Ezra Kissel and Martin Swany

Session Layer Burst Switching for High Performance Data Movement

Funded in part by

Introduction

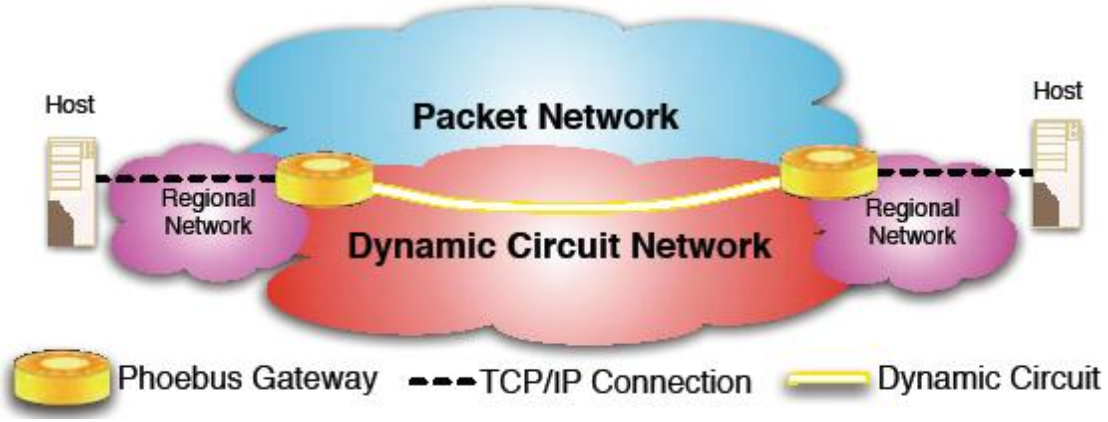
- Bulk data movement remains a key issue
- Application performance has not kept pace with advances in network technologies
- Existing protocols struggle to provide adequate throughput over heterogeneous network paths
- Time to reevaluate end-to-end arguments?
- What role can embedded data movement services play within the network?

Common problems

- Modern networks span a variety of technologies
  - Each may have very different characteristics
- Protocol tuning is necessary in order to achieve good performance
- TCP is reactive, "closed-loop"
  - RTT is critical to responsiveness
- The "wizard gap" remains

Supporting bulk data movement

- Mass storage catching up performance-wise (e.g. SSDs)
- 10G becoming commonplace, 100G on the horizon
  - Link aggregation, DWDM
- "Hybrid" networks
  - Dynamically allocate some links for high-demand flows
  - Virtual circuits
  - Emerging passive optical networks (PONs)
- How to effectively utilize these high-performance paths?



Phoebus and XSP

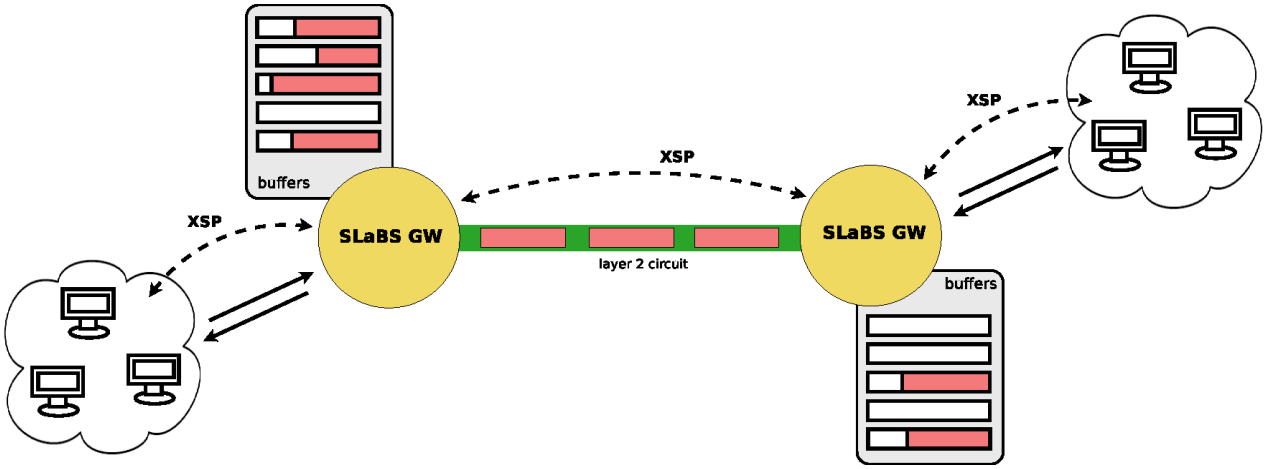
- The Phoebus project aims to help bridge the performance gap in modern networks
  - Brings high bandwidth and dynamic networks to end-users – easily!
  - Is based on the concept of an end-to-end "session" that enables multiple adaptation and buffer points, "gateways", in the network
  - Provides a notion of intelligence within the network
- Phoebus offers a gateway for legacy application to use advanced networks such as the Internet2 ION virtual circuit network
- eXtensible Session Protocol (XSP), a session-layer protocol for IP networks, provides generalized messaging between gateways and intermediate devices and services
- Standard interfaces at the edge, innovation at the core

Why a session layer?

- A session layer provides explicit control over adaptation/buffer points in the network
- Optimization of existing transport protocols
  - Congestion-based to rate-based
  - Shorter feedback loops
- Authorization and Authentication
  - Rich expression of policy via e.g. the Security Assertion Markup Language (SAML)
- Explicit control of Session-PDUs (SPDUs)
  - Buffering, forwarding, multi-pathing, coalescing, etc.

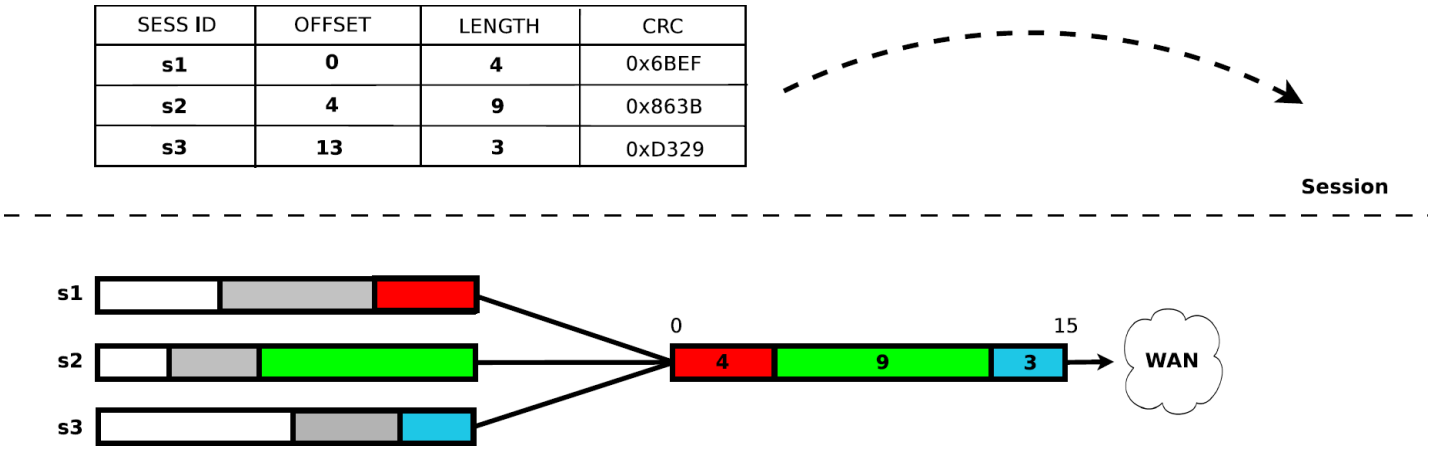
SLaBS

- Apply burst switching concepts at session-enabled gateways
  - Send relatively large PDUs versus small layer-3, layer-4 PDUs common today
- Schedule and optimize bursts over dedicated resources
- Reduce protocol overhead
- Hide provisioning latencies



Slabbing and SPDUs

- Observation: better utilization can be achieved with N elements sending at maximum rate for 1/Nth of time slot versus N elements competing for 1/Nth of bandwidth
- XSP enables Session-PDU (SPDU) formation – a "slab"
- Buffering required in order to multiplex incoming SPDUs into right-sized bursts



SLaBS prototype

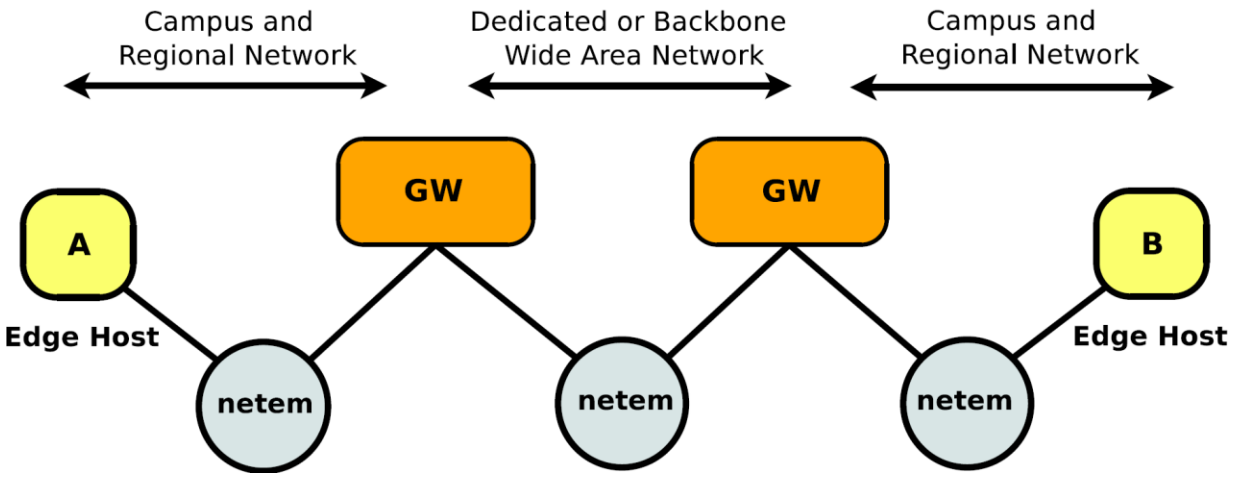
- Modular extension to Phoebus Gateway
  - Dynamically sized ring-buffer implementation
  - Buffers incoming flows from the "edge"
- Out-of-band burst signaling with XSP
  - SPDU "slab" control information
  - Active session between gateway peers manage connections
  - Enables coarse-grained error recovery at the SPDU level
- Slabs transmitted over UDP data channel
  - Reduced overhead over high latency, virtually error-free links

Early testing goals

- Does this approach even work?
  - Spoiler: yes!
- Evaluate SLaBS performance compared to direct, competing TCP flows in common networking conditions
- Collect measurements for a well-known file transfer tool (GridFTP) and a network performance benchmark (iperf) using both approaches
- Determine bursting performance for UDP data channel
  - Estimator for future lower-layer implementations

Experimental setup

- 7-node, 10G-connected testbed (Myricom NICs)
- Standard Linux tuning (2.6.26) and driver configuration
- Netem and PSpacer enable latency and bottleneck emulation

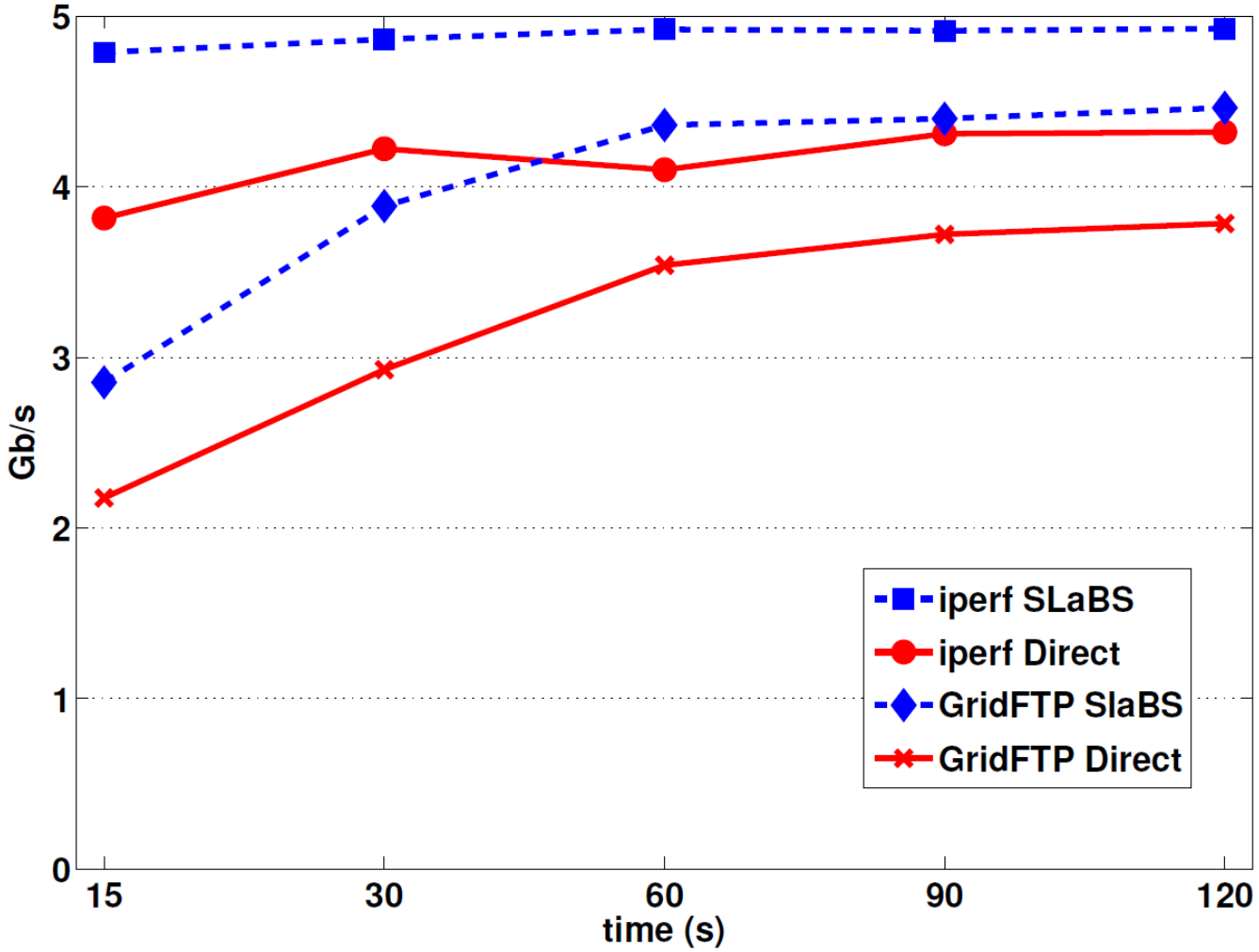


Methodology

- Memory-to-memory copies to avoid file system bottlenecks
- Bottleneck WAN link to introduce contention for available link capacity
  - 4 parallel TCP streams
- ~10ms RTT on edge links to simulate typical regional network connectivity
- Varied latency over WAN link to simulate various national and trans-continental backbone paths
- Repeatable experiments, results averaged over 10 identical runs

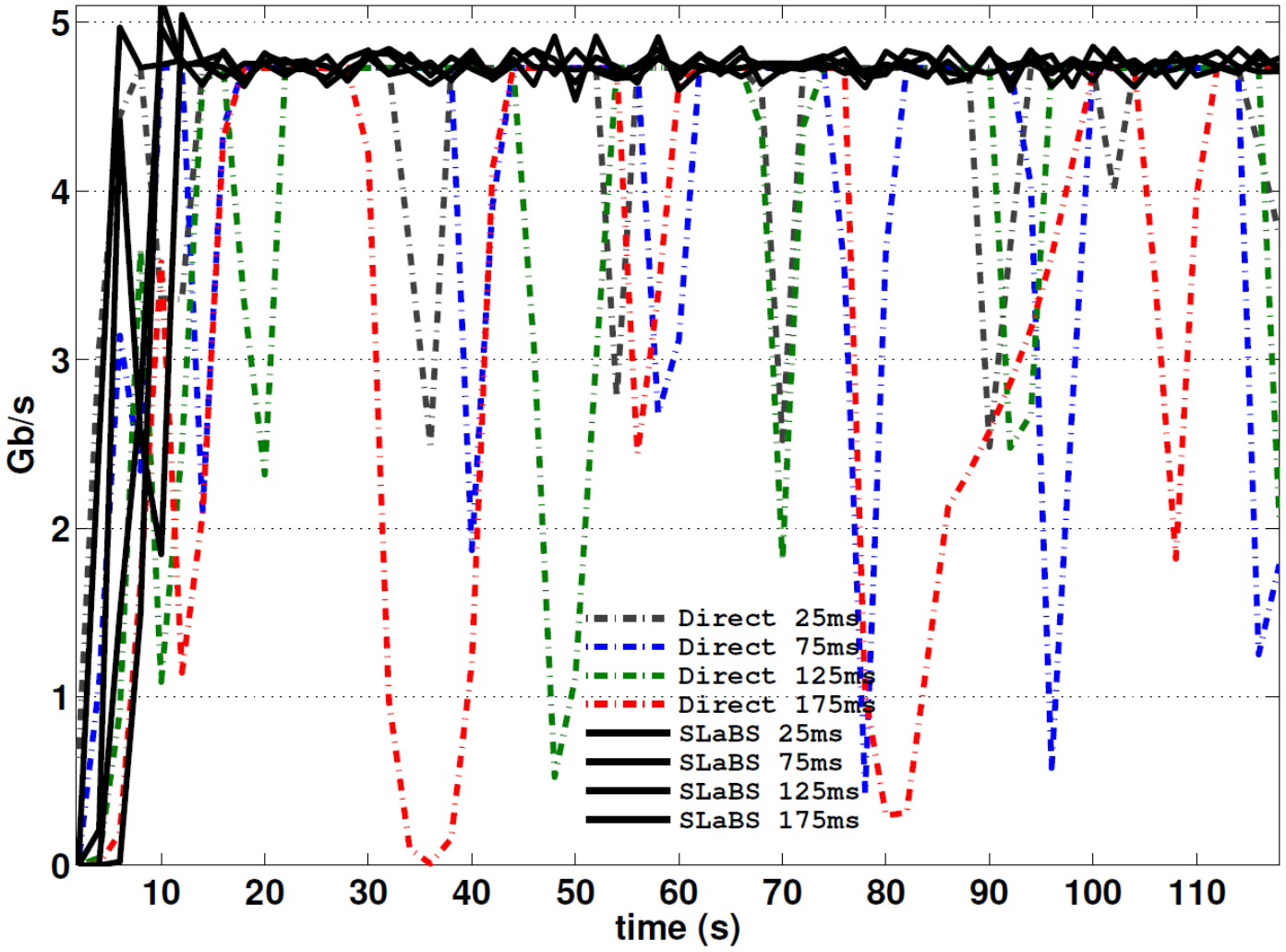
GridFTP and iperf

4 parallel streams over 5G WAN bottleneck with 115ms RTT

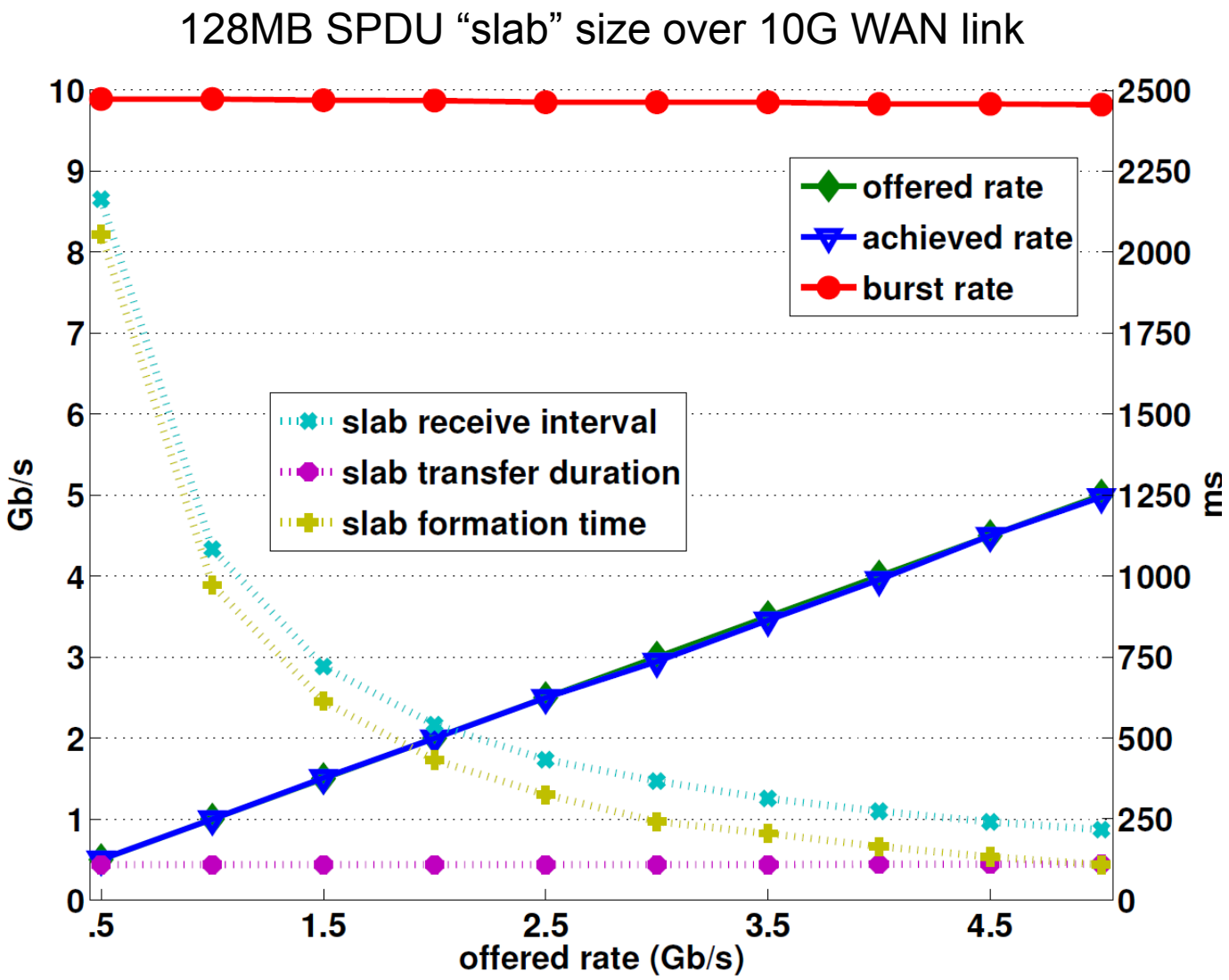


GridFTP with and without SLaBS

4 parallel streams, 5G WAN bottleneck with increasing latency



Bursting performance



Future work

- Improve SLaBS buffer implementation and performance
- Evaluate E2E connection reliability, control, and fate sharing
- Investigate large-scale deployments
  - Simulation, scheduling considerations
- Hiding provisioning delays with extended buffers
  - Dynamic resource allocation
- Multi-path at slab granularity

Summary

- SLaBS is an "in-the-network" approach to data movement
  - Supported by Phoebus and XSP
- Right-sized SPDU formation ("slabbing") is important
- Better utilization of high-performance network resources
  - Improved throughput for bulk data flows
- A departure from traditional E2E connection management
  - Session-layer protocol facilitates more precise control for heterogeneous and dynamic network environments