Introduction to Networked Graphics

IEEE Virtual Reality 2011

Anthony Steed

Part 1:
Introduction
Networking Fundamentals

Introduction to Networked Graphics

IEEE Virtual Reality 2011

Anthony Steed

Goals

- Understanding the application domain
- Understanding the constraints the theoretical and practical constraints of distributed simulation over the Internet
- Common strategies for mitigating constraints whilst satisfying
- Practical tools for building networked virtual environments and networked games
- Examples and literature that you can use to further your own knowledge

Session 1 (9:00 – 10:30)

- *Introduction*
- - What are networked graphics?
- - Potted history of networked graphics
- *Networking Fundamentals*
- - Basic Internet technologies
- - Basic networking strategies
- - Why are all types of networked graphics "non-standard" networking applications?

Session 2 (11:00 –12:30)

- *Actual Internet Performance*
- - What bandwidth can we expect?
- - Sources of latency
- - What other issues must we consider?
- *Requirements and Constraints*
- - Requirements on consistency
- - Requirements on latency
- - User response to inconsistency and latency

Session 3 (13.30 – 15:00)

- *Latency*
- - Mitigation strategies
- - Playout delays, local lag and dead reckoning
- *Scalability*
- - Management of awareness
- - Interest specification
- - Server partitioning

Session 4 (15:15 – 16:45)

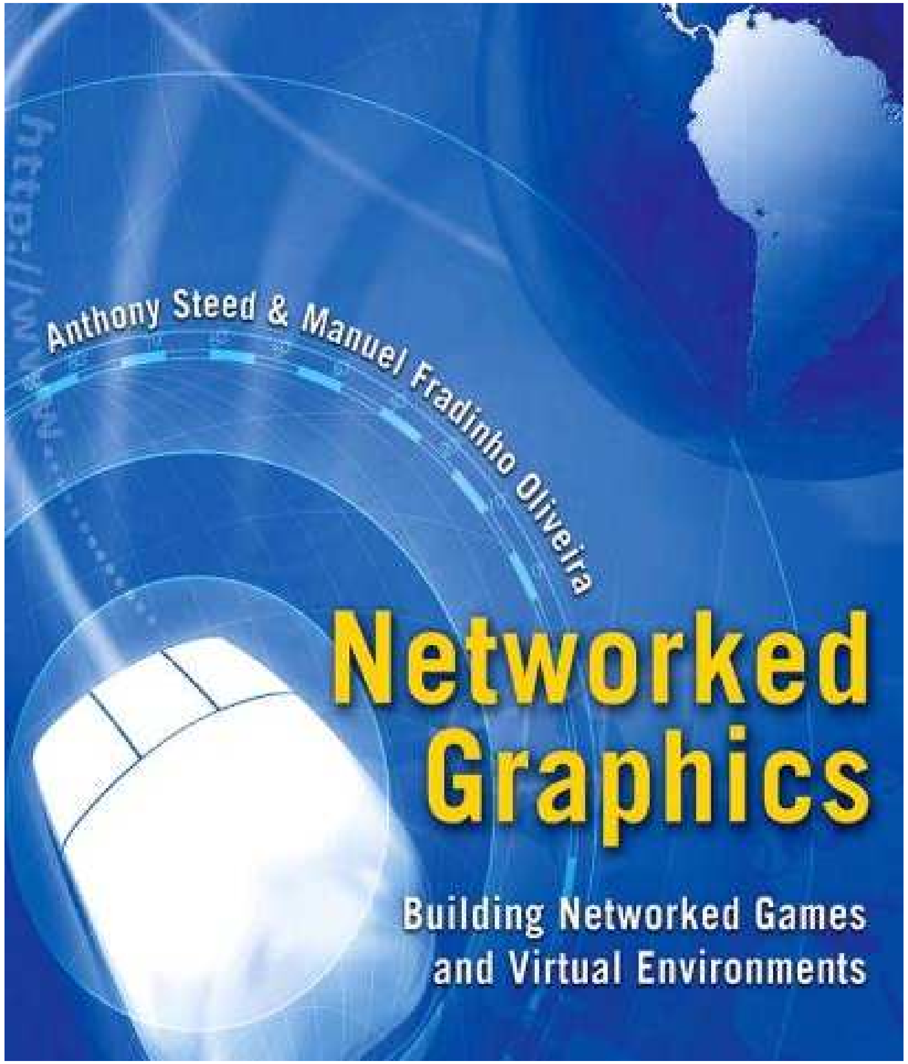
- *Application Support*
- - Security, Protocol decisions
- - Persistence
- *Tools*
- - Middleware
- - Networked engines
- *Research Issues*
- - Scalable peer-to-peer, thin clients
- - Standards, etc.

Materials

- See handout
- All materials available from speaker
- Also on:
  - http://www.networkedgraphics.org/materials/ieeevr-2011

- Many more materials (code, blog posts, paper links, etc.) on:
  - http://www.networkedgraphics.org/

For Much More Detail



- Anthony Steed & Manuel Oliveira
- Published by Morgan Kaufmann (2009 US, 2010 EU&Other)
- 576 pages
- Covers today's topics in more depth plus more background

Introduction

Introduction to Networked Graphics

IEEE Virtual Reality 2011

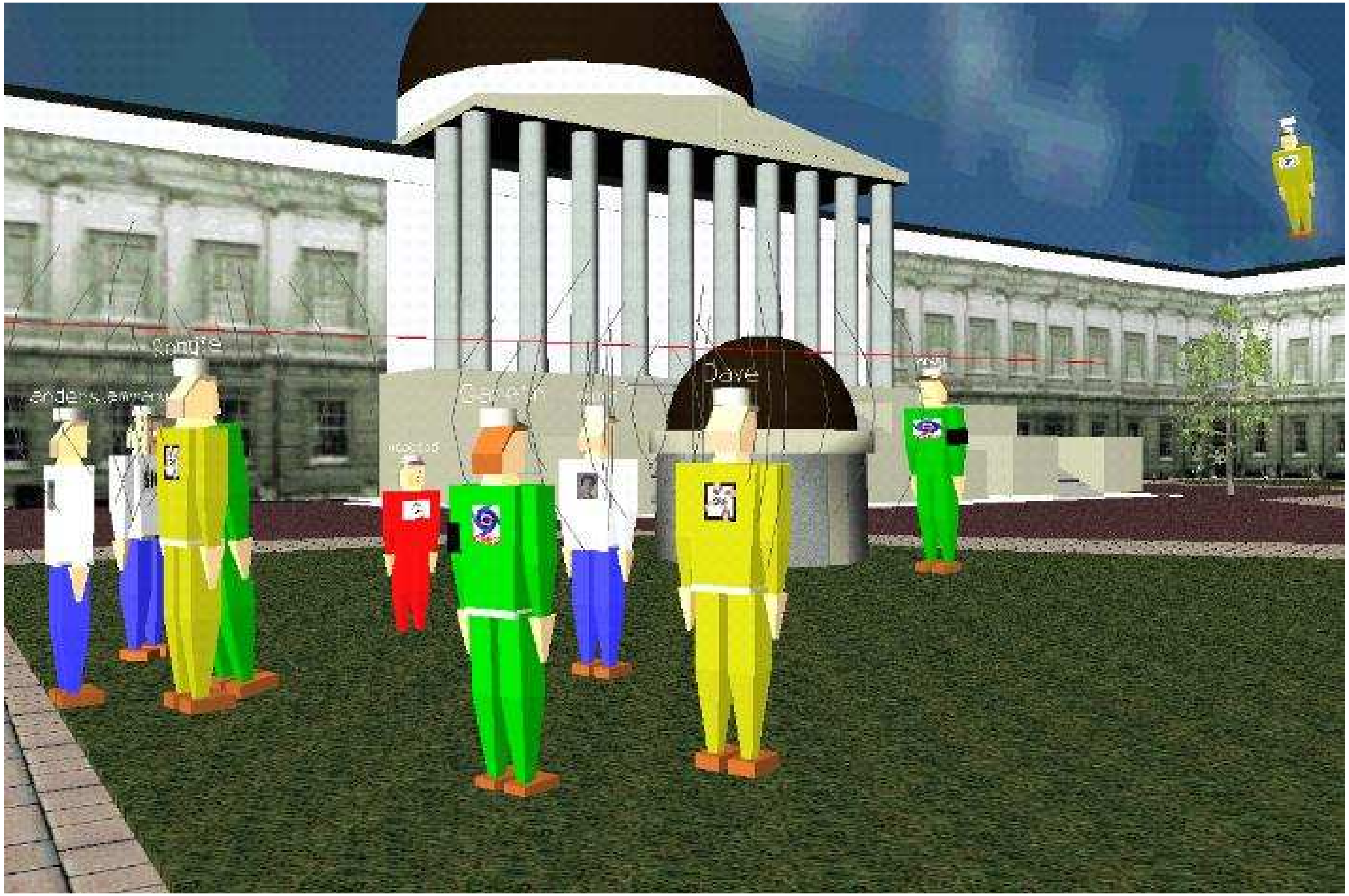
Anthony Steed

Introduction

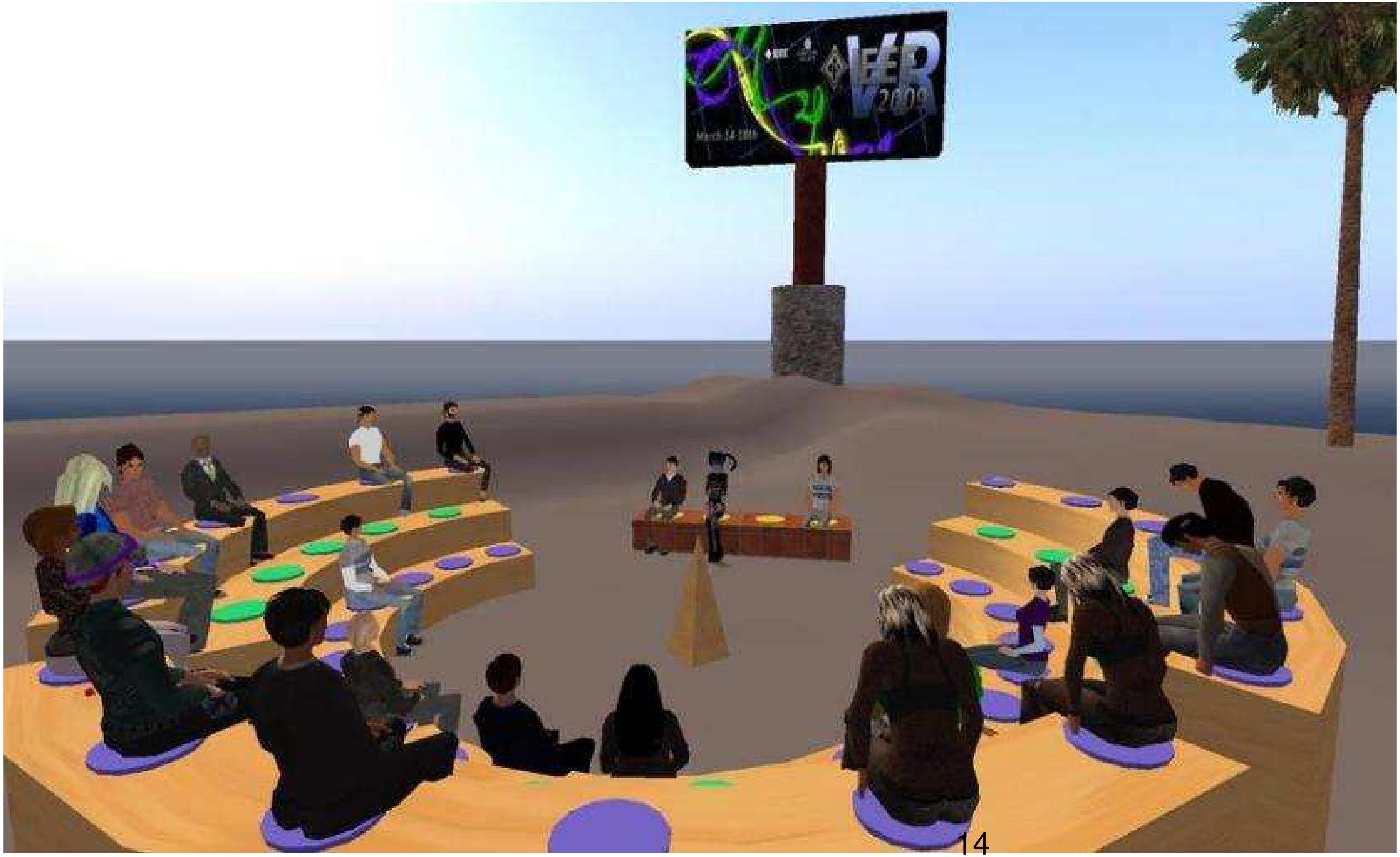
- What are networked graphics?
- Potted history of networked graphics

WHAT ARE NVES AND NGS?

DIVE



Second Life



Burnout™ Paradise

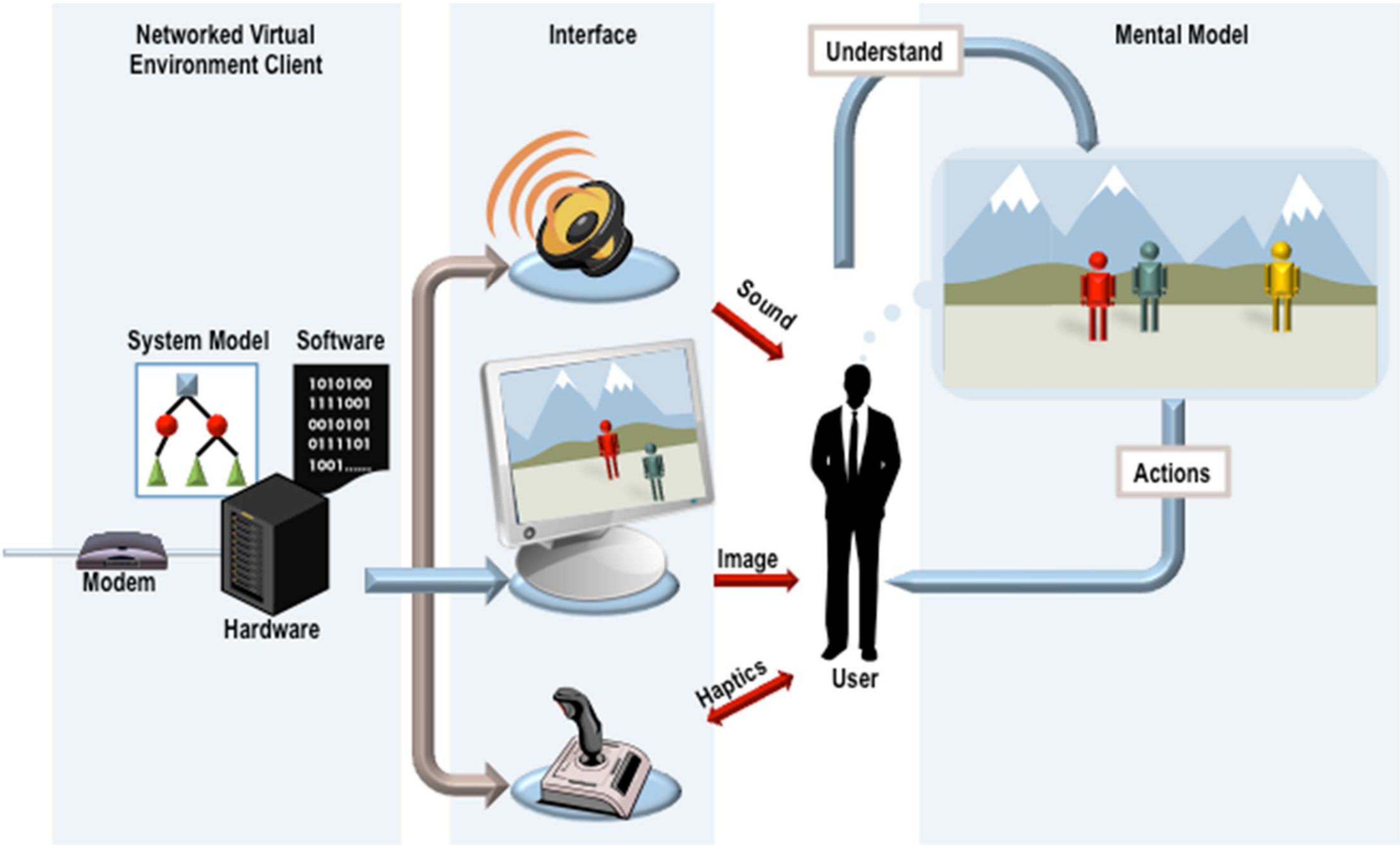
Common Themes

- 3D virtual environment
- Real-time changes
- Collaboration with other users
  - Representation of users in the world (typically as avatars, but also cars/tanks/etc.)
  - Text communication
  - Voice
- Virtual environment might mirror a real place but typically is a fantastic place

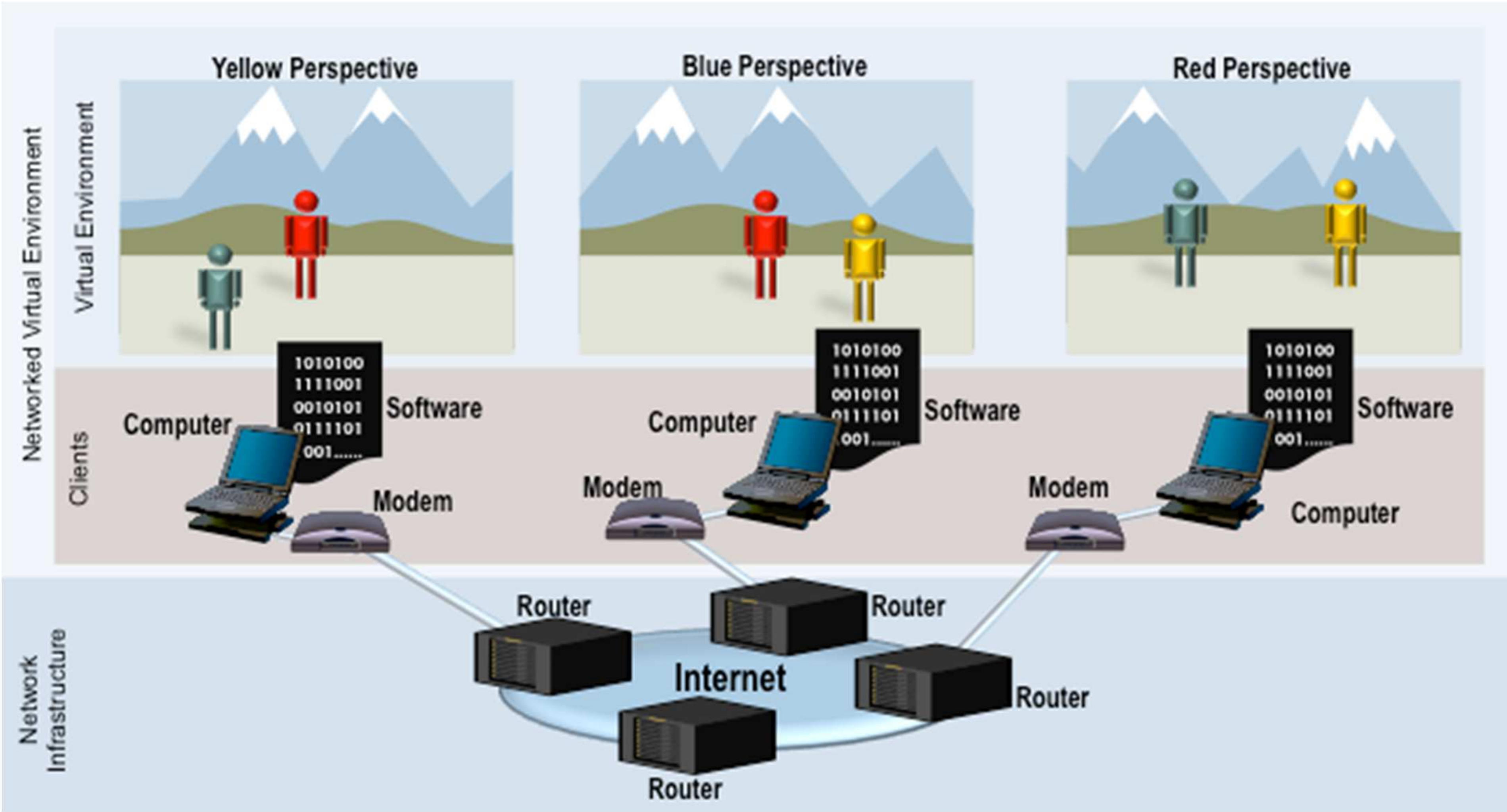
Common Themes

- One *client* is usually responsible for generating the view for one user
- A set of clients creates the *illusion* of a shared virtual environment
- "Illusion" because
  - Virtual environments can involve detailed *models*
  - Information about changes in models takes time to travel across communication links

Virtual Environment Client



Networked Virtual Environment



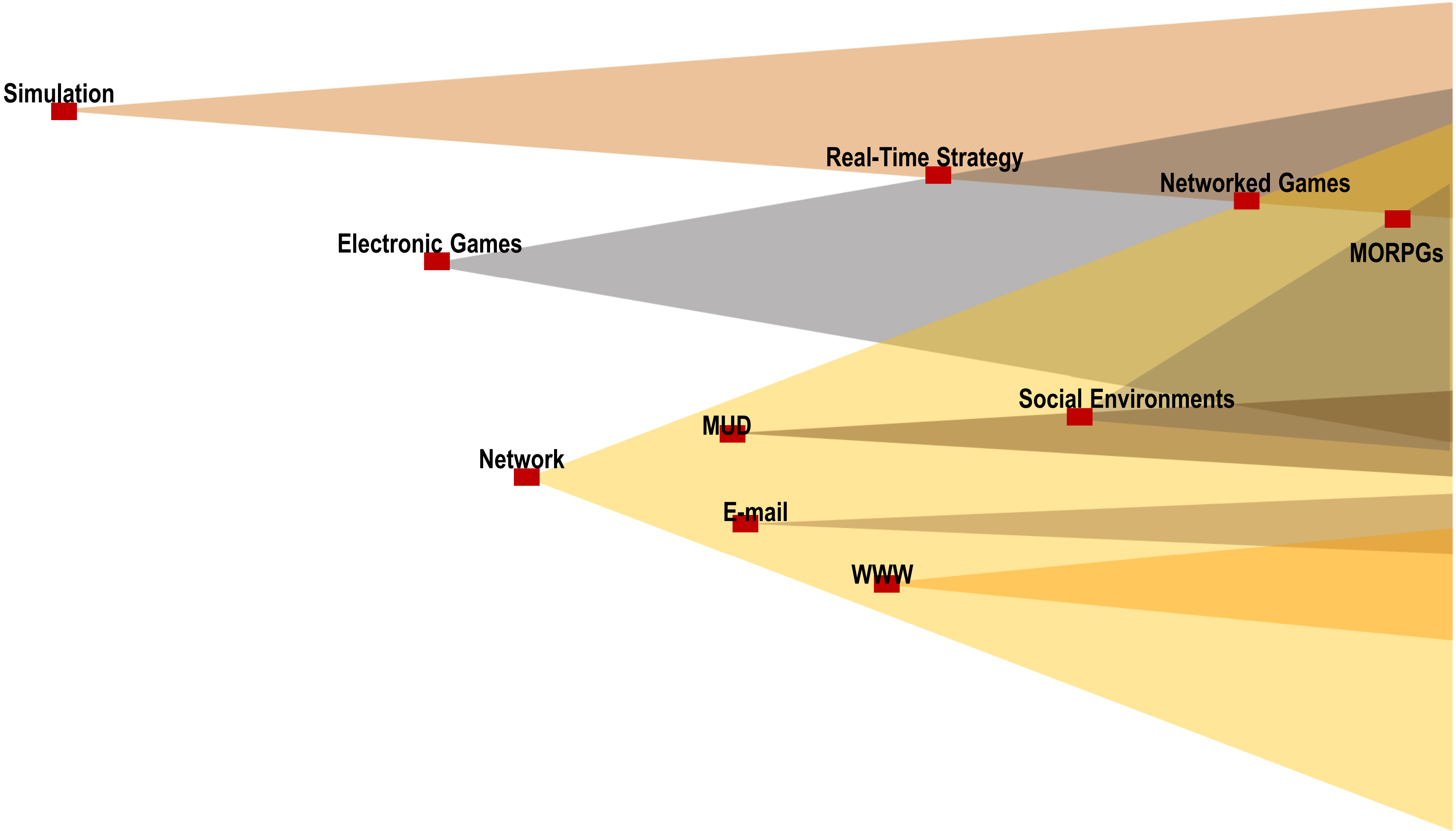
Consistency and Plausibility

- *Local plausibility* is the appearance of consistency of only local actions
- *Shared plausibility* is the appearance of properties being the same as observed by users
  - *Objects that are in the background need not be consistent*
  - *Further: only things that might be the focus of joint attention can be discussed and be different*
- A local implausibility might be an obvious thing to talk about!

Why NVEs are not Standard Network Applications

- Unlike video/audio streaming, or web browsing, in an NVE or NG client, networking is NOT the main activity: rendering probably is
- Some information changes very quickly and smoothly
  - E.G. player positions
- Can incorporate other web-enabled media
  - Audio/video
- Often require bulk download of assets

SOME HISTORY


Simulation
Real-Time Strategy
Networked Games
Electronic Games
MORPGs
Social Environments
MUD
Network
E-mail
WWW

ARPANET

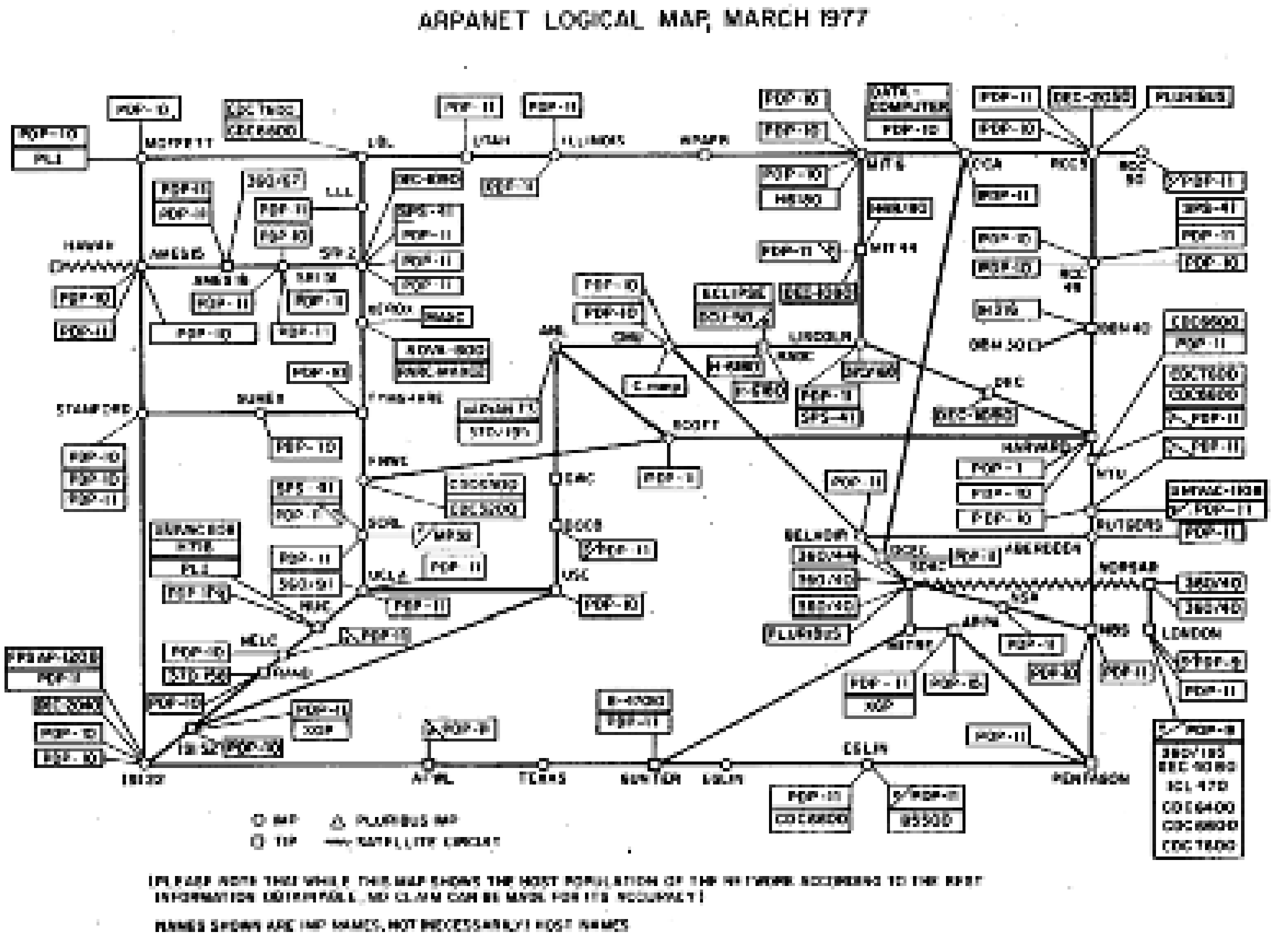
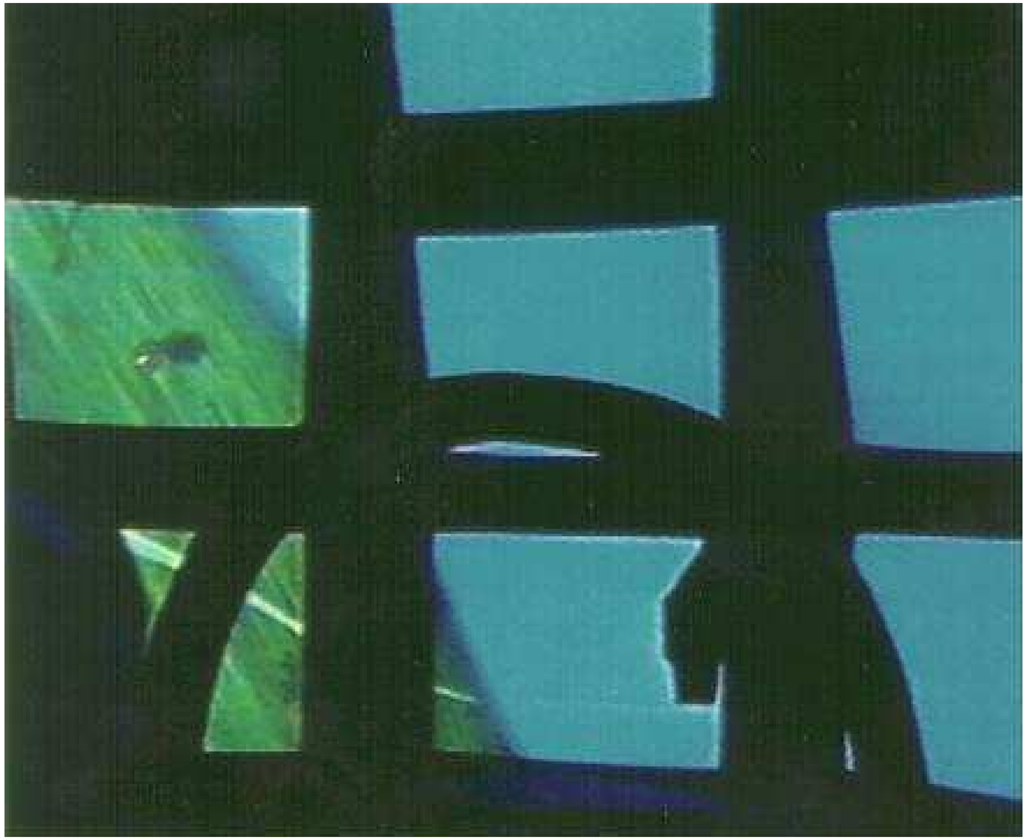


Diagram of ARPANET circa 1977

SIMNET



SIMNET aircraft simulation. Left: A view of the simulator control. Right: a view from the cockpit of an aircraft simulator. Images from (Harris, 1994)

MUD1

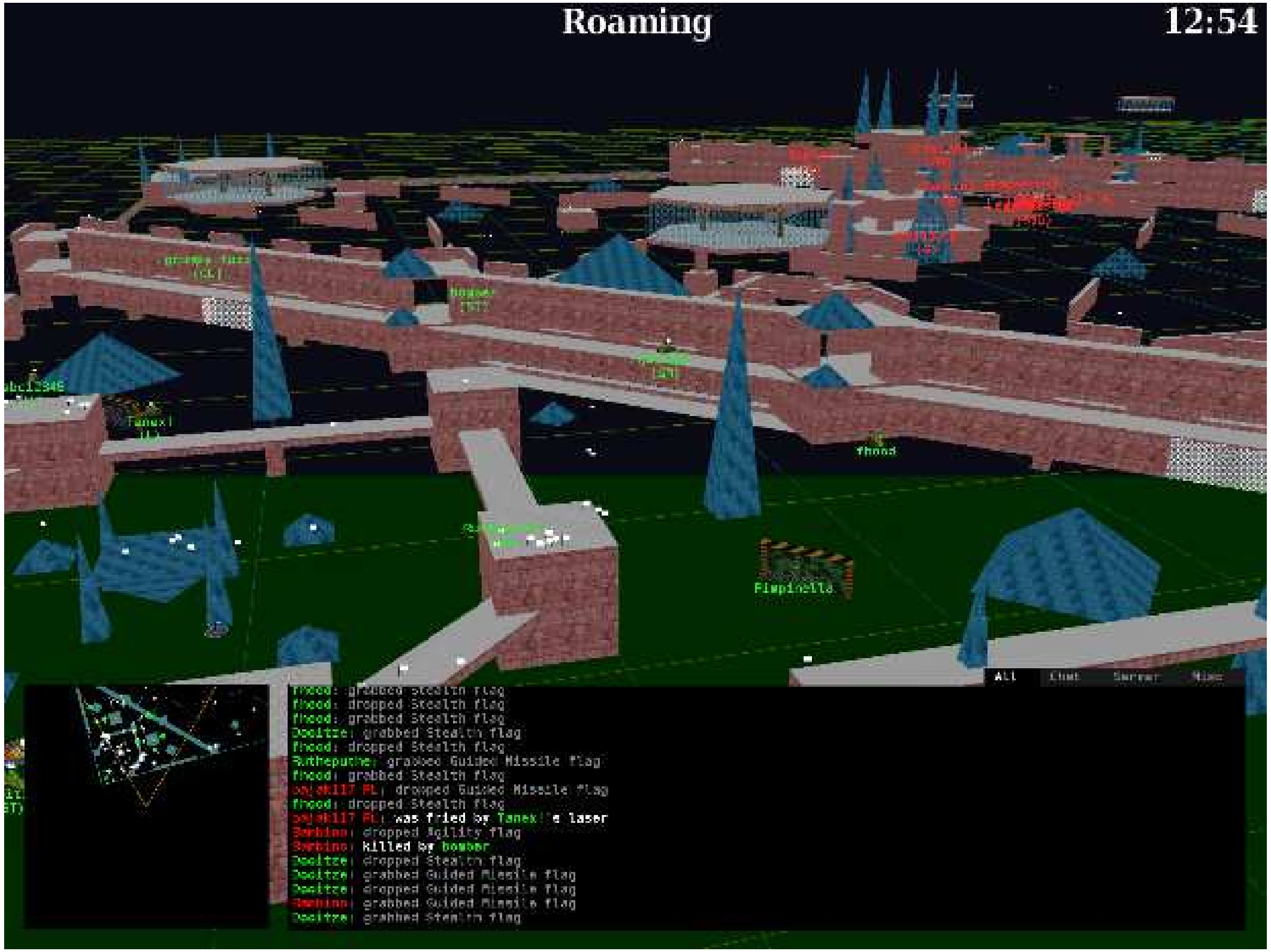
```
ideal atmosphere in which to relax before venturing out into that strange,
timeless realm. A sense of decency and decorum prevails, and a feeling of
kinship with those who, like you, seek their destiny in The Land. There are
exits in all directions, each of which leads into a wisping, magical mist of
obvious teleportative properties...
*n
Dense forest.
You are standing in some dense forest, which slopes down to the south.
*e
Dense forest.
You are standing in some dense forest, which slopes down to the south, where
stones and masonry indicate once stood buildings, over which the forest has
now extended.
*e
Ruin.
This conglomeration of stone which lies forlornly in the grass is all that is
left of the outer buildings belonging to the nearby tin mine, before bad
times fell upon it. Age and the weather have since reduced the once proud
structures to moss-covered misshapen arrangements of rubble. To the north is
a forest, and to the east an enormous cliff at the base of a huge,
snow-capped mountain. Elsewhere is a pony paddock.
There is a thick, silver-looking cord here, coated in a strange, clear
substance.
*get cord
```

A screenshot of the text interface of the MUD1 system over a standard telnet client

Maze (Maze War)

Maze War running at on Imlac PSD-1D at DigiBarn's Maze War 30 Year Retrospective (Digibarn, 2004)

BZFlag



A screenshot of the current BZFlag

DOOM

DOOM™ (iD Software) was the first multi-player first-person shooter to reach wide-spread public attention

Quake

Quake (id Software) brought true 3D and was the basis for several licensed games. Left: the original Quake game. Right: Counter-Strike (Valve Software), originally a modification of the game Half-Life which was based on Quake engine.

Meridian 59

Meridian 59. Left: a view of the client. Right: an army of skeletons invades the Streets of Tos. This screenshot is from a more recent version of the client software.

Ultima Online

Pirates demanding tribute (Schultz, A., 2009)

Active Worlds

Left: Map of Alphaworld in December 1996. Right: Map of Alphaworld in August 2001.

Networking Fundamentals

Introduction to Networked Graphics

IEEE Virtual Reality 2011

Anthony Steed

Networking Fundamentals

- Basic Internet technologies
- Basic networking strategies
- Why are all types of networked graphics “non-standard” networking applications?

BASIC INTERNET TECHNOLOGIES

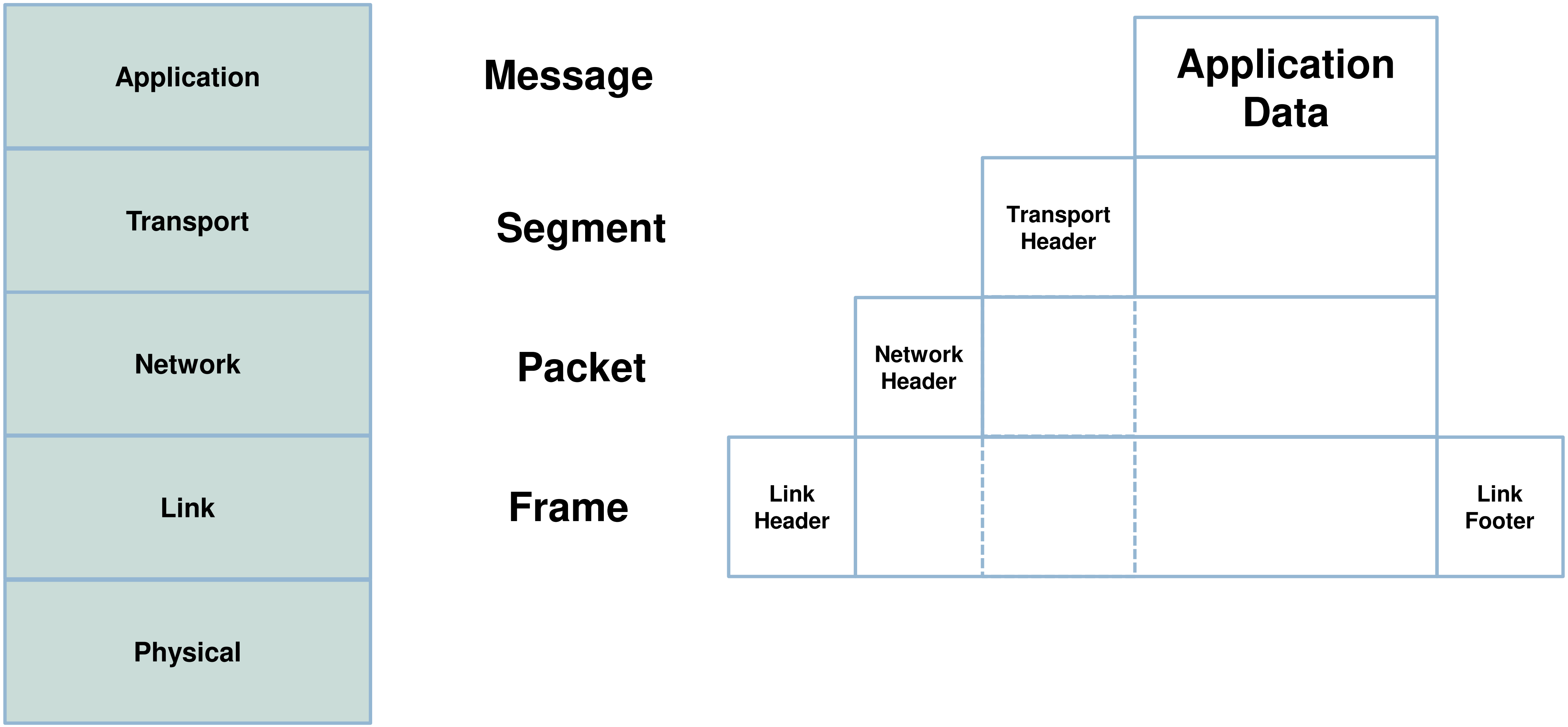
Introduction

- Networks at the heart of *networked* VR
- Many network protocols are out there
  - TCP, UDP, multicast, RTP, etc
  - Choice based on needs
- Properties of the Internet

IP Stack

Application	**DHCP, DIS, DNS, FTP, HTTP, IMAP, RTP, SMTP, SSH, Telnet**
Transport	**TCP, UDP, RSVP**
Network	**IP, ICMP, IGMP**
Link	**Ethernet, 802.11, ADSL**
Physical	**copper wires, fibre-optic cable, radio waves**

From Messages to Frames



APPLICATION LAYER

Application Layer Protocols

- Determine what *messages* are sent between applications
  - Messages defined by *syntax* and *semantics*
- Various standards for messages, typically set by RFCs (Requests for Comments) hosted by the IETF (Internet Engineering Task Force)

E.G. HTTP Request

- If you connect to ***Host*** www.cs.ucl.ac.uk at ***Port*** 80
- And then issue (type!) in ASCII the following message:

```
GET /staff/A.Steed/ HTTP/1.1
Host: www.cs.ucl.ac.uk
```

- And issues (type) two carriage returns
- You get …

…

```
HTTP/1.0 200 Document follows
MIME-Version: 1.0
Server: CERN/3.0
Date: Sun, 08 Feb 2009 15:25:18 GMT
Content-Type: text/html
Content-Length: 16150
Last-Modified: Wed, 21 Jan 2009 17:42:00 GMT

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" dir="ltr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="keywords" content="A. Steed, Anthony Steed, Department of Computer Science, University
College London, virtual environments, virtual reality, computer graphics" />
```

Application Protocol Descriptions

- Often ASCII preamble with binary assets inserted at known or marked positions
- Some messages are designed to be carried over a reliable stream and are of unknown length (likely to be over TCP)
- Some messages are small and it is not important if they get lost (likely to be over UDP)

Common Application Protocols

Service Full Name	Short Name	Port	Transport
File Transfer Protocol	ftp	21	tcp
Simple Mail Transfer	smtp	25	tcp
Domain Name System	dns	53	udp
Finger	finger	79	tcp
HyperText Transfer Protocol	http	80	tcp
Post Office Protocol (Version 3)	pop3	110	tcp
Internet Message Access Protocol	imap	143	tcp
Hypertext Transfer Protocol Secure	https	443	tcp
File Transfer Protocol Secure	ftps	990	tcp
Distributed Interactive Simulation	dis	3000	udp
BZFlag Game Server	bzflag	5154	tcp
Quake Game Server	quake	26000	udp

Domain Name Service (DNS)

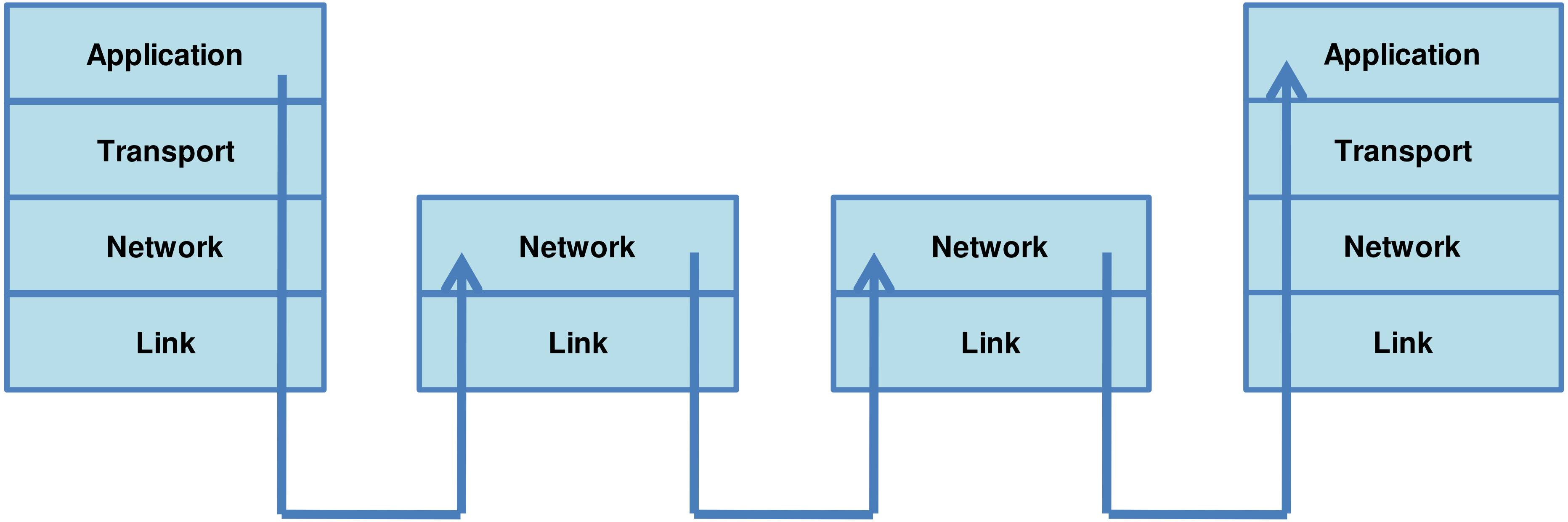
- Maps fully qualified domain names (narok.cs.ucl.ac.uk) to their IP addresses (128.16.5.123)
- Is a network service, thus takes time
- Time is variable because it's a hierarchical search
- Local DNS caches query responses for a time (e.g. 24 hours)
- Otherwise needs to query a canonical domain

TRANSPORT LAYER

Transport Layer Protocols

- User Datagram Protocol (UDP)
  - Send a message (datagram) and forget about it
  - No guaranteed delivery
  - No guaranteed ordering
- Transmission Control Protocol (TCP)
  - Guaranteed, in-order stream of data from one host to another

End to End Principle



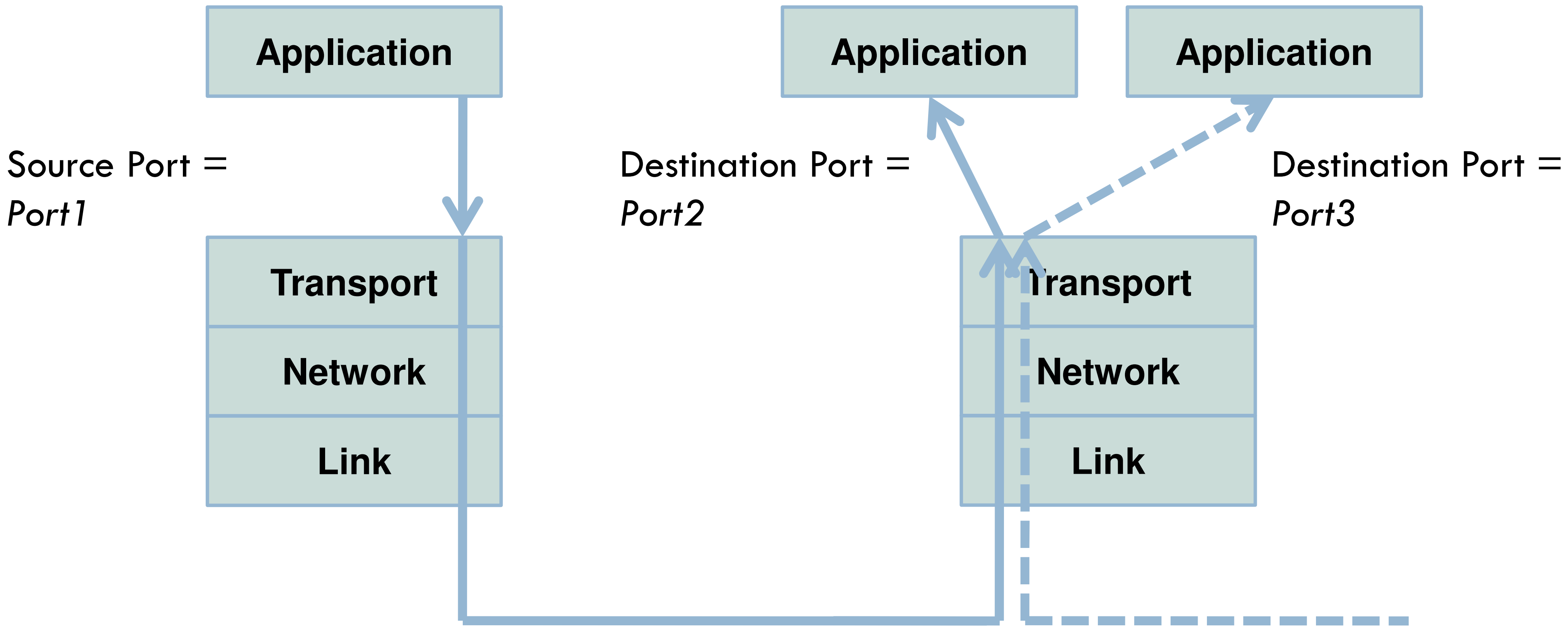
Only the sender and receiver understand TCP or UDP (or other same or higher-level protocols). The other *routers* in the Internet do not.

UDP

- All hosts on the Internet have an IP address
- How does the network know which program wants it?
- You additionally need (for UDP and TCP) a port number
  - These are 16 bits numbers, so must lie in the range 0-65535
  - Some are reserved, see later
- Processes listen for incoming UDP packets
- Need to check the packet for consistency

UDP Segment Layout

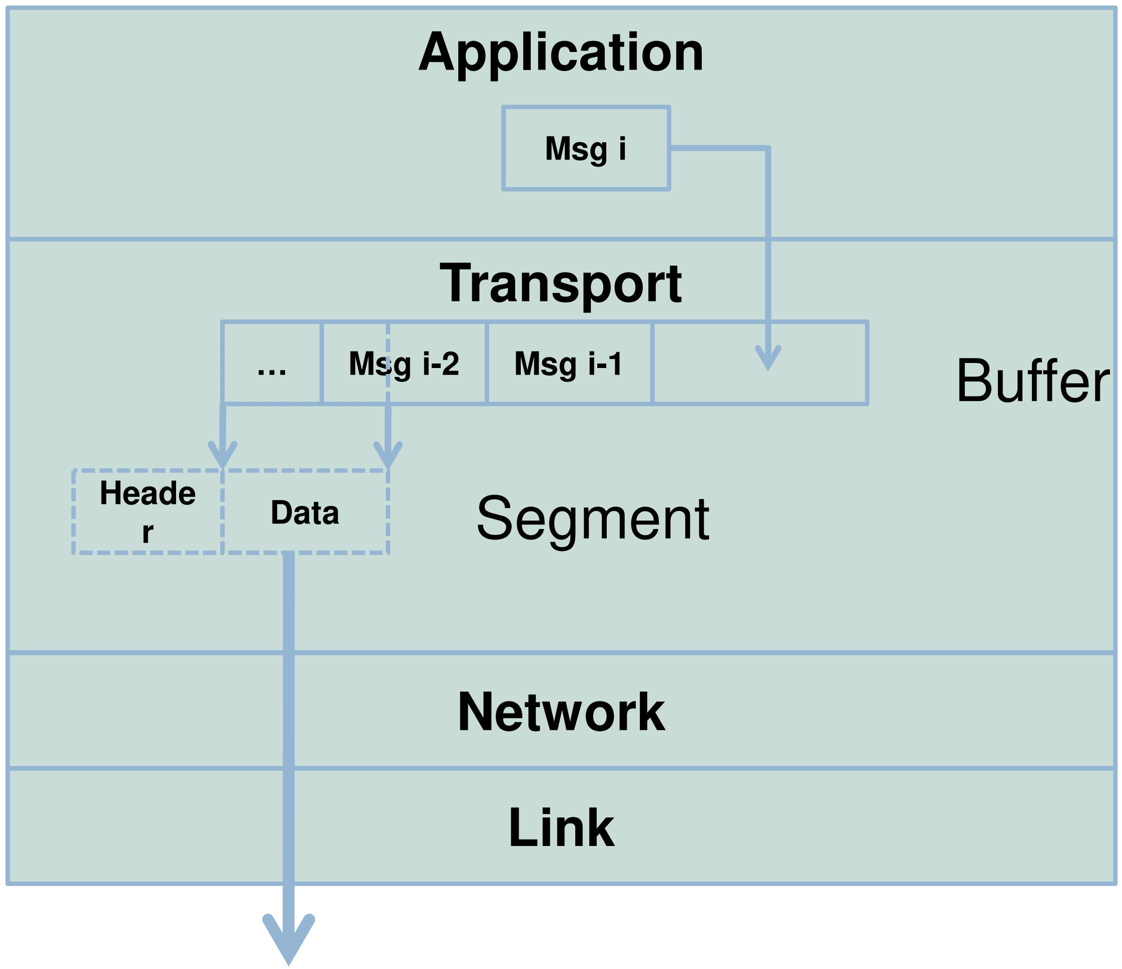
Bits	0 15	16 31
0-31	Source Port	Destination Port
32-63	Length	Checksum
64+	Data	



Application
Application
Application
Source Port =
Port1
Destination Port =
Port2
Destination Port =
Port3
Transport
Network
Link
Transport
Network
Link

TCP

- In comparison to UDP, TCP offers:
  - A connection-oriented services with bi-directional (full-duplex) communication
  - Reliable transmission of messages in each direction
  - Congestion avoidance, using variable rate transmission
  - In order, and non-duplicate delivery of information
- Applications add messages to an outgoing buffer
- The buffer is streamed in packets to the receiver
- The receiver reconstructs the buffer and extract packets

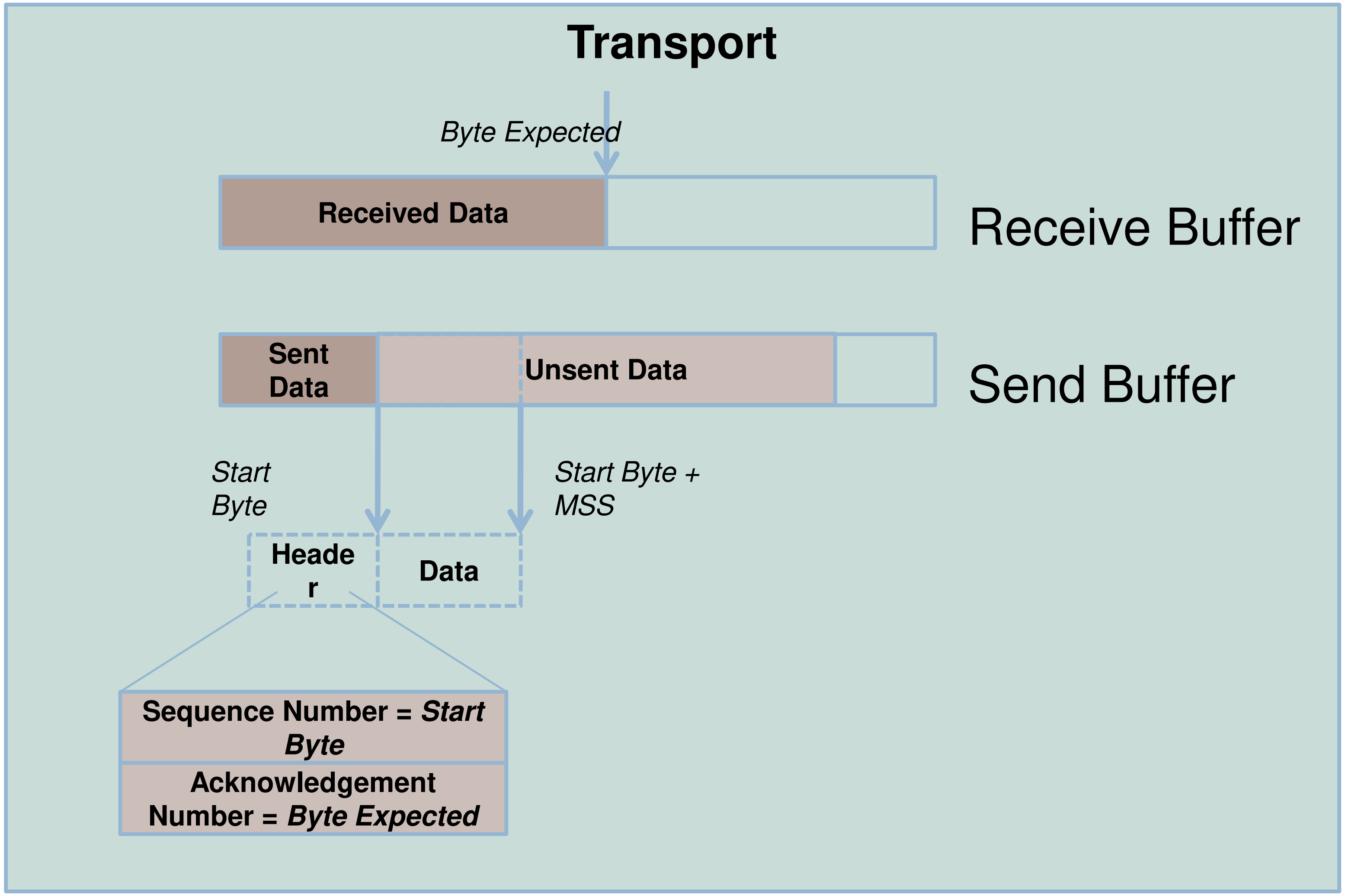

Application
Msg i
Transport
...
Msg i-2
Msg i-1
Buffer
Header
Data
Segment
Network
Link

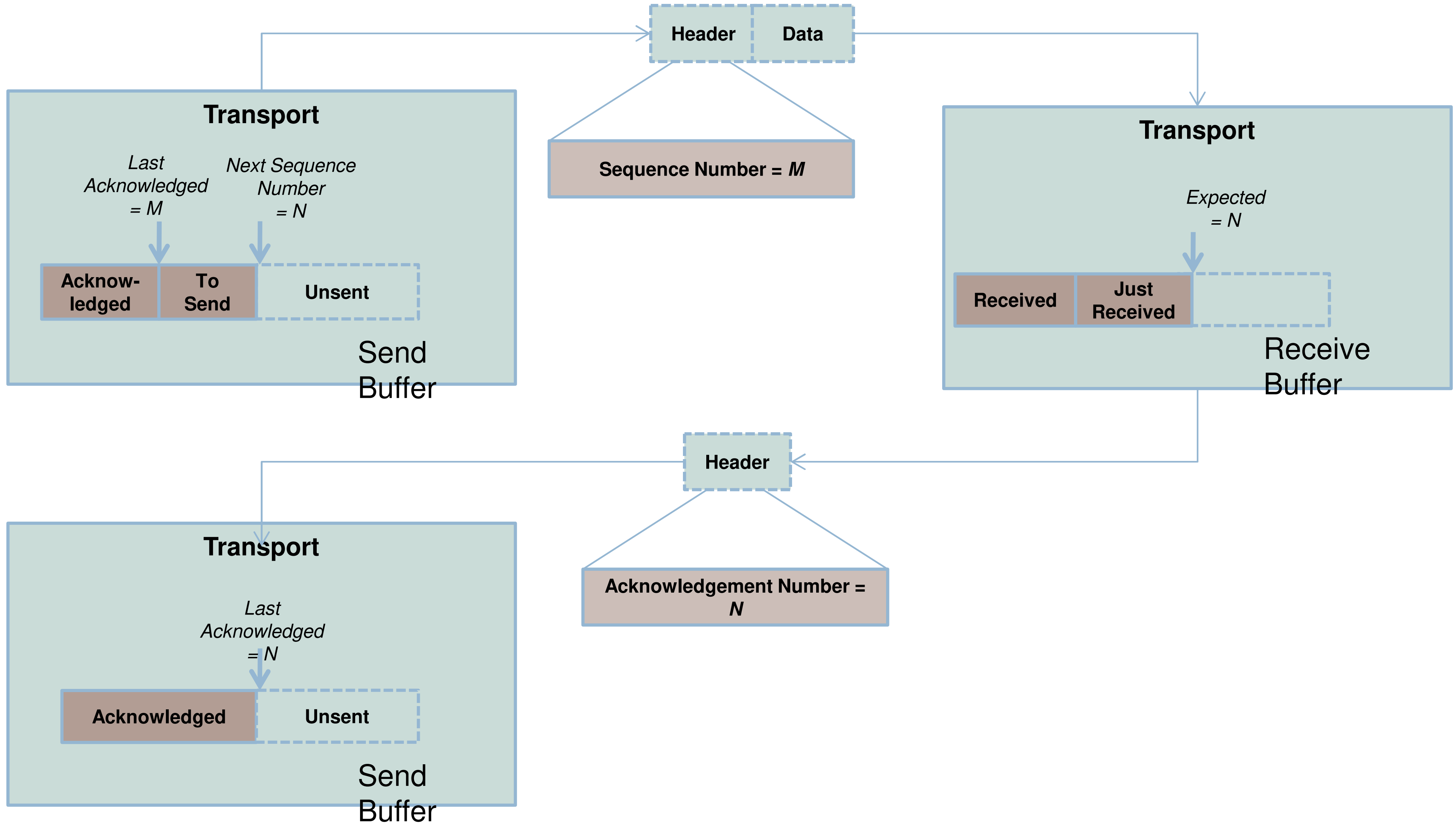
Layout of a TCP Segment

Bits	0 15			16 31
0-31	Source Port			Destination Port
32-63	Sequence Number			
64-95	Acknowledgement Number			
96-127	Data Offset	Not Used	Flags	Receive Window
128-159	Checksum			Urgent Pointer
160-191	Options (Optional)			
160+ 192+, 224+, etc.	Data			

TCP is Bi-Directional

- Even if, logically, data only flows one way, in order to ensure reliability, we need return data which tells us which data has been successfully received (ACK)
- The sender must maintain the buffered data until it receives an ACK

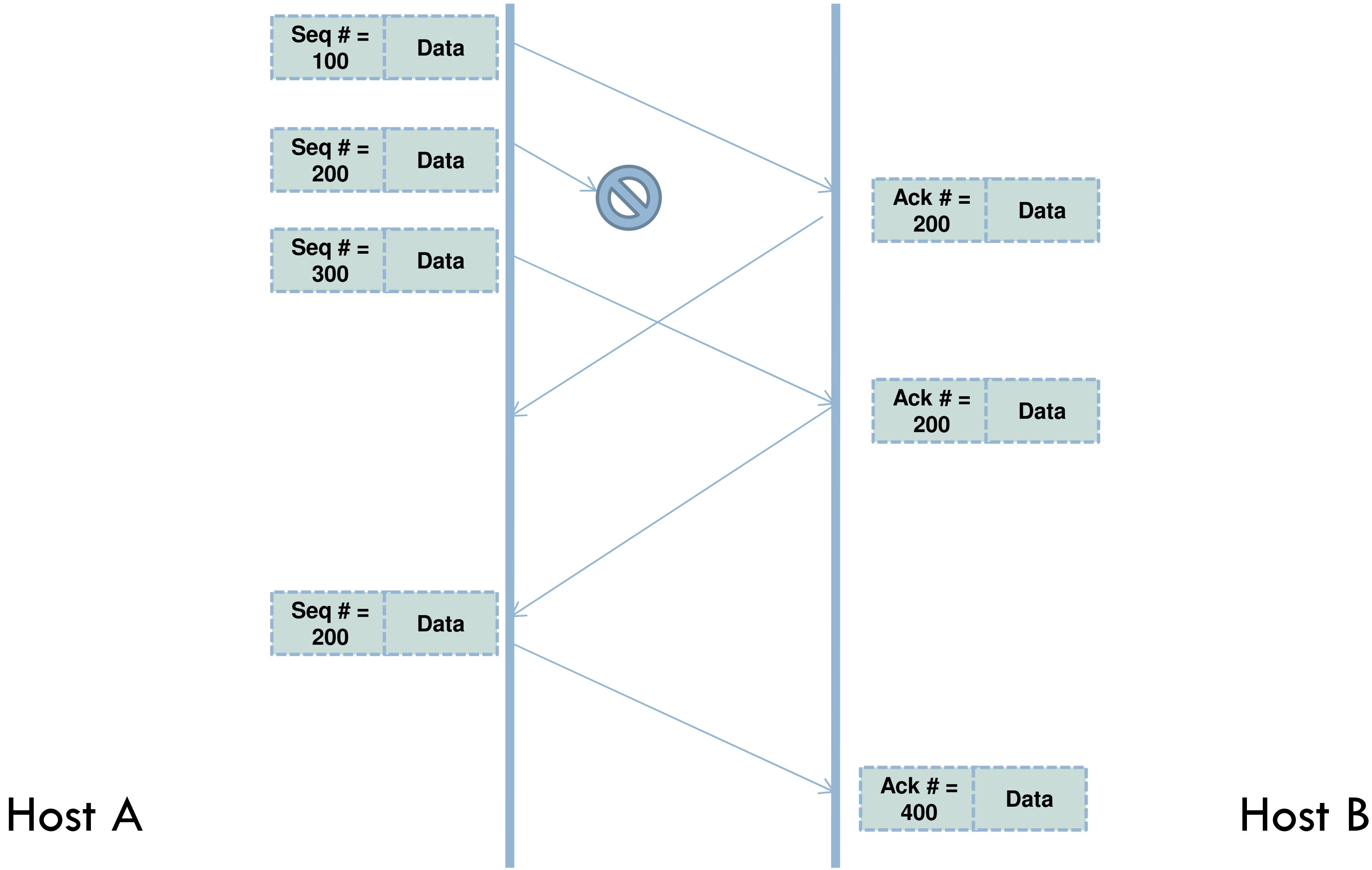

Transport
Byte Expected
Received Data
Receive Buffer
Sent Data
Unsent Data
Send Buffer
Start Byte
Start Byte + MSS
Header
Data
Sequence Number = Start Byte
Acknowledgement Number = Byte Expected


Header
Data
Transport
Last Acknowledged = M
Next Sequence Number = N
Acknow-ledged
To Send
Unsent
Send Buffer
Sequence Number = M
Transport
Expected = N
Received
Just Received
Receive Buffer
Header
Transport
Last Acknowledged = N
Acknowledged
Unsent
Send Buffer
Acknowledgement Number = N

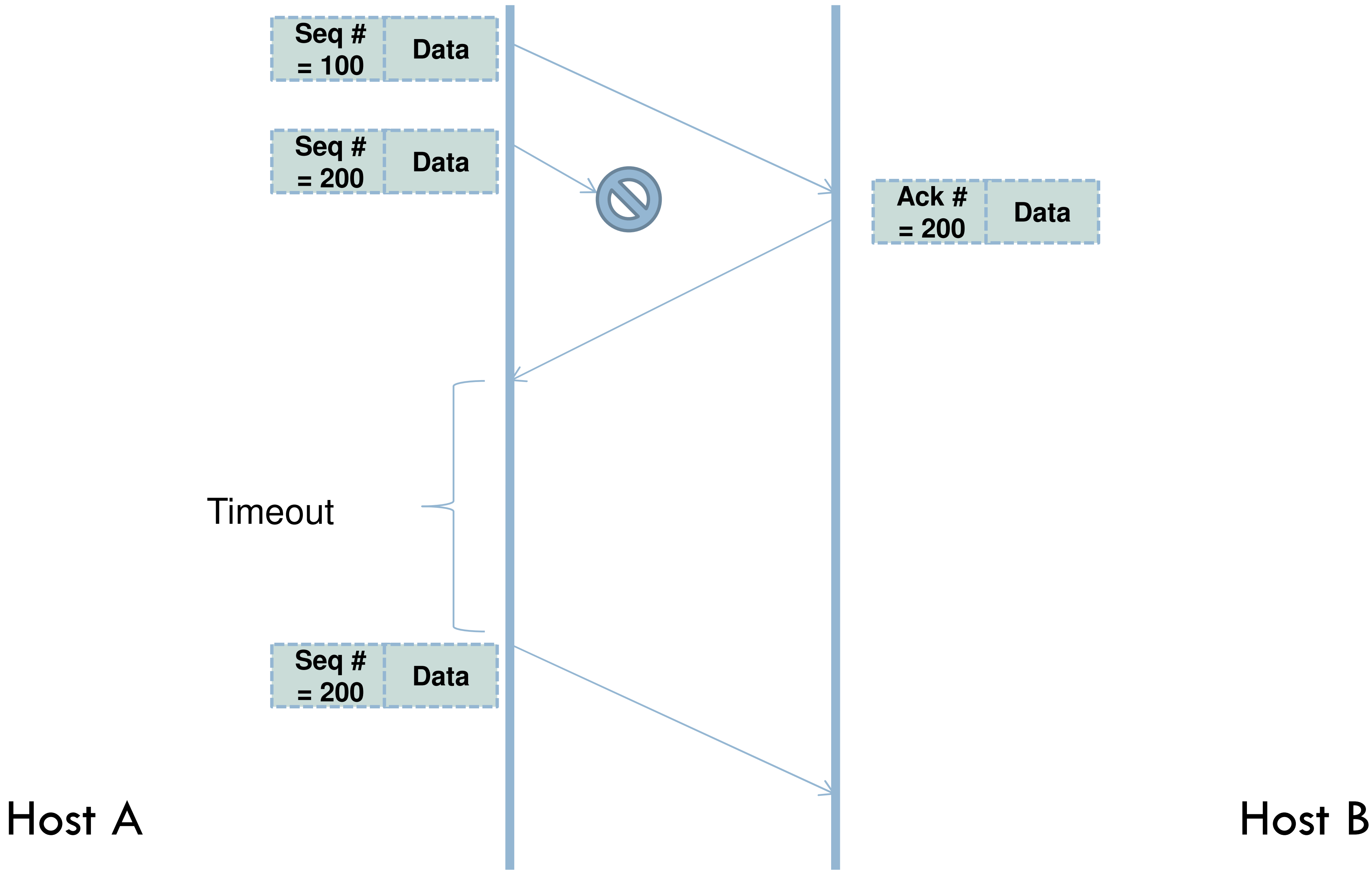
TCP Reliability

- How to detect if something has gone missing
  - A timeout
  - Returning an ACK repeatedly which indicates the buffer hasn't grown

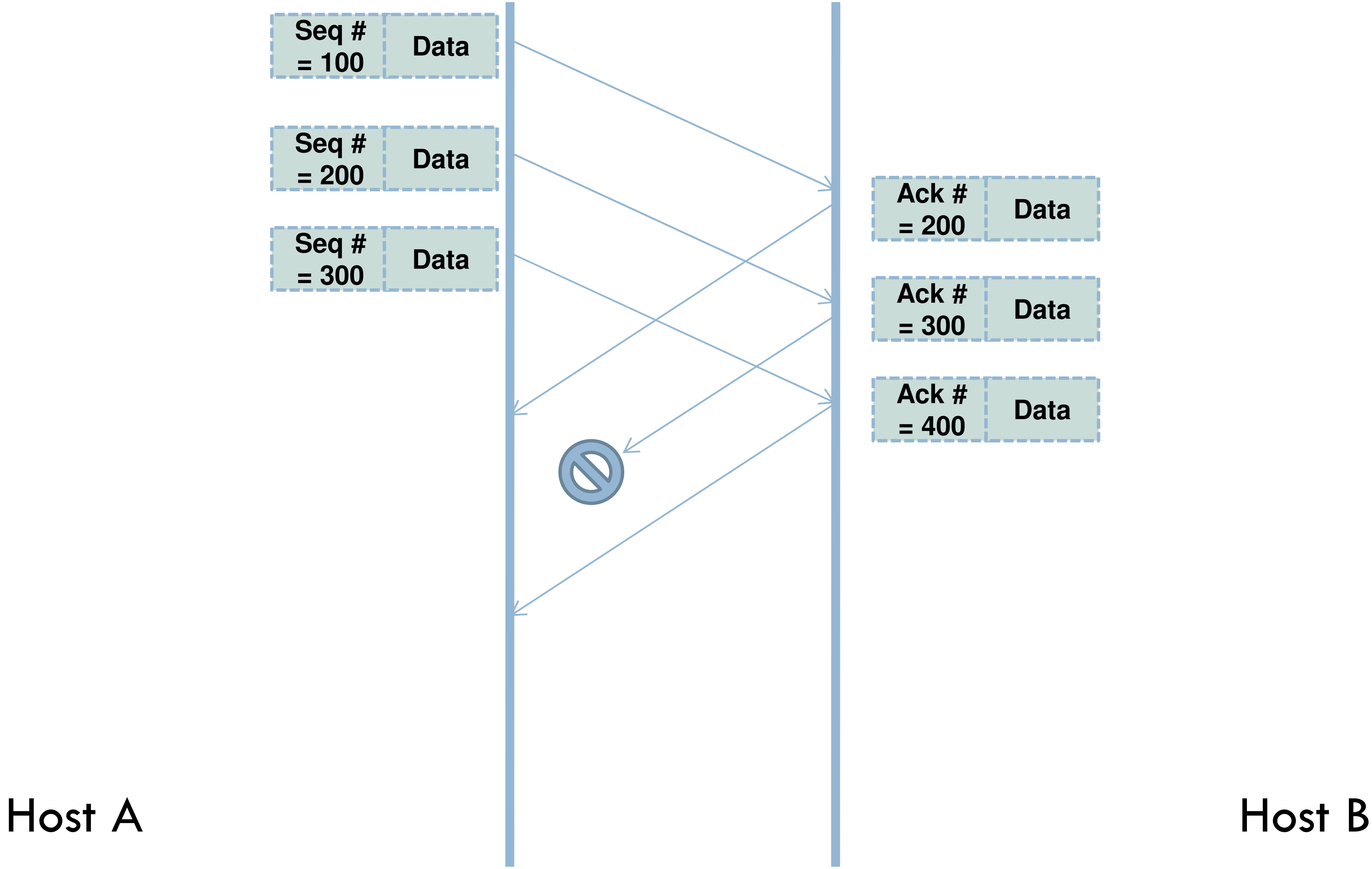
Packet Resent on Duplicate ACK



Packet Resent on Timeout

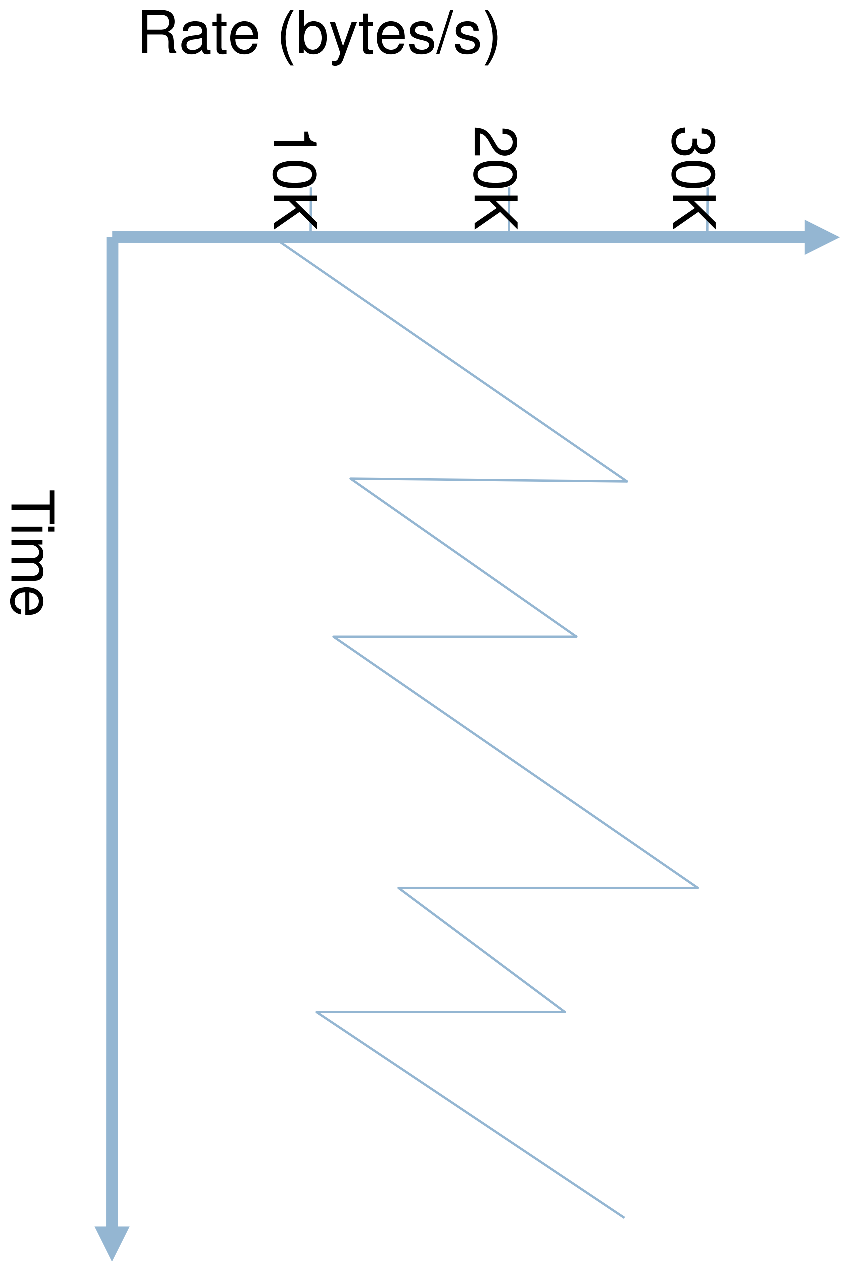


A Lost ACK Doesn't Matter



TCP Fairness

- How does TCP decide when to send packets (with UDP you call “send”)?
- It sends packets within increasing frequency but when they start going missing, it halves its rate
- There are LOTS of variants of TCP
- Protocols are often tested to see if there are TCP-fair, i.e. if N streams share a network link they get 1/N of the bandwidth
- UDP protocols are often NOT TCP-fair, you need to add that functionality yourself


Rate (bytes/s)
10K
20K
30K
Time

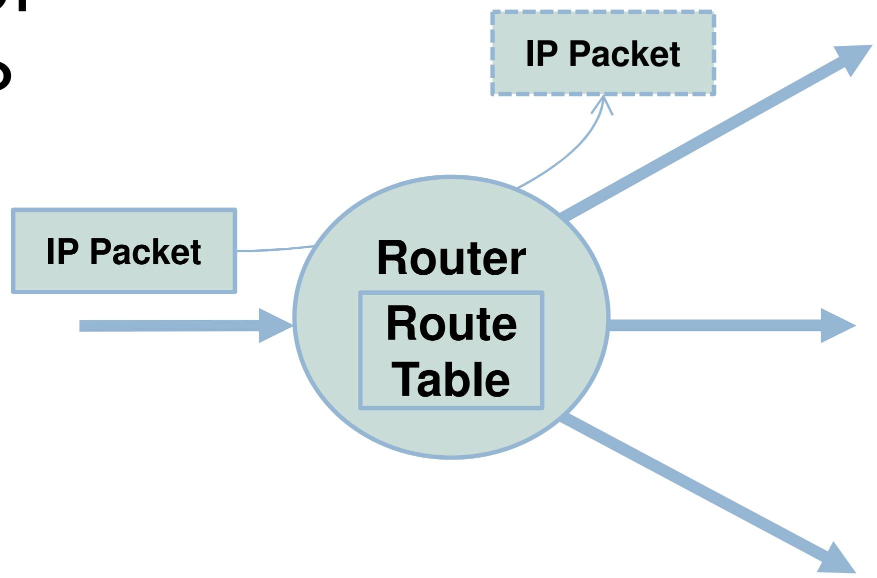
Observations

- If there is lots of data to send TCP can fill up IP packets, UDP might waste network capacity
- There are potentially lots of ACK packets in TCP
- TCP is slow to start, UDP is rapid start
- UDP protocols need to play fair when there is congestion

NETWORK LAYER

Network Layer

- The Internet is a collection of machines that understand IP packets
- A network routes packets from one host to another through routers



IPv4

- In IPv4 addresses are 32 bits in the form 128.16.13.118
- They are running out and IPv6 is ready to be deployed

IP Packets

- Key problem is what happens if links support frames of different size
  - E.G. Ethernet is 1500bytes
- The solution is that IP supports packet fragmentation, where a large packet is broken in to smaller ones: the end point must then reassemble them
- Obviously try to pick a *maximum transmission unit (MTU)* that avoids this

IP Packet Format

Bits	0 15			16 31	
0-31	Version	Header Length	Type of Service	Total Length	
32-63	Identification			Flags	Fragment Offset
64-95	Time to Live		Protocol	Header Checksum	
96-127	Source Address				
128-159	Destination Address				
160-191	Options (Optional)				
160+ 192+, 224+, etc.	Data				

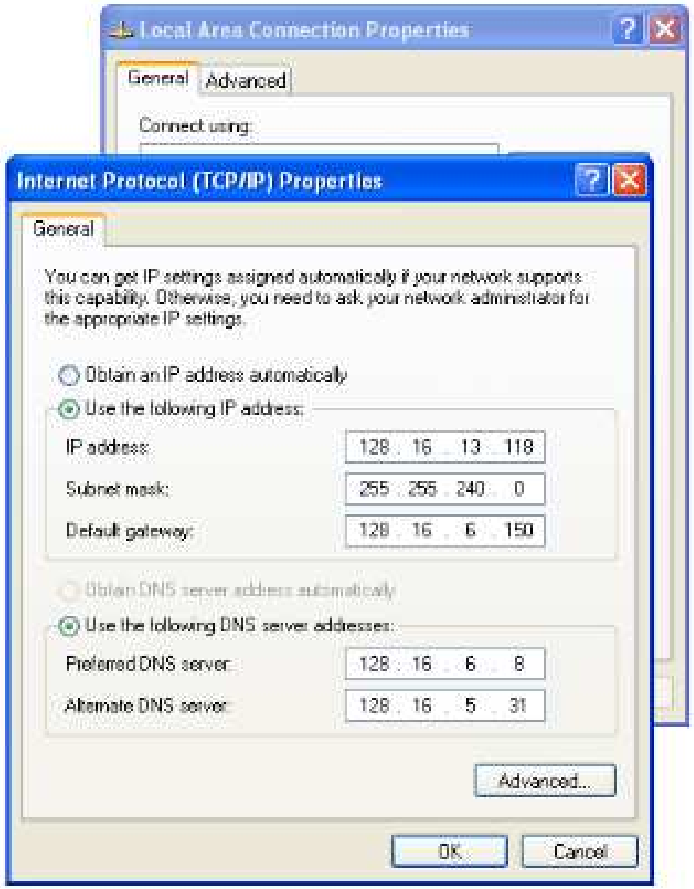
Protocol Types

- This is necessary to tell the receiver what the IP packet contains. E.G.s
  - 1: Internet Control Message Protocol (ICMP)
  - 2: Internet Group Management Protocol (IGMP)
  - 6: Transmission Control Protocol (TCP)
  - 17: User Datagram Protocol (UDP)
  - 89: Open Shortest Path First (OSPF)

LINK AND PHYSICAL LAYER

Link and Physical Layer

- The one we all have experience with is Ethernet, either wired or wireless
- Our experience is that for a specific Ethernet interface, we either need to:
  - Set IP address manually
  - Get an address automatically using DHCP
- DHCP is actually an application-layer protocol
- In both cases, we are making a mapping between the MAC address of the Ethernet adapter and the IP address



Local Area Connection Properties
General
Advanced
Connect using:
Internet Protocol (TCP/IP) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address: 128 . 16 . 13 . 118
Subnet mask: 255 . 255 . 240 . 0
Default gateway: 128 . 16 . 6 . 150
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server: 128 . 16 . 6 . 8
Alternate DNS server: 128 . 16 . 5 . 31
Advanced...
OK
Cancel

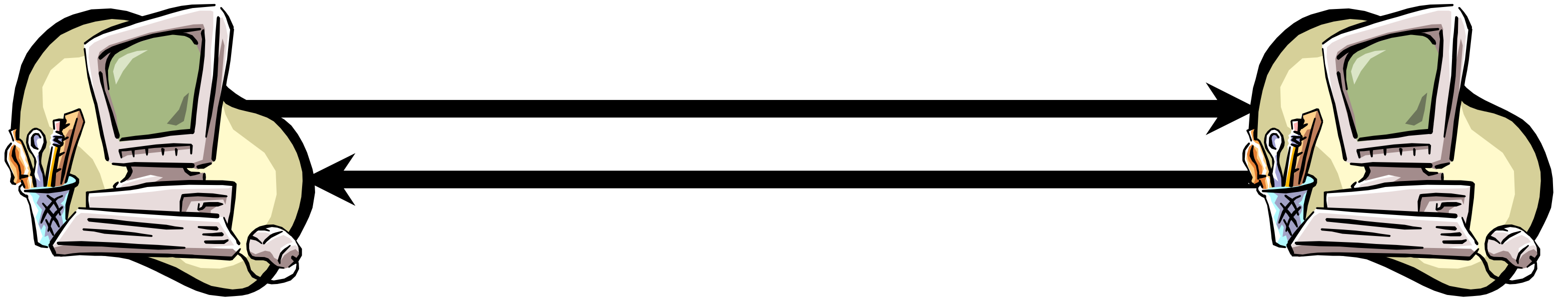
Bits	0 15	16 31
0-31	Destination MAC Address …	
32-63	… Destination MAC Address	Source MAC Address …
64-95	…Source MAC Address	
96-127	EtherType	Data
…	Data	
…	CRC Checksum	

BASIC NETWORKING STRATEGIES

Architectures

- Peer to Peer
- Client/Server
- Hybrid

Consider Just Two Machines

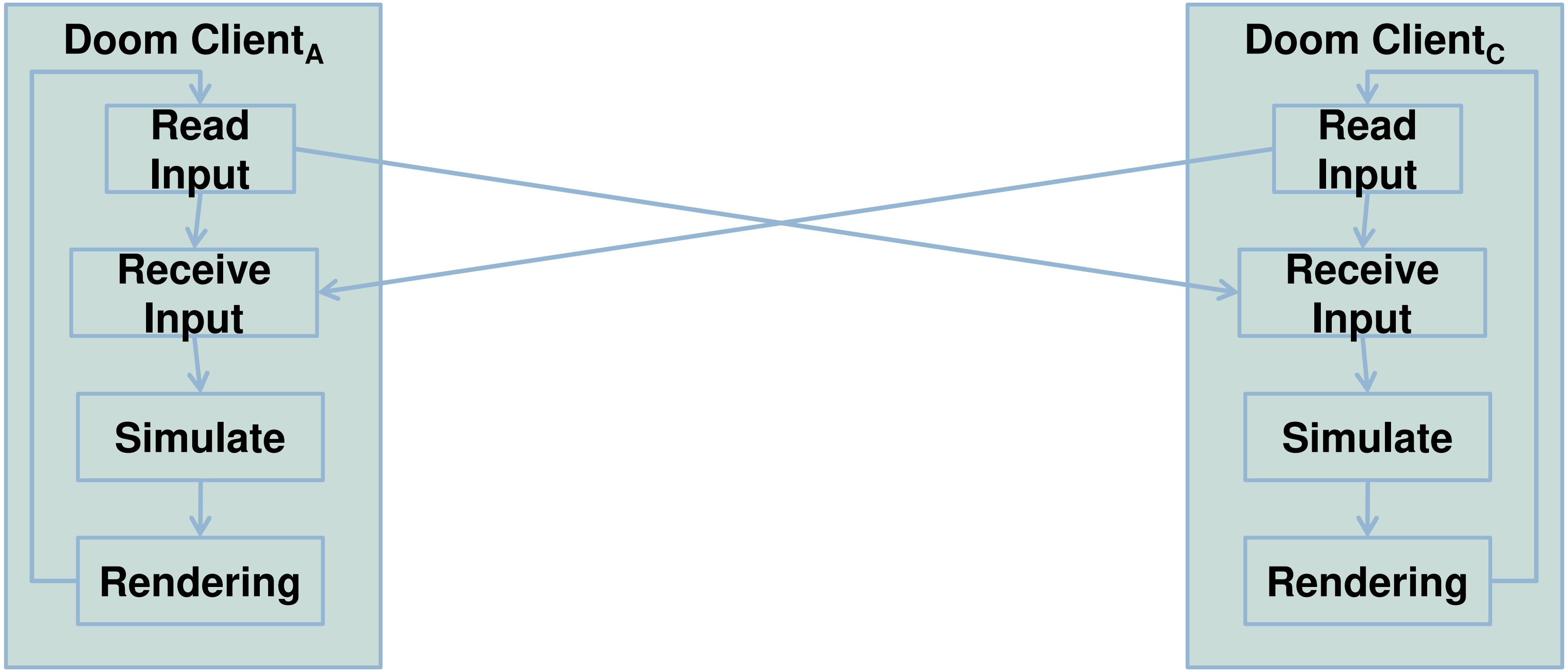


- What is the relationship between them?
- Peers?
- Master/slave? Client/server?
- Does one have data the other one does not?

Peer to Peer with Two Clients

- Need to decide separation of responsibilities
  - E.G. Each client simulates one player's actions
- Need to communicate sufficient information to the other that they can get both get the same state
- Assumes that they have the same information other than real-time input
- Can be achieved simply with sending input to each other

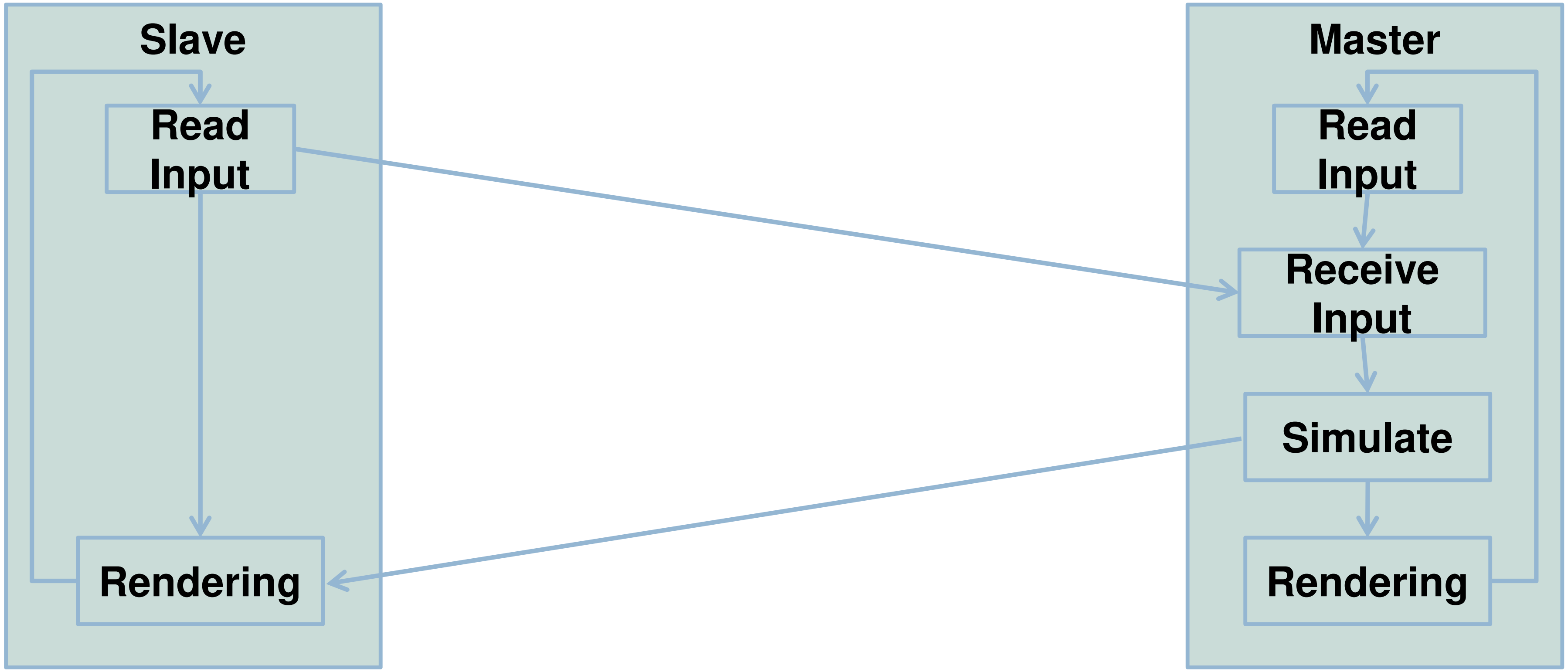
For Example DOOM



Master Slave with Two Clients

- One process calculates results of input
- Necessary if simulation is non-deterministic

For Example

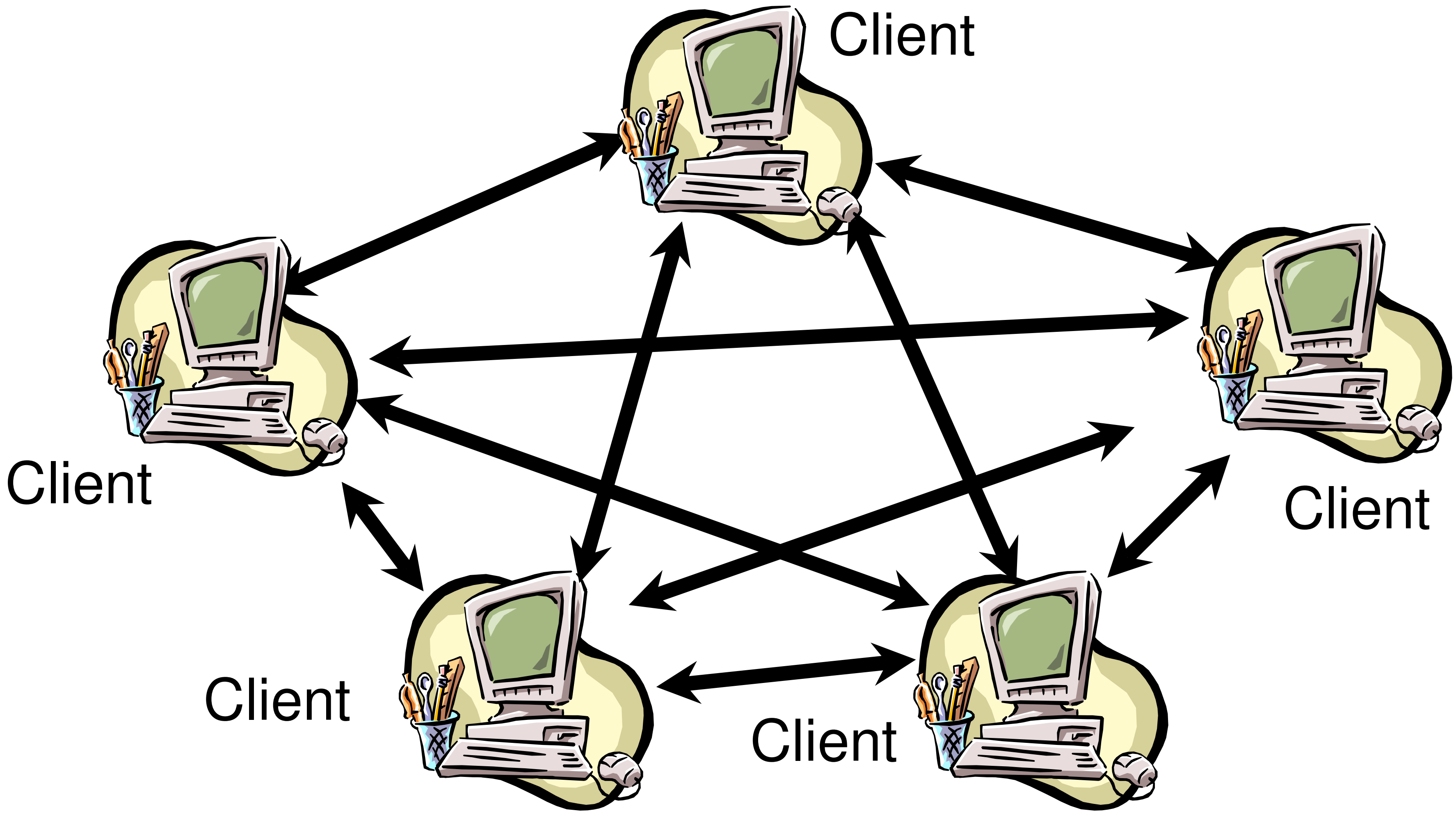


Slave
Read Input
Rendering
Master
Read Input
Receive Input
Simulate
Rendering

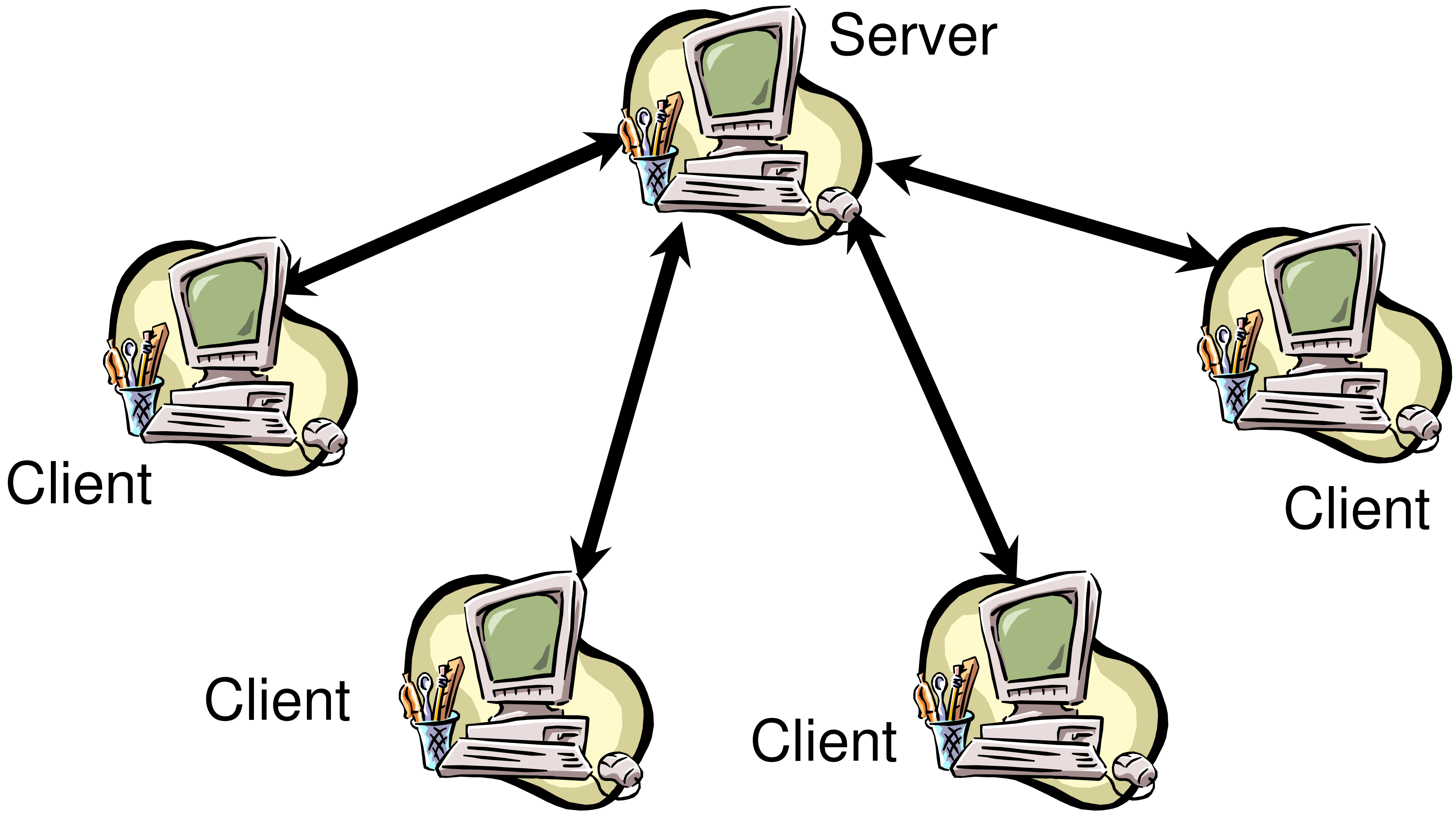
More Clients

- The same issues exist:
  - Who is responsible?
  - Who has the necessary data to evolve the state?
  - Who can be **trusted** to evolve the state?

Peer to Peer Architecture



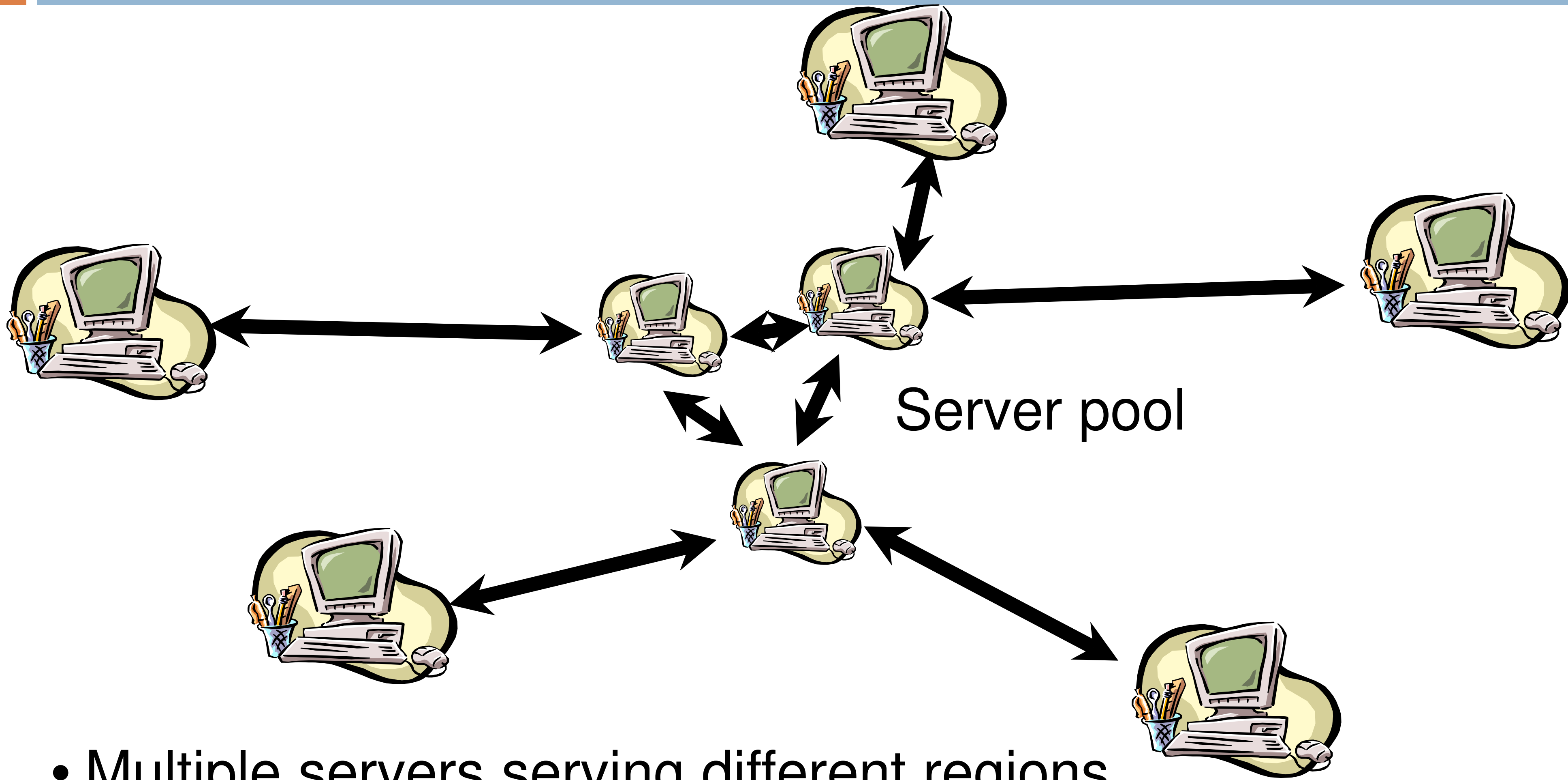
Client-Server Architecture



Implications

- Peer to Peer
  - Data need to be sent multiple times on the network links might vary in bandwidth & latency
  - Clients need to manage multiple connections
- Client Server
  - The Server is a bottleneck
  - Clients manage one connection
  - Server can have privileged data, and can probably be trusted
  - Latency is higher
  - Synchronization is easy

Hybrid Architectures



- Multiple servers serving different regions
- Multiple service types & service layers

Which Protocol to Use?

- If there is an application layer protocol that is appropriate use that!
- UDP
  - Good for fast changing data, and initial start update
  - Good for position information
- TCP
  - Good for reliable data, and bulk data transfer
  - Good for data assets and critical information such as score

Which Protocol to Use?

- Some people implement “reliability-lite” on top of UDP
- Other platforms mix UDP & TCP
  - There are many catches with this
- Many platforms support application layer protocols such as HTTP or FTP for bulk asset transfer

SUMMARY

- NVEs & NGs have a long history, but it is in the last 10 years that they have really taken off
- The Internet is a *best effort* network where applications need to deal with latency & loss
- There are various architectures that support NVEs & NGs

- In Part 2 we will look in detail at NVEs and NGs requirements, and compare this to what we actually get from the Internet today