The Logistical Session Layer

Martin Swany

Network Logistics

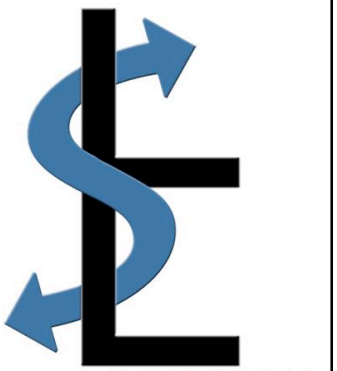
- The definition of Logistics

  "…the process of planning, implementing, and controlling the efficient, effective flow and storage of goods, services and related information from point of origin to point of consumption."

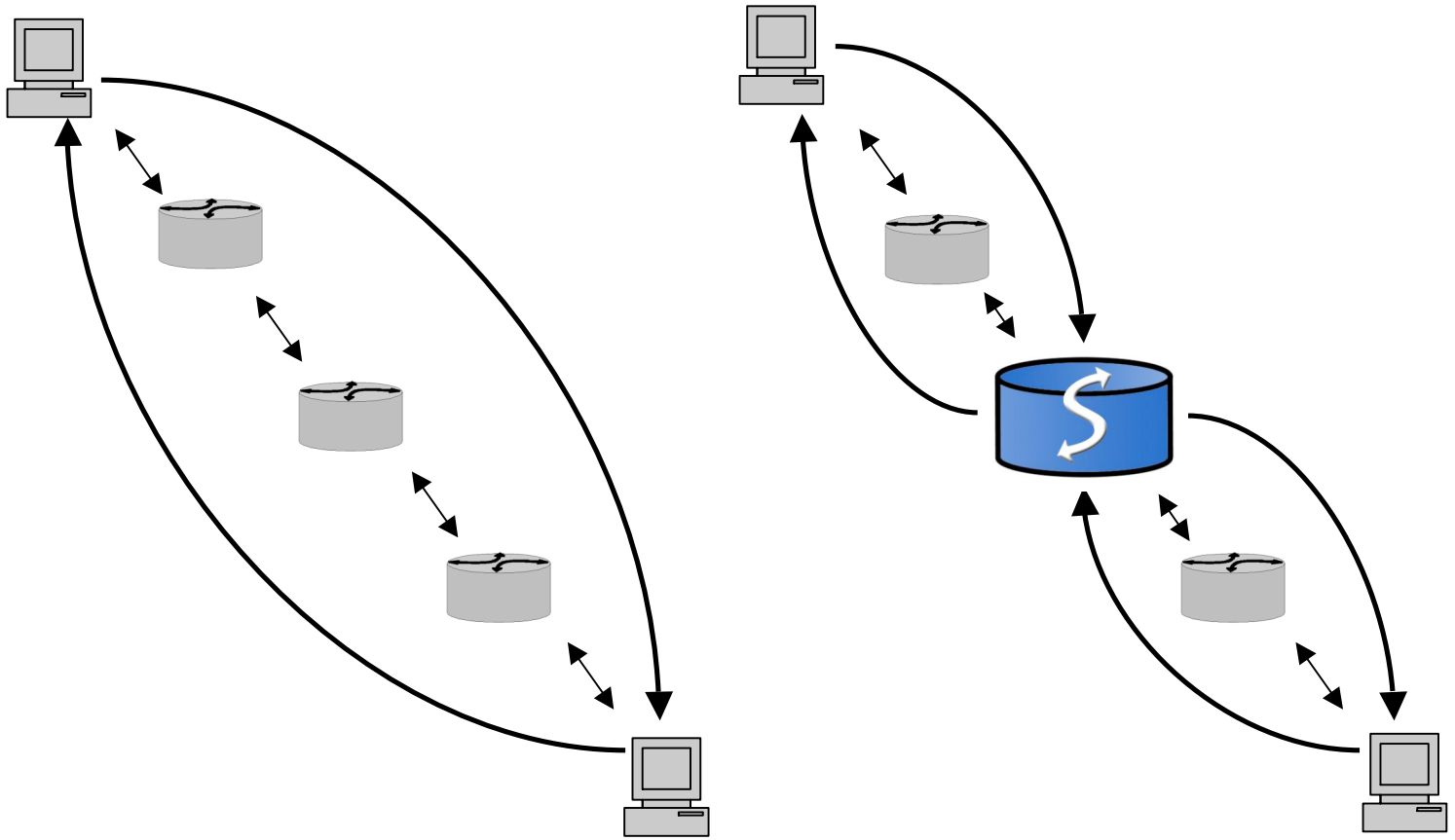
- Shipping and distribution enterprises make use of storage (and transformation) when moving material
- Optimizing the flow, storage, and access of data is necessary to make distributed computing environments viable

The Logistical Session Layer

- LSL allows systems to exploit “logistics” in stream-oriented communication
- LSL Service Nodes (*depots*) provide short-term logistical storage and cooperative data forwarding
- The primary focus is *improved throughput* for reliable data streams
  - Both unicast and multicast
- A wide range of new functionality is possible

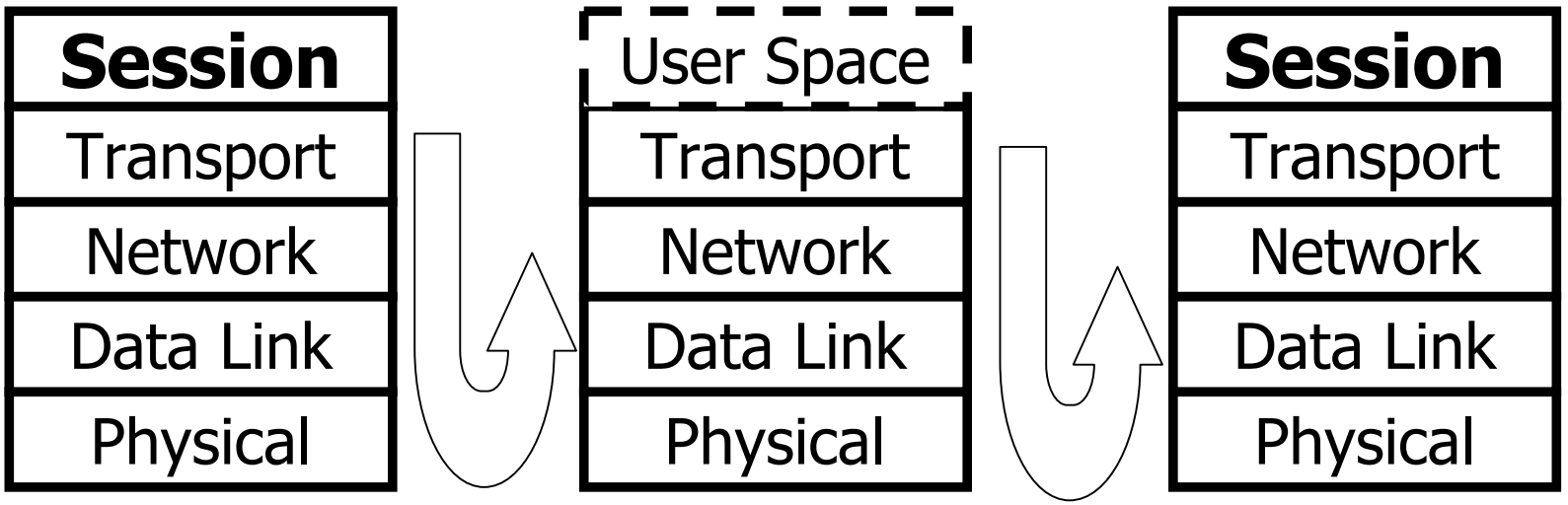


The Logistical Session Layer



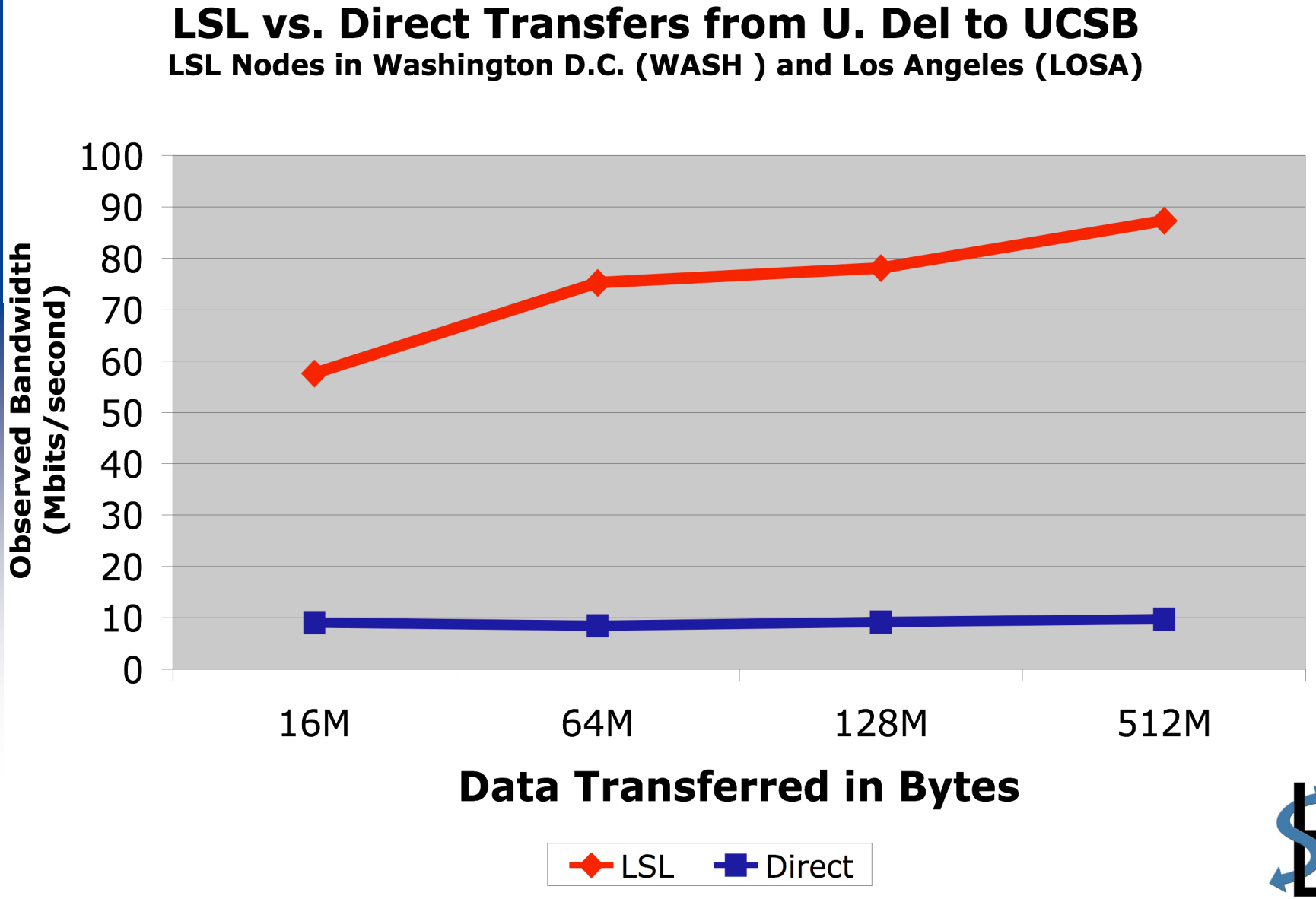
Session Layer

- 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



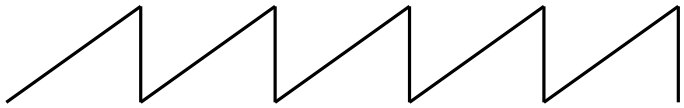
US Test Deployment

LSL Performance Improvement



TCP Overview

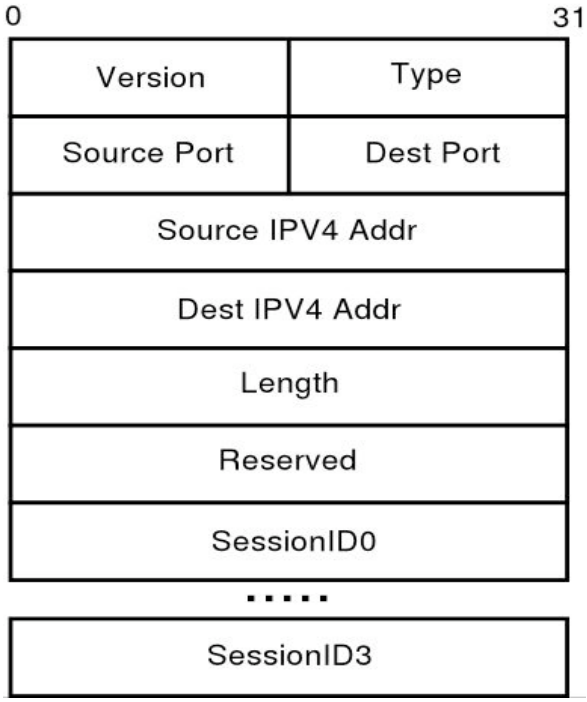
- TCP provides reliable transmission of byte streams over best-effort packet networks
  - Sequence number to identify stream position inside segments
  - Segments are buffered until acknowledged
  - Congestion (sender) and flow control (receiver) “windows”
  - Everyone obeys the same rules to promote 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
  - Problems with high bandwidth delay product networks



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

Protocol Architecture

- Logistical Service Nodes (LSNs) at strategic locations in the network
- LSNs provide a data store
- LSNs provide a locus of control
- Session-layer protocol establishes transport-layer connections



The End to End Arguments

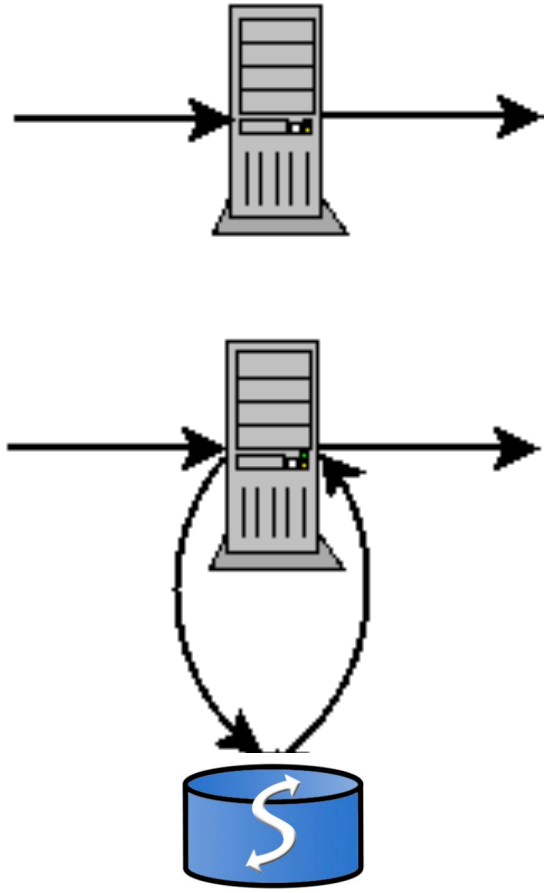
- Why aren't techniques like this already in use?
- Recall the end-to-end arguments
  - E2E Integrity
    - Network elements can't be trusted
    - Duplication of function is inefficient
  - Fate sharing
    - State in the network related to a user
  - Scalability
- Network transparency
- Network opacity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

LSL Implementation

- The LSL client library provides compatibility with current socket applications
  - Although more functionality is available using the API directly
- LD_PRELOAD for function override
  - `socket()`, `bind()`, `connect()`, `setsockopt()`…
  - Allows Un*x binaries to use LSL without recompilation
- Daemon runs on all Un*x platform
  - Forwarding is better on Linux than on BSD

Transparent Intercept

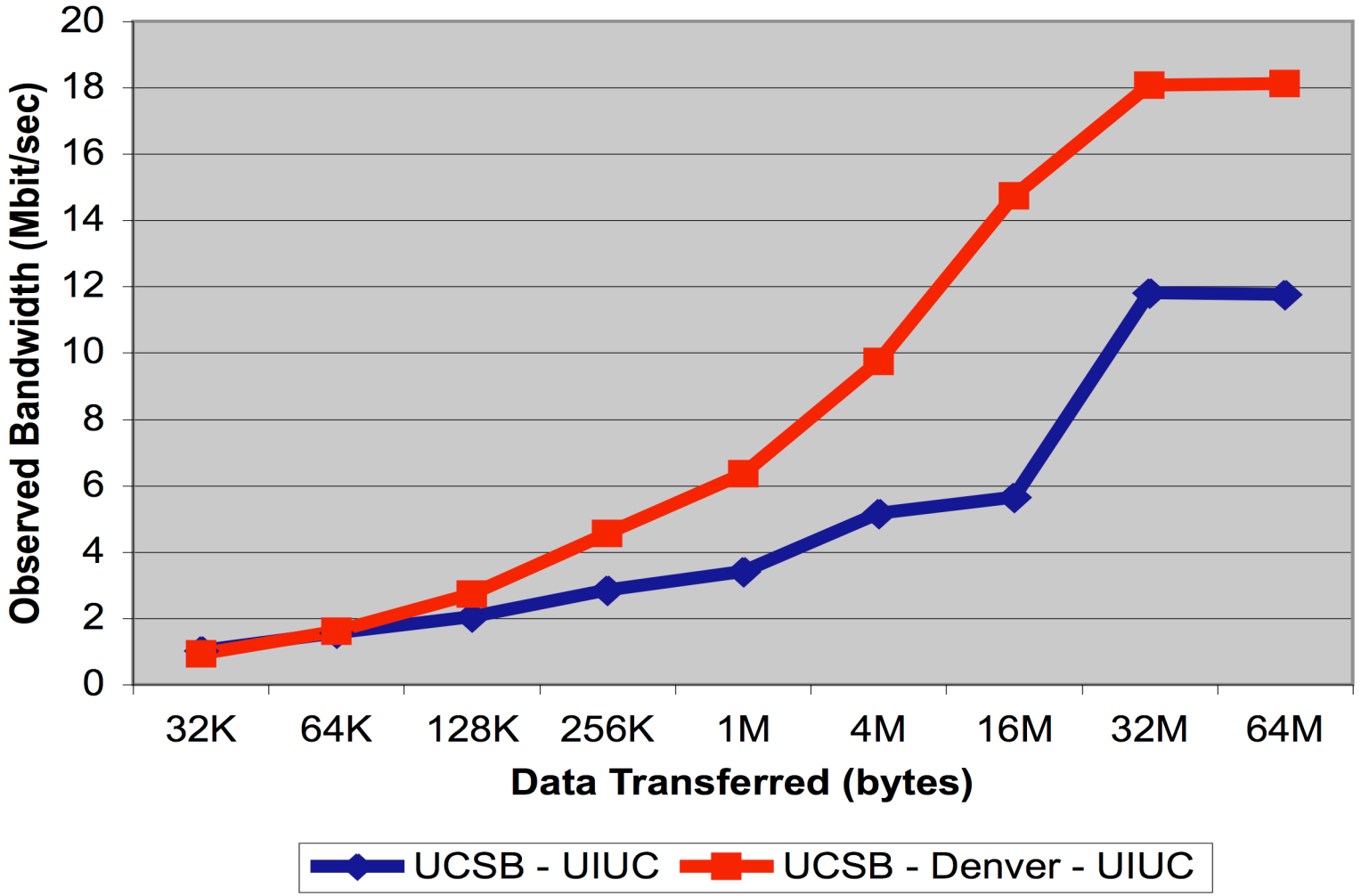
- Intercept the TCP SYN with IP Tables (on Linux)
- Redirect to local LSL process
- Establish connection with appropriate LSNs or end node
  - Based on policy
- Transparent to end hosts



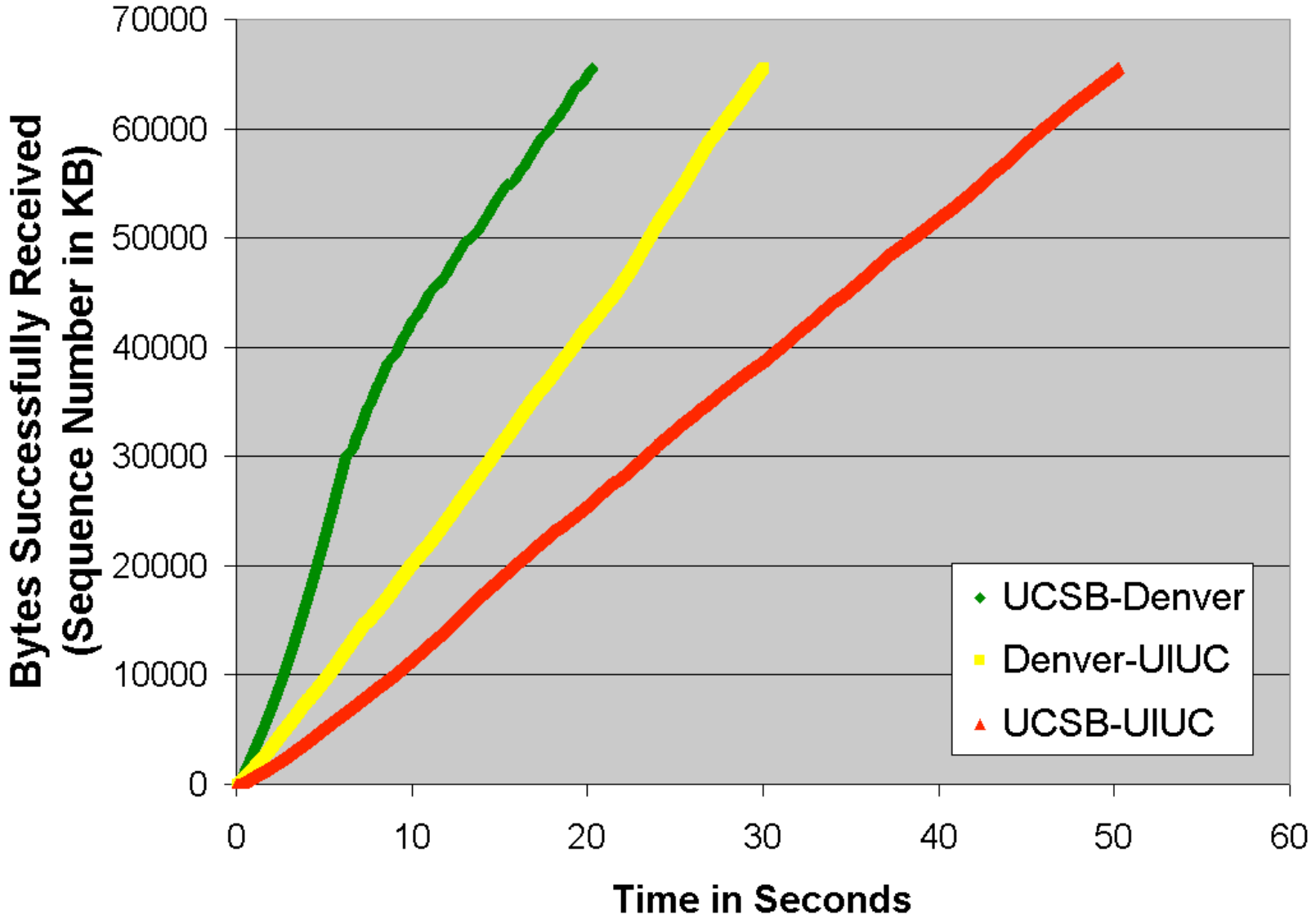
Cascaded TCP Dynamics

- Recall TCP's sequence number and ACKs
  - We can observe the progress of a TCP connection by plotting the sequence number acknowledged by the receiver
- For this experiment, we captured packet-level traces of both LSL and end-to-end connections
  - 10 traces for each path and subpath were gathered
  - We compute the average growth of the sequence number with respect to time
- The following graphs depict average progress of a set of transfers

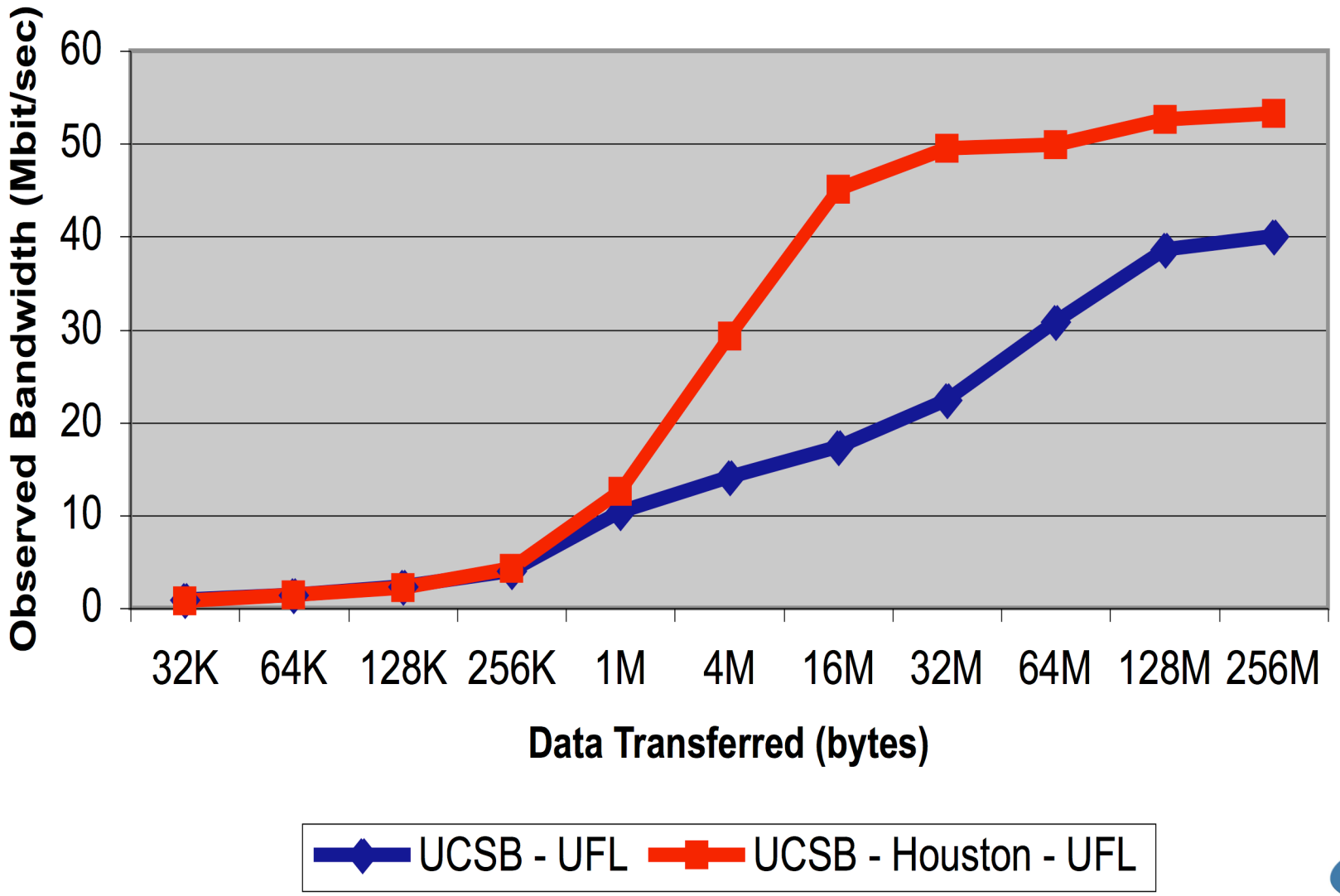
LSL Speedup - UCSB/UIUC



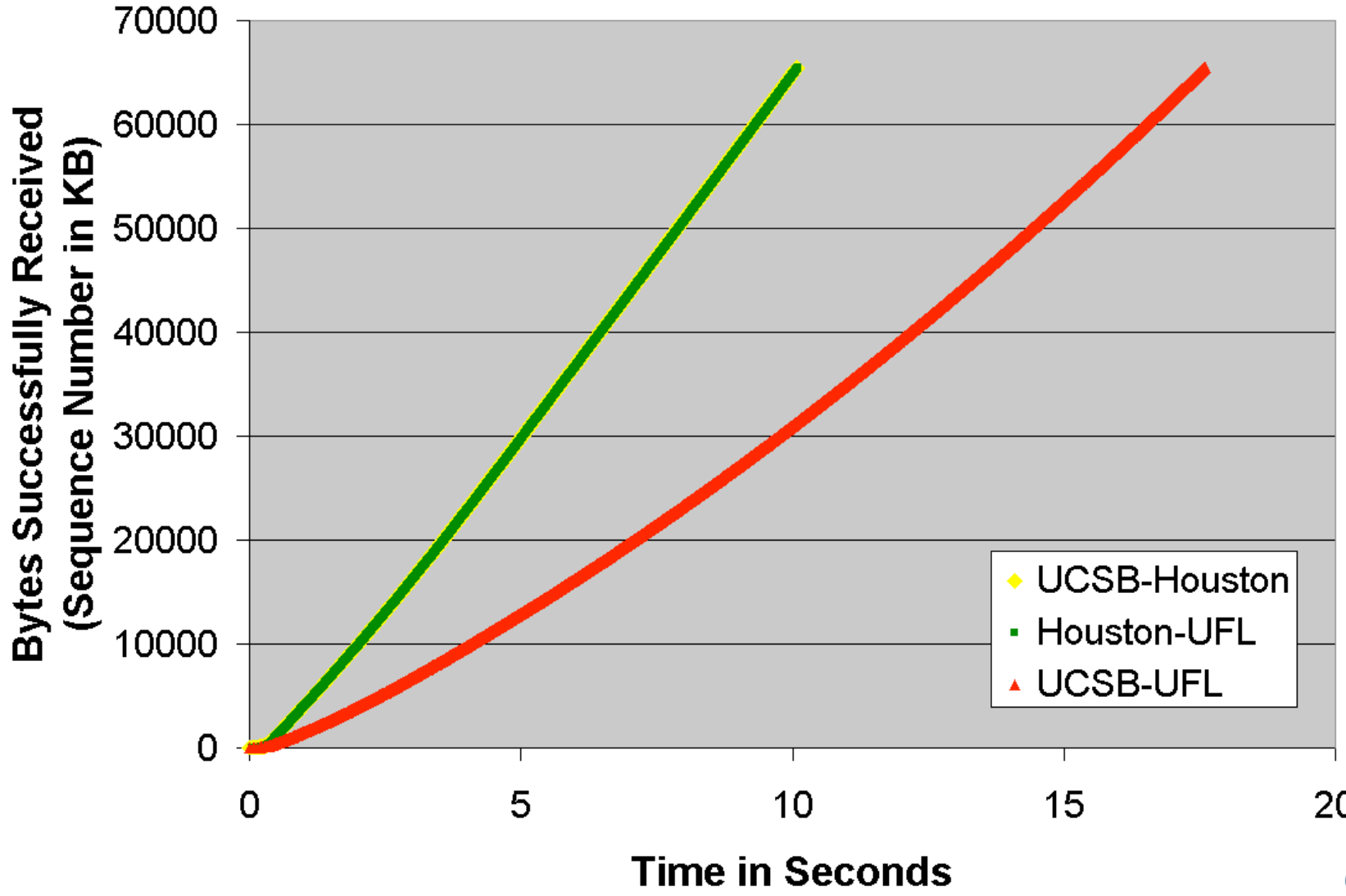
UCSB->Denver->UIUC (64M)



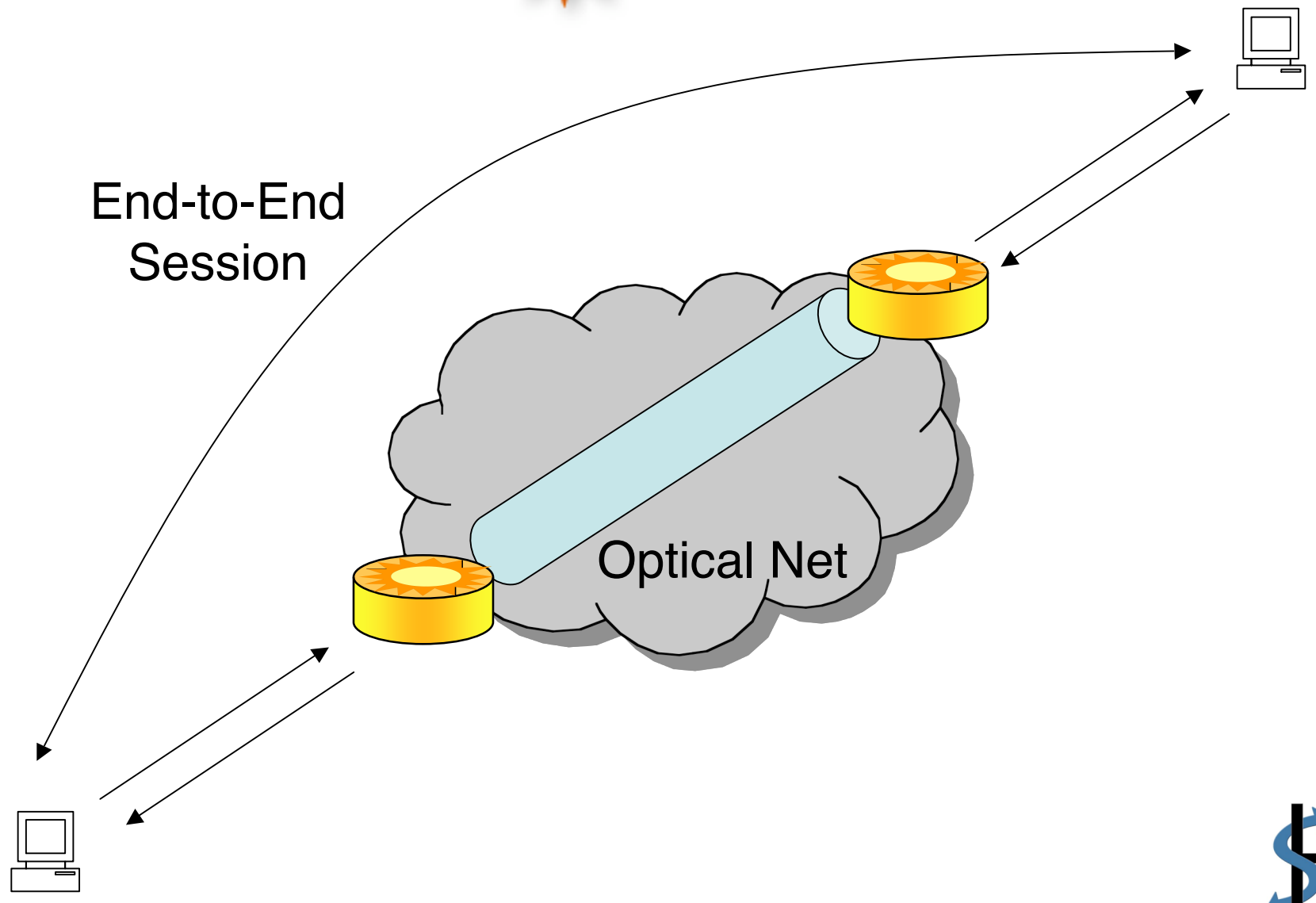
LSL Speedup - UCSB/UF



UCSB->Houston->UFL (64M)



Phoebus



Benefits for Switched Optical Networks

- Push adaptation points toward shared edge networks
- Reduced number of contending entities for scheduling (1 < Entities < Nodes)
  - Advanced scheduling brokers
- Buffering for efficient coarse-grained burst switching
- Trusted signaling points able to evaluate rich policy definitions
  - IP-based ACLs are insufficient

LSL-NP

- Implementation of LSL on Network Processor from Intel
- The IXP2800 can forward at 10Gb/sec with ~60 cycle budget
  - We're using the IXP2400 which differs in the number of packet handling Micro-engines
- Xscale processor handles connection establishment
- Microengines do as much of the work as possible
- Many router vendors can insert an NP blade

LSL Summary

- Logistical data overlays can significantly improve performance for data movement
  - Demonstrated speedup
- Think of a *session* as the composition of network-specific transport layers
- There are many cases in which a single transport protocol from end to end might not be the best choice
  - Network heterogeneity
    - Wireless
    - Optical (with time-division multiplexing)
- Potential to become a new model rather than short-term solution for TCP's problems

A New Model?

- Hierarchical Flow Control
  - Recall: $BW = \frac{mss}{rtt\sqrt{p}} * C$
- Separate integrity assurance from flow control
- Utilize a series of depots that accept megagrams over transport-layer connections
  - 100s of MB rather than KB
- TCP becomes part of a link layer, TCP-type algorithms used at the megagram level

Discussion Points

- Incremental deployability
  - No need to upgrade end hosts
  - Reduced load on end hosts
- TCP friendly at the edges
- Adaptation point for network tuning
  - Signaling, Transport

Generic Session Layer (GSL)

- New version of the LSL protocol that unifies the LSL and Phoebus work
- Support for Session-Layer Frames
  - Essentially the megagrams discussed previously
- Rendezvous protocol for changing IP addresses
  - IPv6 multihoming issues
  - Is Shim6 really the answer?

Questions or Comments?

- Thank you for attending!