# CERN openlab Summer 2006: Networking Overview

Martin Swany, Ph.D.

Assistant Professor, Computer and Information Sciences, U. Delaware, USA

Visiting Helsinki Institute of Physics (HIP) at CERN

swany@cis.udel.edu, Martin.Swany@cern.ch

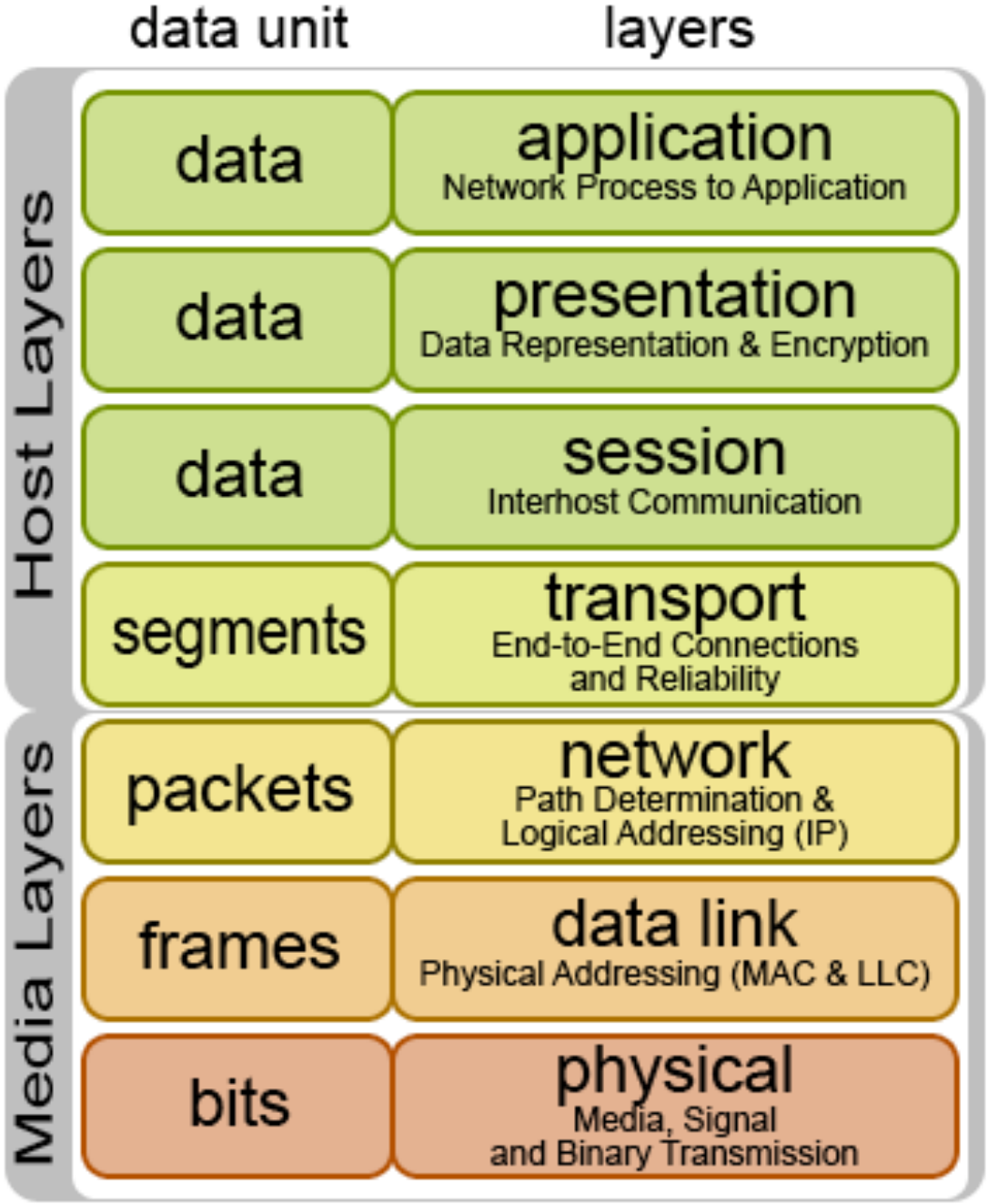
# Overview

- Introduction to the OSI Model
- Overview of the Internet Protocols
- Network Performance
- Advanced Interconnects

# The OSI Model

- Open Systems Interconnection (OSI)
  - Model and protocols defined by the ISO
- Framework and protocols developed to allow different networks to communicate
  - The protocols have all but died, but the model is widely referenced
- Each layer provides well-defined interface to the layer above
  - And each layer uses only the services of the layer below
- Each layer adds a header
  - some also a trailer

# OSI Model


data unit
layers
Host Layers
Media Layers
data
application
Network Process to Application
data
presentation
Data Representation & Encryption
data
session
Interhost Communication
segments
transport
End-to-End Connections and Reliability
packets
network
Path Determination & Logical Addressing (IP)
frames
data link
Physical Addressing (MAC & LLC)
bits
physical
Media, Signal and Binary Transmission

# OSI Layers

- Physical Layer
  - Concerned with transmission of bits and bytes
  - Standards for electrical, mechanical and signaling interfaces
    - What do bits and bytes look like “on the wire”
- Link Layer
  - Groups bits and bytes into frames and ensures correct delivery
  - Handles errors in physical layer
  - Adds bits (head/tail) + checksum (receiver verifies checksum)
  - Sublayers: LLC – Logical Link Control and MAC – Medium Access Control

# OSI Layers

- Network Layer
  - This is the “Packet” layer
  - Transmission and addressing of packets
  - Chooses the best path for the packet (routing)
    - Each packet gets routed independently to its destination
  - Connectionless
  - Unreliable, best effort service
- Internet Protocol - IP
  - Currently, most hosts are using Version 4, which features 32 bits for addresses
  - Version 6 is coming “soon” and features 128 bit addresses
  - The netmask is a string of bits which are “and”ed with the address to determine the network

# OSI Layers

- Transport Layer
  - The “end to end” layer
  - UDP - User Datagram Protocol
    - Simple addressing (port number) for direct use of datagrams
  - TCP - Transport Control Protocol
    - Ensures reliable service (network layer does not deal with lost messages)
    - Breaks massage into segments, assigns a sequence number and sends them
    - Builds reliable network connection on top of IP (or other protocols)
  - SCTP - Stream Control Transport Protocol
    - Manages multiple streams of communication within a single association
    - Also has provisions for “partial reliability”

# OSI Layers

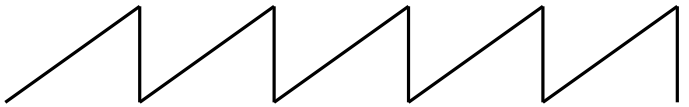
- Session Layer
  - Establishes, maintains and terminates sessions across networks
  - Session Initiation Protocol for VoIP, etc.
  - Some consider remote login initiation or TCP handshake to be instances
- Presentation Layer
  - Translates application → network format (big endian)
  - Can potentially include De-/Encryption, Compression...
- Application Layer
  - DNS, FTP, SMTP, NFS, …

# Network Protocols - QoS

- Each IP packet has bits for identifying a Type of Service (ToS)
  - When used by Differentiated Services (DiffServ) they are called DiffServ Code Points (DSCP)
- These bits can be used to affect ingress and egress disciplines on routers
  - Queues of different priorities allow bandwidth reservation
- These mechanisms can provide improved Quality of Service (QoS) for certain applications
  - Can also be based on source and destination IP addresses

# TCP Details

- TCP provides reliable transmission of byte streams over best-effort packet networks
  - Sequence number to identify stream position inside segments
  - Segments are buffered until acknowledged
  - Congestion (sender) and flow control (receiver) “windows”
  - Everyone obeys the same rules to promote stability, fairness, and friendliness
- Congestion-control loop uses ACKs to clock segment transmission
  - Round Trip Time (RTT) critical to responsiveness
- Conservative congestion windows
  - Start with window O(1) and grow exponentially then linearly
  - Additive increase, multiplicative decrease (AIMD) congestion window based on loss inference
  - “Sawtooth” steady-state



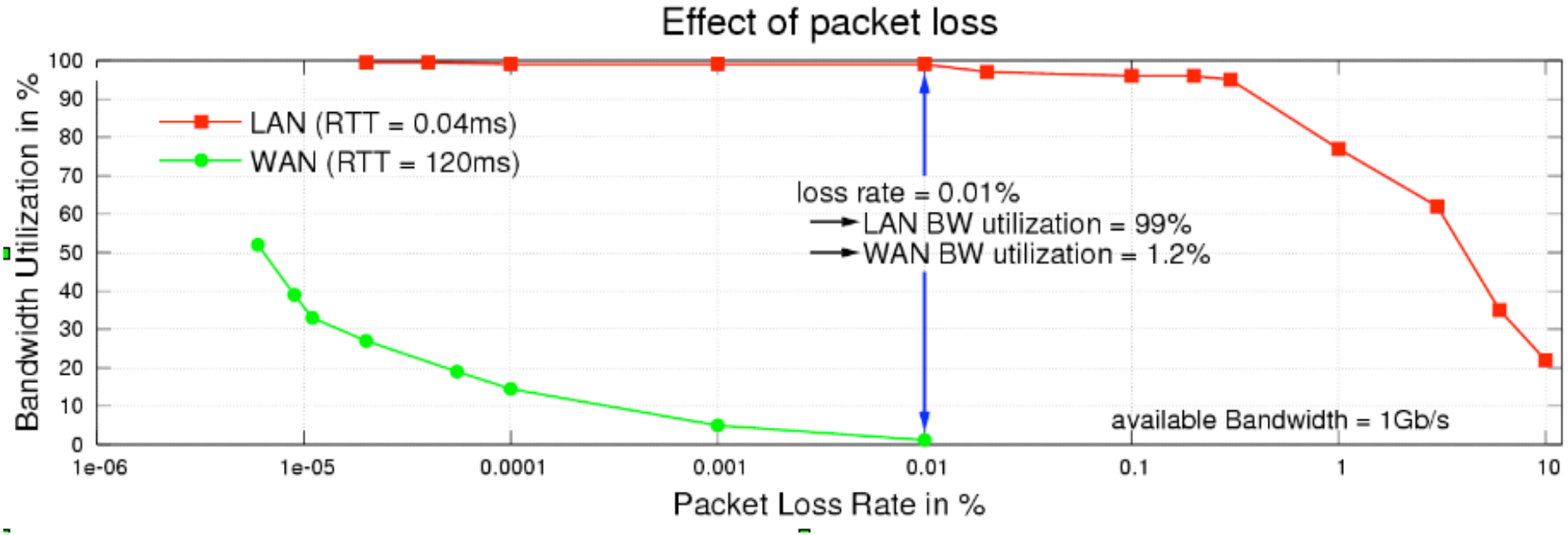
$$BW = \frac{mss}{rtt * \sqrt{loss}} * C$$

# TCP Issues

- TCP has issues with high bandwidth-delay product networks
- TCP must buffer data in the kernel until it has been acknowledged
  - Standard TCP Window (*nix): 32kBytes - 256kBytes
  - 10Gb and 100ms delay: min. TCP window ~= 128 Mbytes
- Other issues due to the AIMD behavior of TCP congestion control
  - Packet loss reduces the window
  - Many packets must be sent and ACKed to increase it

# Network Performance

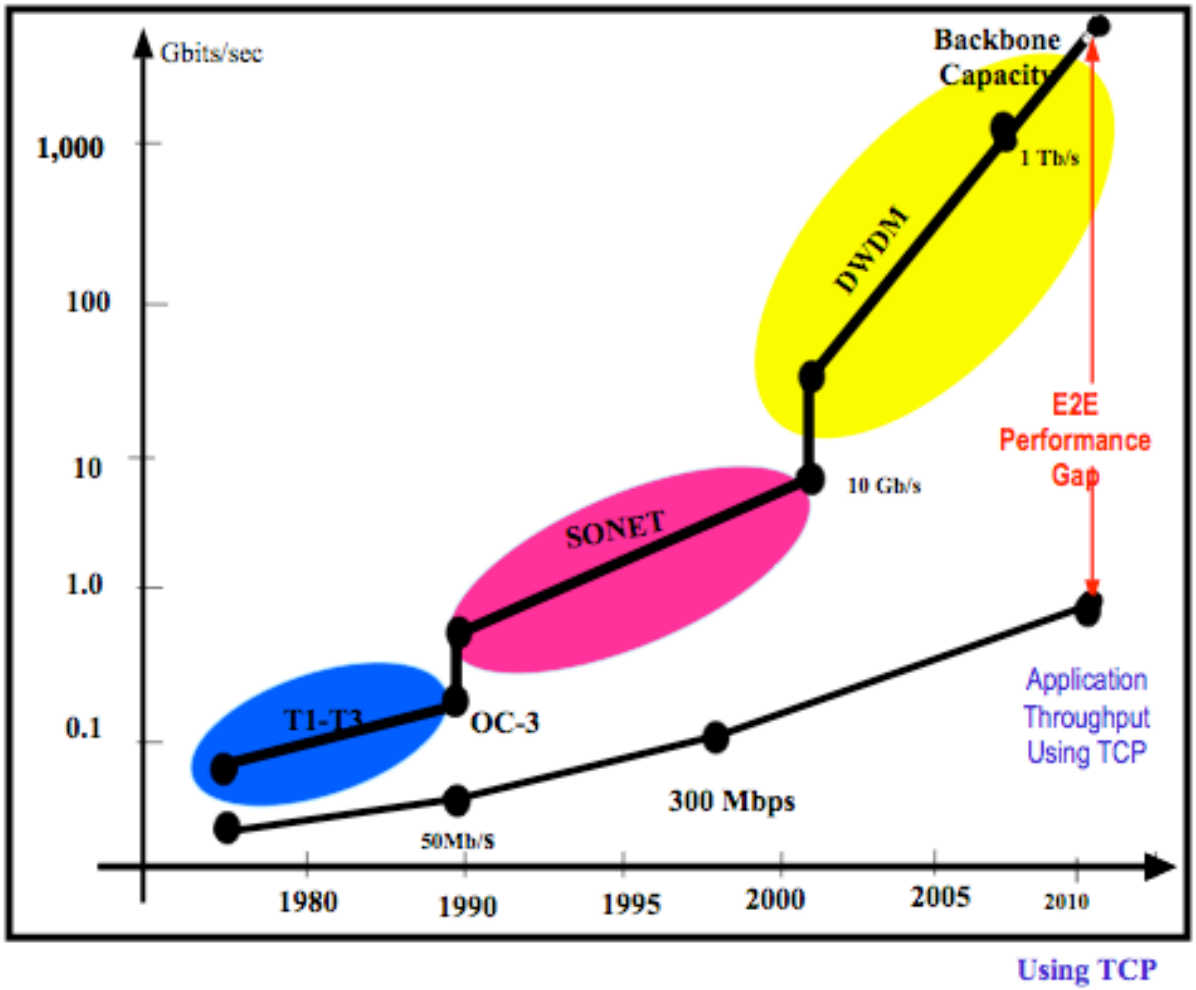
$$BW = \frac{mss}{rtt * \sqrt{loss}} * C$$



# TCP Issues - Data Link Packet Sizes

- TCP is sensitive to the Maximum Segment Size (MSS)
- The MSS should fit inside the Maximum Transfer Unit (MTU)
  - The maximum size of a non-fragmented IP packet
- The IP MTU depends on the link MTU
- Standard Ethernet - 1500 bytes
- High end equipment supports up to 9216 byte
  - (Intel 10Gb NICs support 16114 byte MTU !!)
- Having end to end support for large MTUs is challenging
  - Coordination of all parties
  - Expense of equipment

# Network Performance




– Network speeds can increase dramatically but users' throughput increases much more slowly
  - Source: US Department of Energy

# Network Performance Techniques

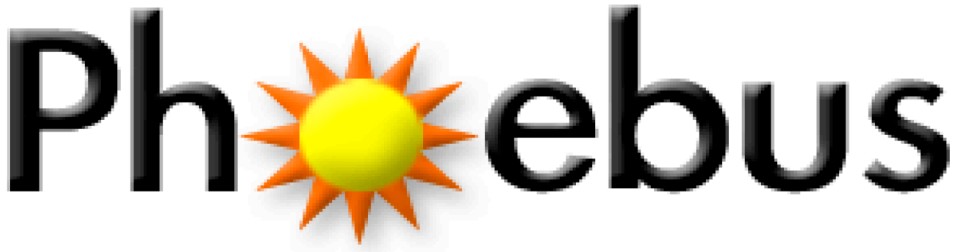
- Transport signaling for high bandwidth-delay networks
- Have the host do less work for high bandwidth interfaces
  - Reduce the number of interrupts from the network interface card
    - Interrupt coalescing (more than one frame before interrupting)
    - Larger MTUs also help with this
  - Reduce copying of data in the host

# Transport Signaling

- Research groups have produced various TCP variants to address the problems with modern networks
  - BIC, CUBIC, FAST, HS-TCP, HTCP
- Essentially trying to find a control strategy that is fair, yet can take advantage of available bandwidth
- Recent work *seems* to indicate that all have shortcomings under some conditions

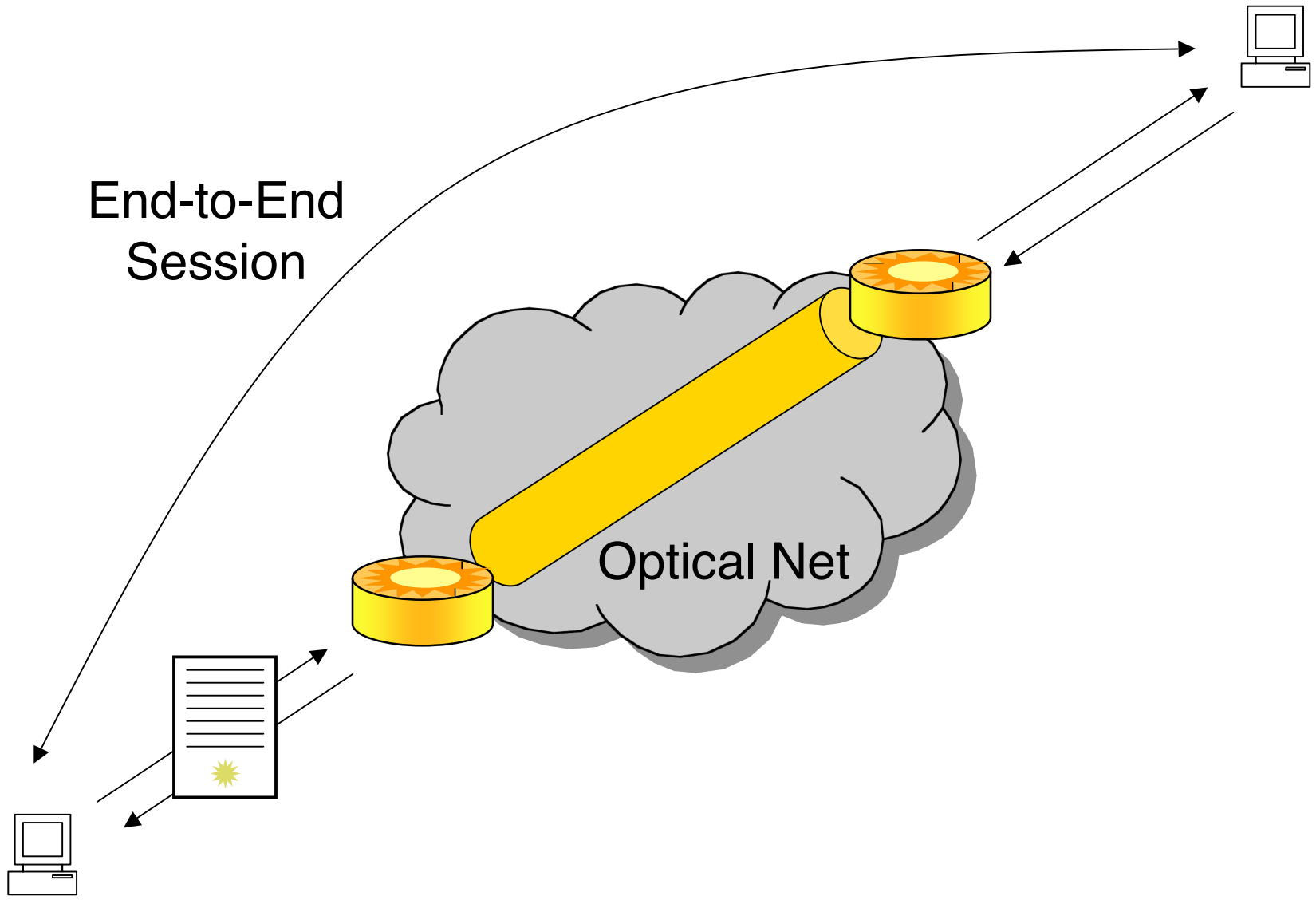
# Optical Networks

- Bandwidth of networks continues to increase
- One interesting development is wave-division multiplexing (WDM)
- This allows for “parallel” circuits within a single fiber
- Dramatic increase in bandwidth, if we could only use it effectively
- One solution is to allow demanding applications to allocate bandwidth on demand

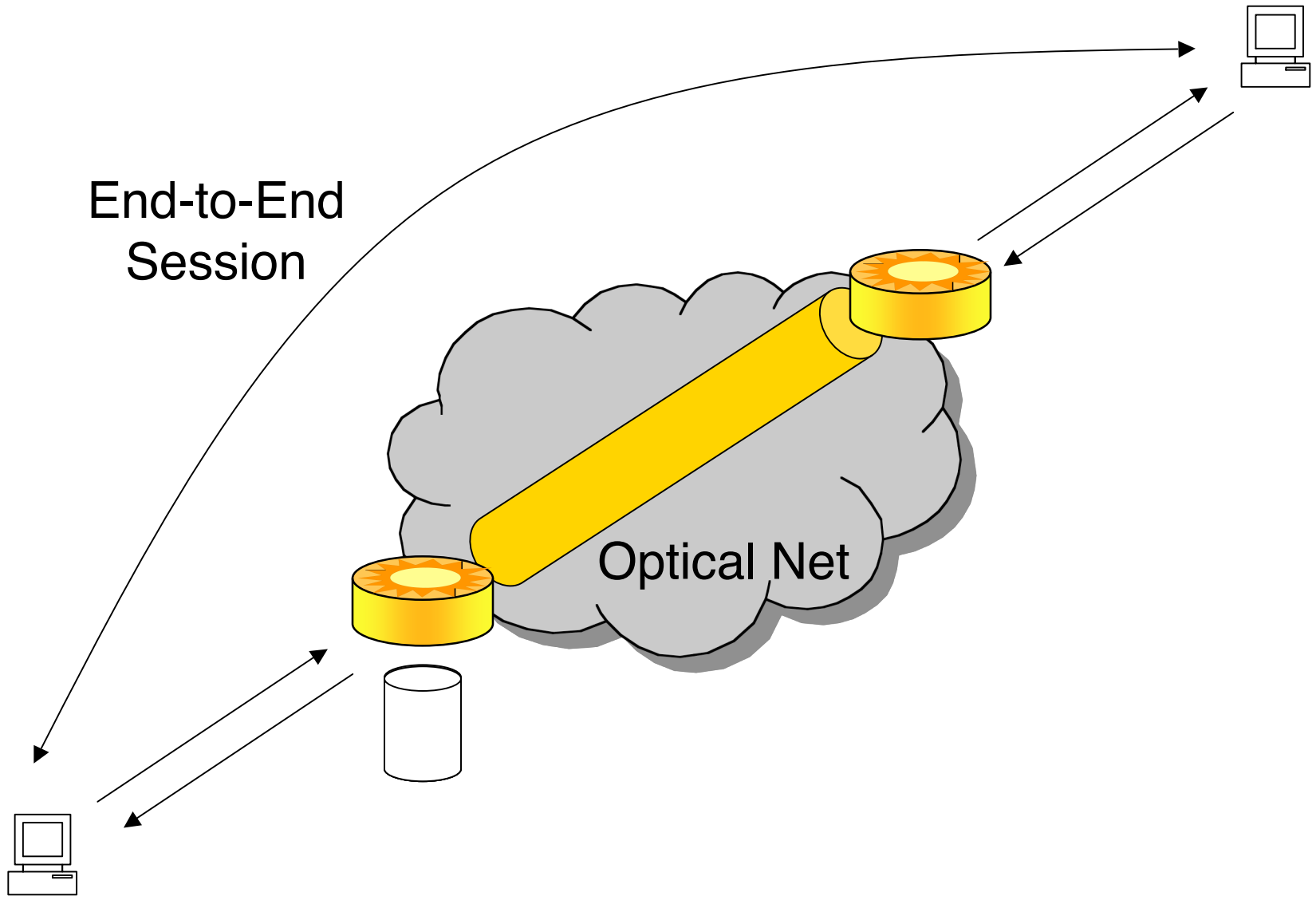



- Phoebus is a project in my group that is targeted at optical networks
  - Based on previous work called the Logistical Session Layer (LSL)
- Service Nodes provide short-term storage and cooperative data forwarding
- Provide adaptation points for segment-specific transport protocols
- Also can provide *improved throughput* for reliable data streams

# Phoebus

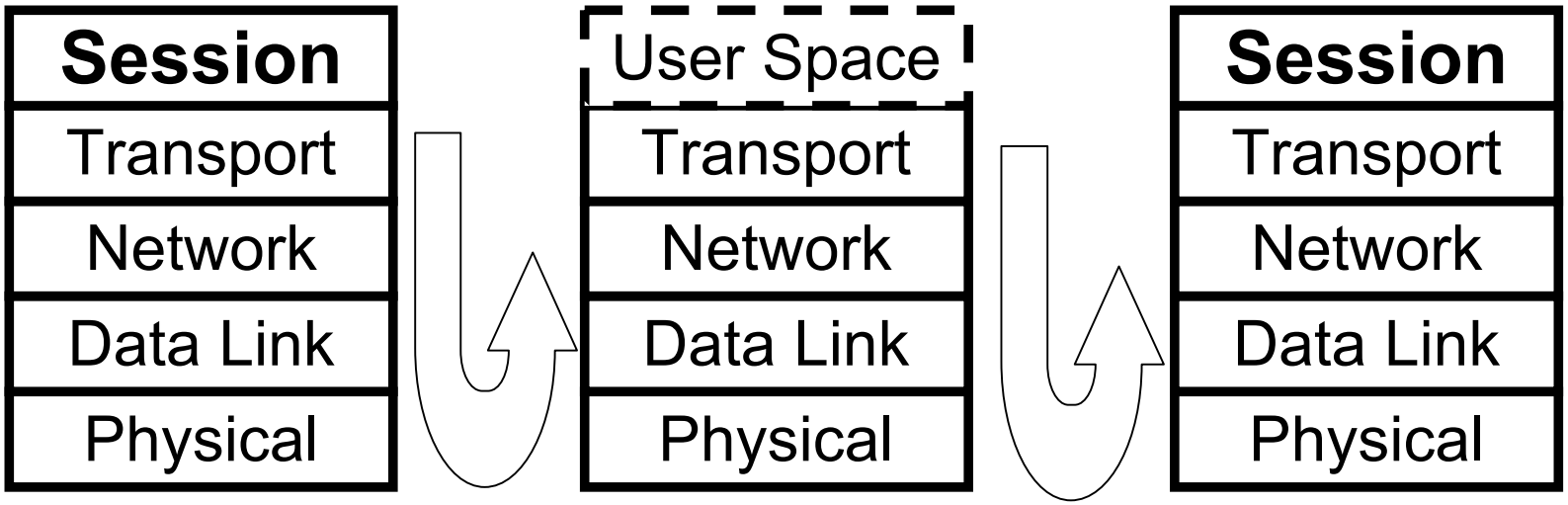


# Phoebus

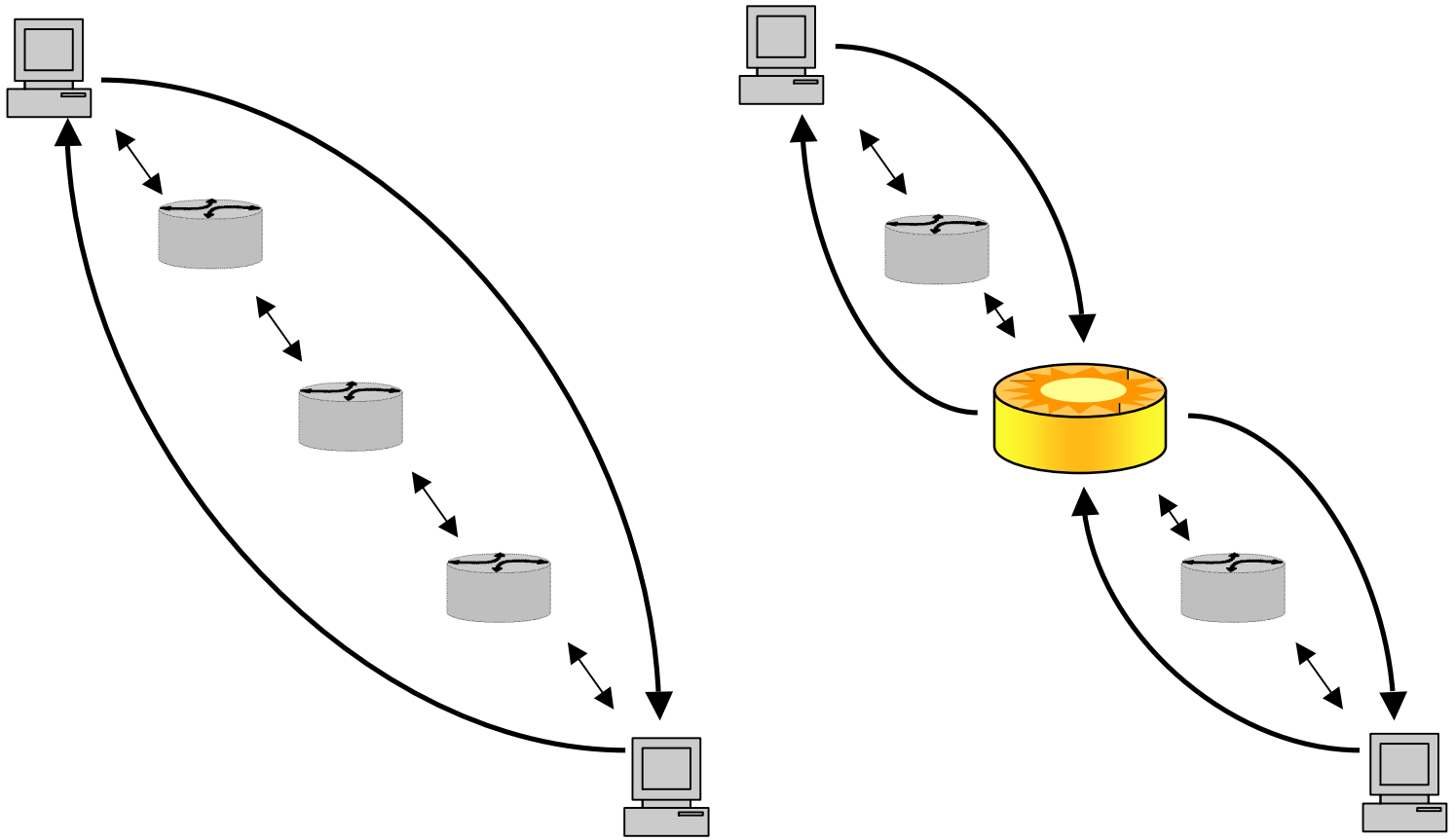


# Session Layer

- A *session* is the end-to-end composition of *segment-specific* transports and signaling
  - More responsive control loop via reduction of signaling latency
  - Adapt to local conditions with greater specificity
  - Buffering in the network means retransmissions need not come from the source

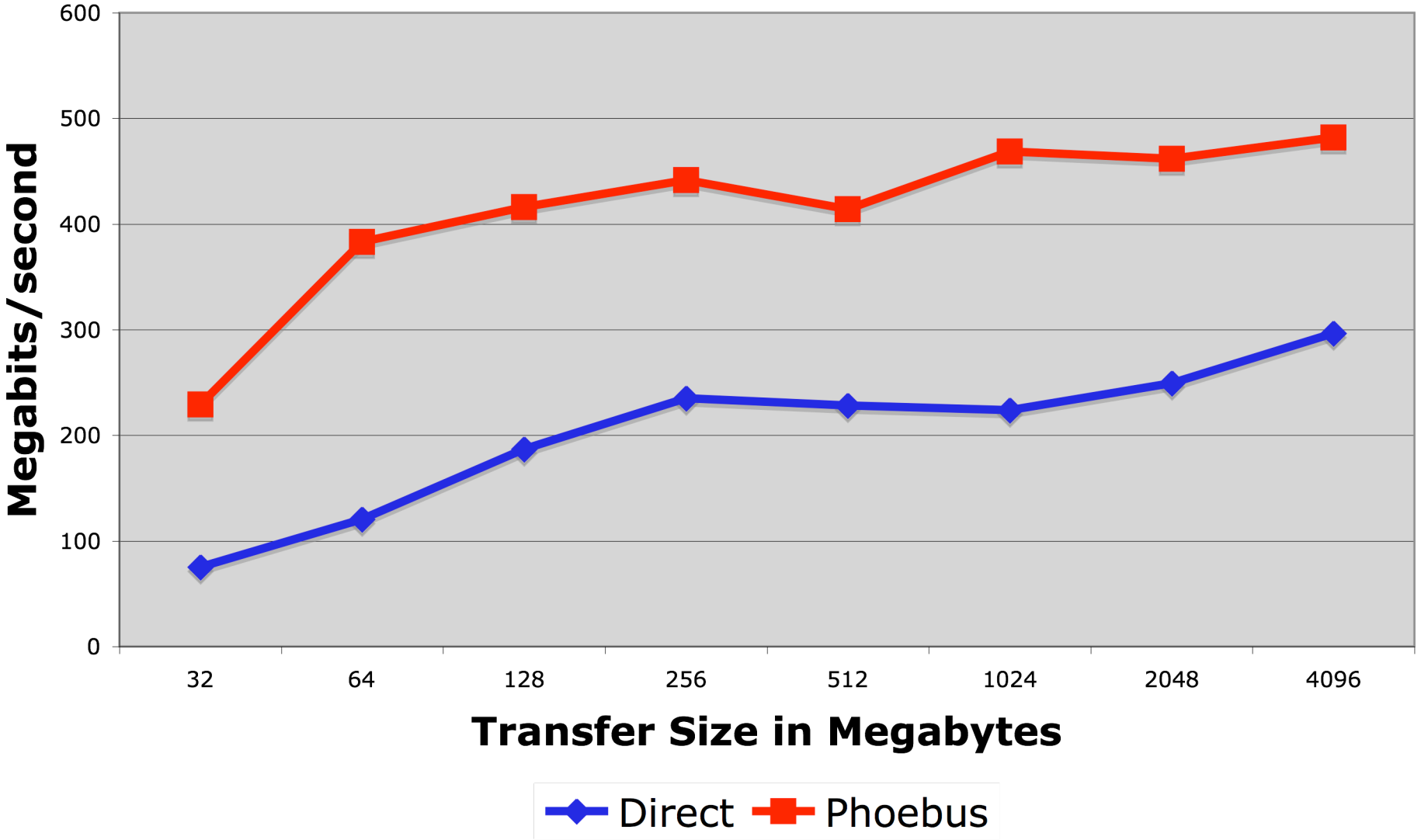


# The Phoebus Session Protocol



# Phoebus Performance

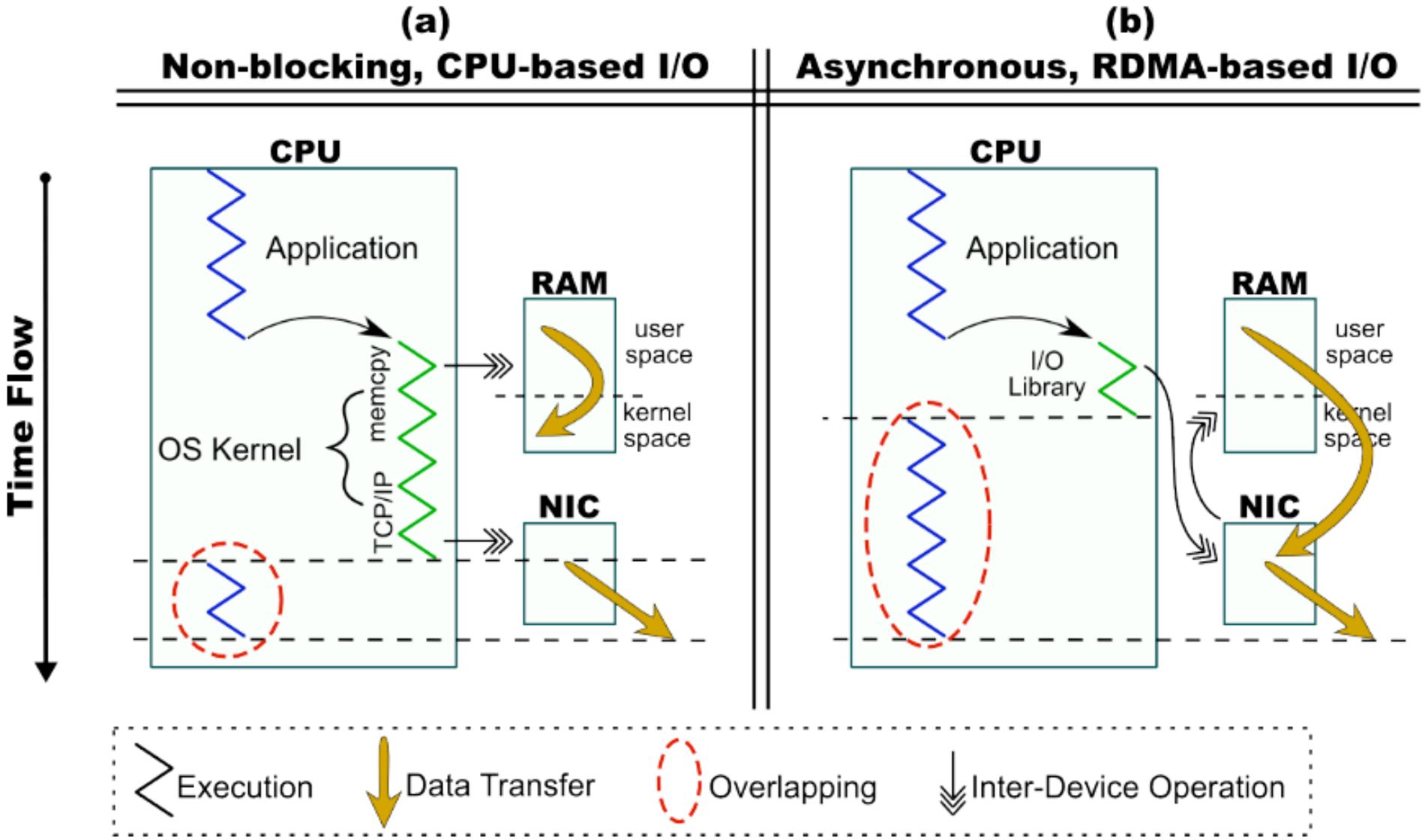
## Bandwidth Comparison



# The End to End Arguments

- Why aren't relays like this already in use?
- Recall the "End-to-End Arguments"
  - E2E Integrity
    - Network elements can't be trusted
    - Duplication of function is inefficient
  - Fate sharing
    - State in the network related to a user
  - Scalability
- Network transparency
- Network opacity
- The original assumptions regarding network scalability and complexity may not hold true any longer

# OS Bypass


(a)
Non-blocking, CPU-based I/O
(b)
Asynchronous, RDMA-based I/O
Time Flow
CPU
Application
OS Kernel
memcpy
TCP/IP
RAM
user space
kernel space
NIC
CPU
Application
I/O Library
RAM
user space
kernel space
NIC
Execution
Data Transfer
Overlapping
Inter-Device Operation

# Advanced Interconnects

- Feature rich (R)DMA capability
  - Some also offload functionality from the system although the right amount is a matter of some debate
- Examples include Myrinet, Quadrics, Infiniband
- In a fairly recent development, the OpenIB effort has merged with iWARP effort and generalized into OpenFabrics
  - This could provide a universal API acessing advanced networks

# Advanced Interconnects

- Programming is an open question
  - Early work such as Unet identified basic primitives for OS-bypass functionality
- Virtual Interface Architecture (VIA) standardized by Microsoft, Intel and Compaq
- Memory must be registered
  - Either directly or indirectly
- Non-blocking calls to “post” sends and check for completion
- In comparison to the Internet architecture, there is very little in the way of a Transport layer

# OpenFabrics - Verbs

- Very much like the VIA architecture
- Based on the VAPI originally developed by Mellanox
- Handles memory registration and functions for managing send and receive descriptors
- Kernel component and user component

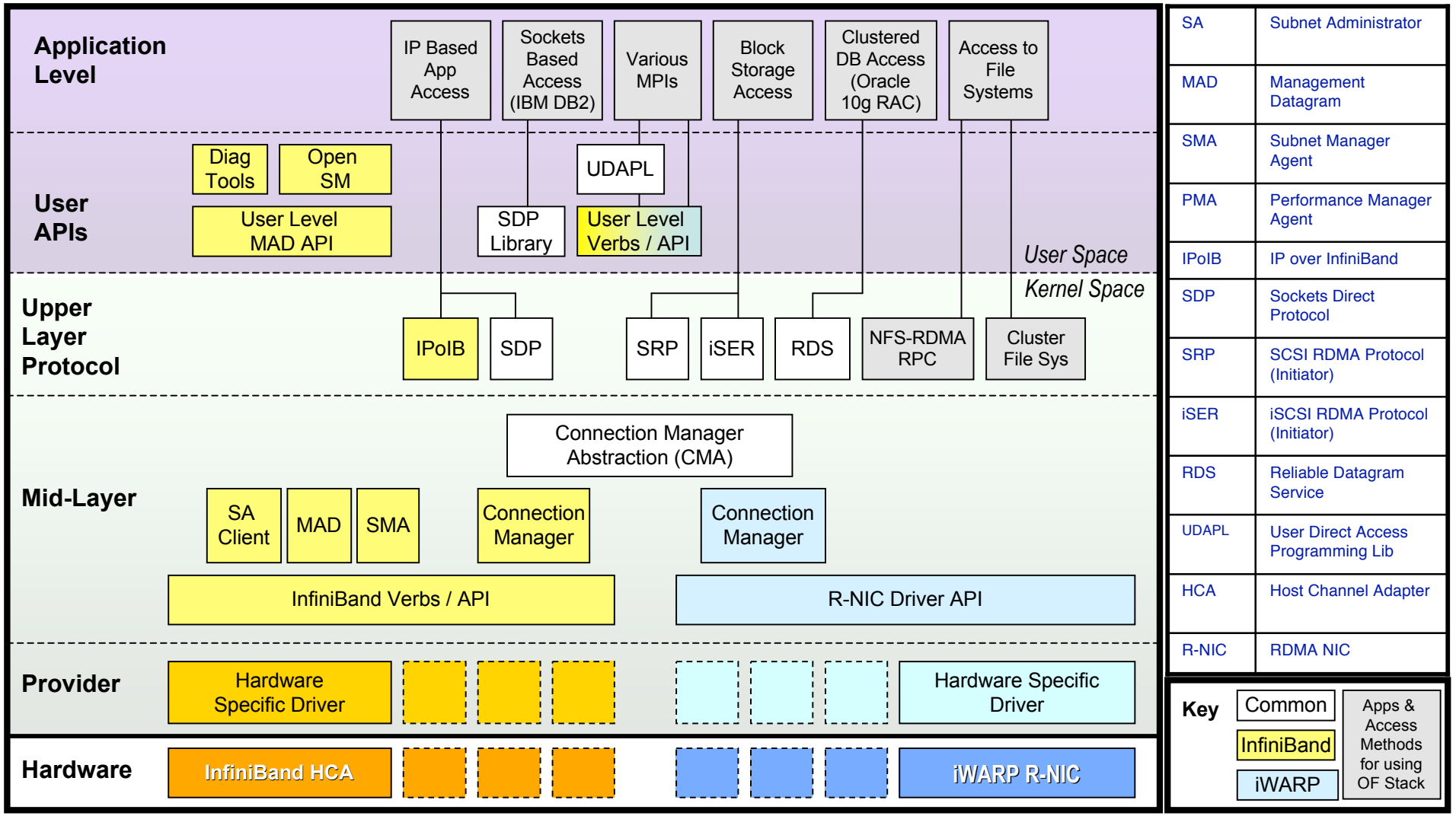
# OpenFabrics - DAPL

- Direct Access Provider Library
- From the Direct Access Transport (DAT) Collaborative
- Provides explicit support for RDMA
  - Remote memory descriptor
- Also a user and kernel component
  - uDAPL and kDAPL

# OpenFabrics - SDP

- Uses the standard sockets interface in an attempt to make it easier for applications to take advantage of fast networks
- Good performance in many cases

# OpenFabrics Software Stack




| | |
|---|---|
| SA | Subnet Administrator |
| MAD | Management Datagram |
| SMA | Subnet Manager Agent |
| PMA | Performance Manager Agent |
| IPoIB | IP over InfiniBand |
| SDP | Sockets Direct Protocol |
| SRP | SCSI RDMA Protocol (Initiator) |
| iSER | iSCSI RDMA Protocol (Initiator) |
| RDS | Reliable Datagram Service |
| UDAPL | User Direct Access Programming Lib |
| HCA | Host Channel Adapter |
| R-NIC | RDMA NIC |

**Key**

- Common
- InfiniBand
- iWARP

Apps & Access Methods for using OF Stack

# end

- Questions?