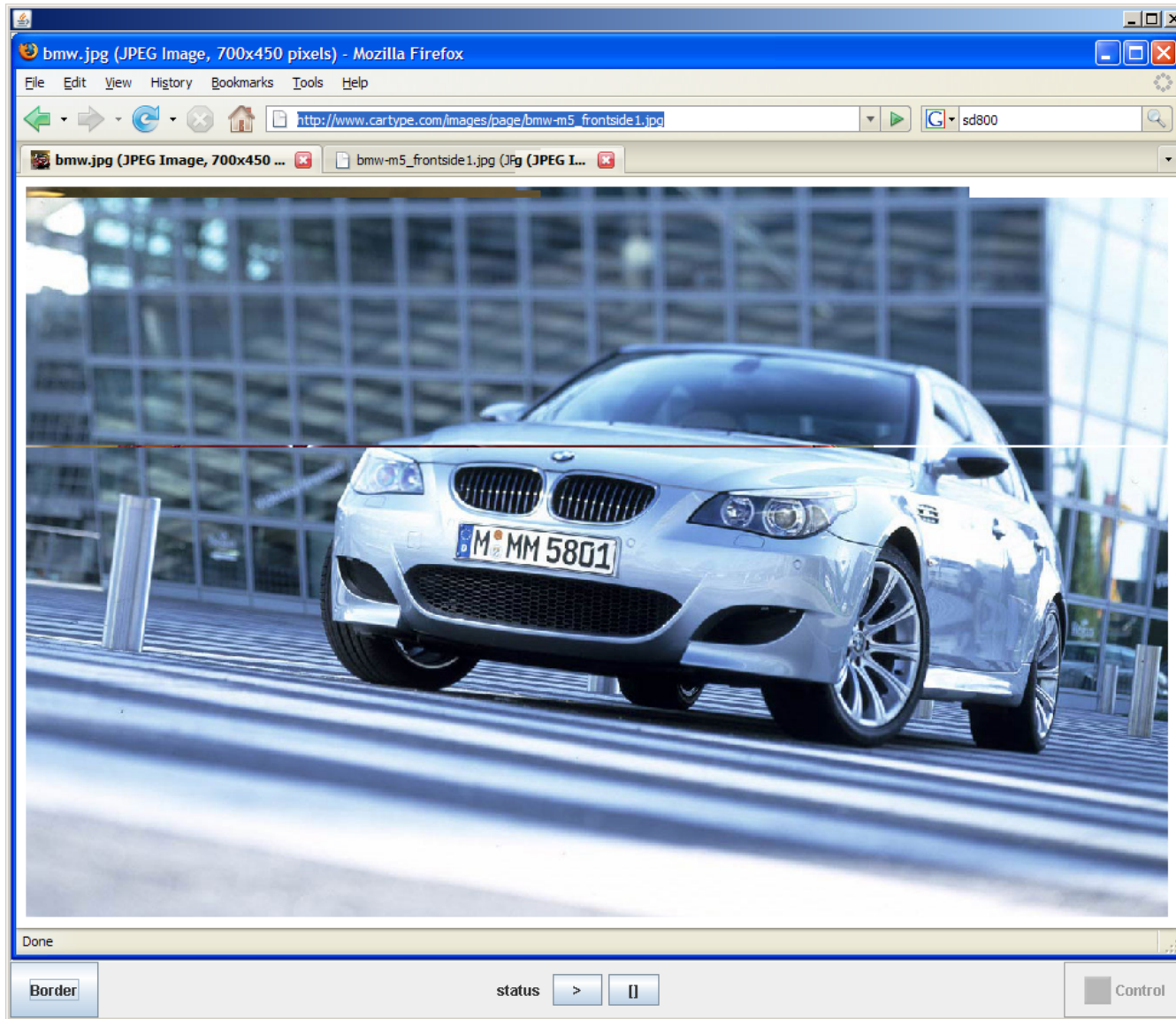


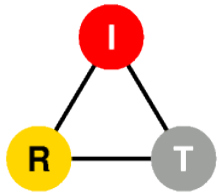
The effects of Packet Loss

- 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



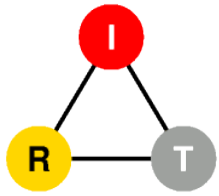
The effects of Packet Loss





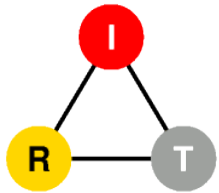
Challenges

- 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

