Optical Friendly HiSpeed File Transfer Protocol for Enabling Next Generation Nomadic Virtual PC Services

ECOC 2011, Geneva

HPCN, Universidad Autonoma de Madrid

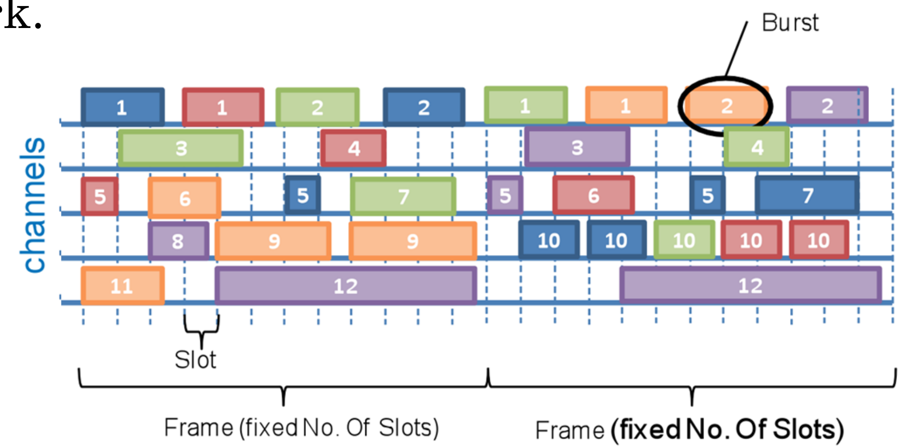
Telefonica I+D

CONTENTS

- Quick MAINS Overview
  - MAINS Optical Network: OBST/TSWON
  - MOTIVATION
    - TCP and OBST/TSWON
    - Details
  - MAINS Architecture & Service
  - Distributed Datacenter Model
- MAINS Transfer Protocol
  - Protocol Characteristics
  - Results
  - Implementation Details

MAINS Optical Network (OBST/TSWON)

- TSWON (U. of Essex, UK): Tunable Sub-Wavelength Optical Network
  - delivers highly flexible statistically-multiplexed optical network infrastructure guaranteeing contention-free packet/circuit services. Both a time-shared utilization of optical resources (ie. wavelengths) and a two-dimensional tunability (frequency-domain and time-domain) across all the ingress nodes of the optical network.



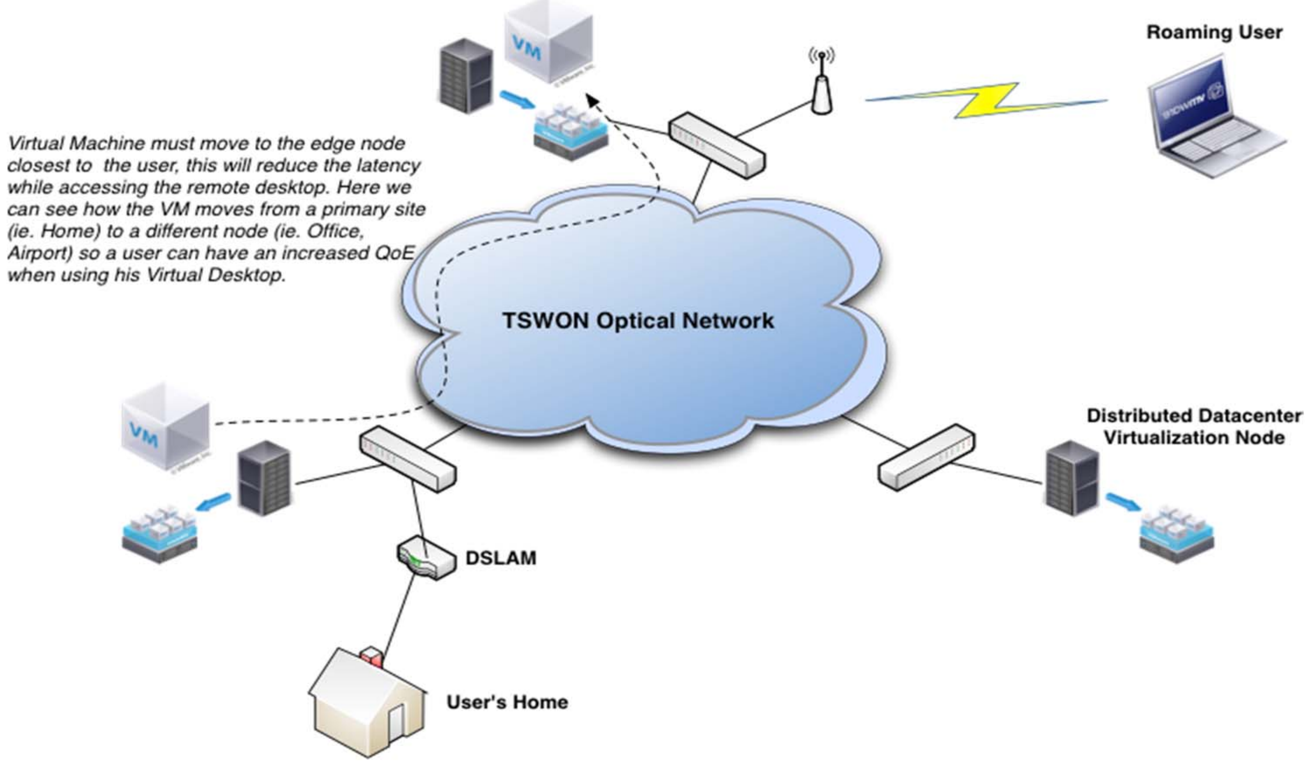
Motivation: TCP and OBST/TSWON, Not Always Good Enough.

- These optical architectures suffer **no congestion**, light paths provide **end-to-end connections**. TCP, however, is highly focused at avoiding it (slow-start, fast-retransmit, throttling, SACKs, etc). Contention shouldn't cause throttling.
- TCP/IP ACK's not piggy-backing data frames are precisely that: small packets. Not adequate for underlying fabric.
- Standard TCP doesn't perform too well in LFN (high bandwidth-delay product) anyways.

Motivation: Details

- **Tunability Speed**: laser tuning occurs in the 40-50ns range for state-of-the-art optical switches.
- **Traffic**: small and sparse traffic may prove a worst-case scenario. Transmission of a 60B data frame at 10Gbps takes ~48ns. That's a tuning overhead of 50%. If these tiny packets' inter-packet times are large enough, optical resource allocation will be suboptimal, hurting network utilization.
- **Applications**: streaming (audio/video), remote desktop protocols, bulky data transfers (HD video caches), etc. All generate a great traffic load in one direction but small and sparse traffic in the reverse direction.

MAINS Architecture & Service



- Roaming Virtual PC service: VM images moves to access node through which user connects, drastically reducing latency and greatly improving QoE of remote desktop session.
  - Large VM images must be transferred from access node to access node.
- To maximize network utilization we must avoid small packets.
  - Model avoids streaming remote desktop session through the metro network.
  - However, VM file transfer must be optimized.

MAINS AND THE DISTRIBUTED DATACENTER

- Applications injecting the tiny traffic we wish to avoid may be distributed over the edge nodes, thus avoiding such traffic in the metro-network.
- Again, user proximity to the service server will provide an enhanced QoE.
- ie. Remote Virtual PC
  - Virtual PC HD = 20GB
  - 20-minute RDP trace (web browsing, word processing, mailing, etc) = 2.0 GB worth of traffic.
    - ~1/3 of that traffic in terms of #packets was <79B.
    - Distributed datacenter model avoids pushing that traffic through the metro network.
- Distributed datacenter requires we move quickly and efficiently large volumes of data from edge-node to edge-node (Virtual PC images).

Protocol Characteristics

- NAK based Protocol.
  - Acceptable due to physical layer's low BER in metro-like distances.
- Not general purpose; file transfer protocol.
- Implementations
  - UDP
  - Ethernet (routing unnecessary when lightpath established)
- TCP-style 3-way handshake to initiate and terminate connections. Thus, stateful protocol.
  - Handshake establishes: ***connection port, MTU, filename, filesize***.
- Transfer happens in two phases (excluding handshakes).
  - Phase 1: Transmit the complete file, full-blast. Effectively converting this into a packet-capture process at the receiver.
  - Phase 2: Coalesce NAK's into large NAK frames, and request the retransmission of missing file offsets.

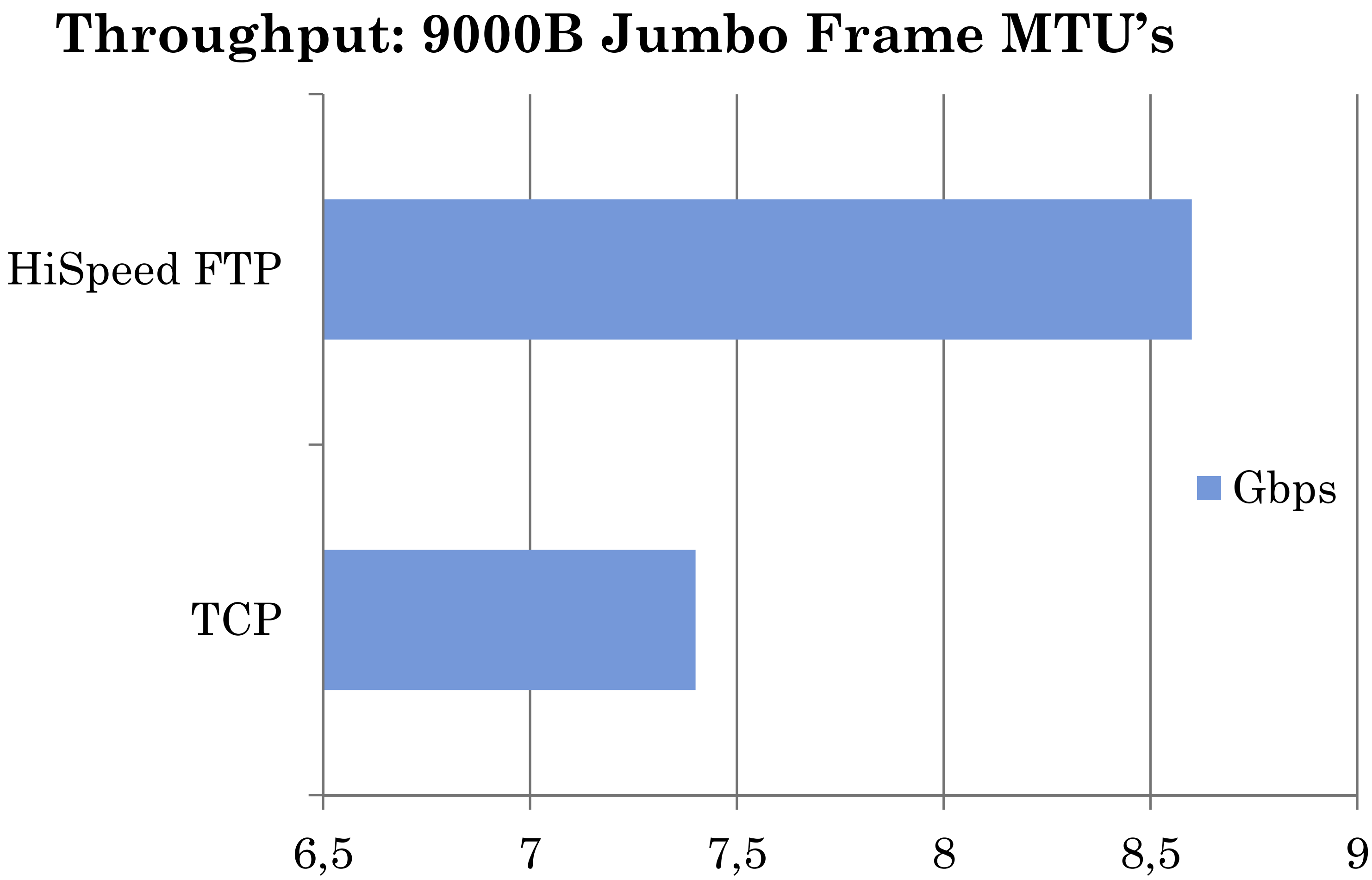
Protocol Characteristics

- Advantages:
  - Great percentage of packets sent at MTU size (>99.9% for 10 MB files and larger), maximizing OBST/TSON optical resource utilization and minimize the impact of tuning overhead.
  - Minimizes the number of packets sent.
  - Always attempting to make full use of link capacity.
  - Fairly sequential memory-HD access, this always yields slightly better performance (a lot better in the case of HD's).
  - Straight-forward implementation.
- Disadvantages vs TCP:
  - Application specific (file transmission).
  - Decreased flow rate/packet drop control.

RESULTS

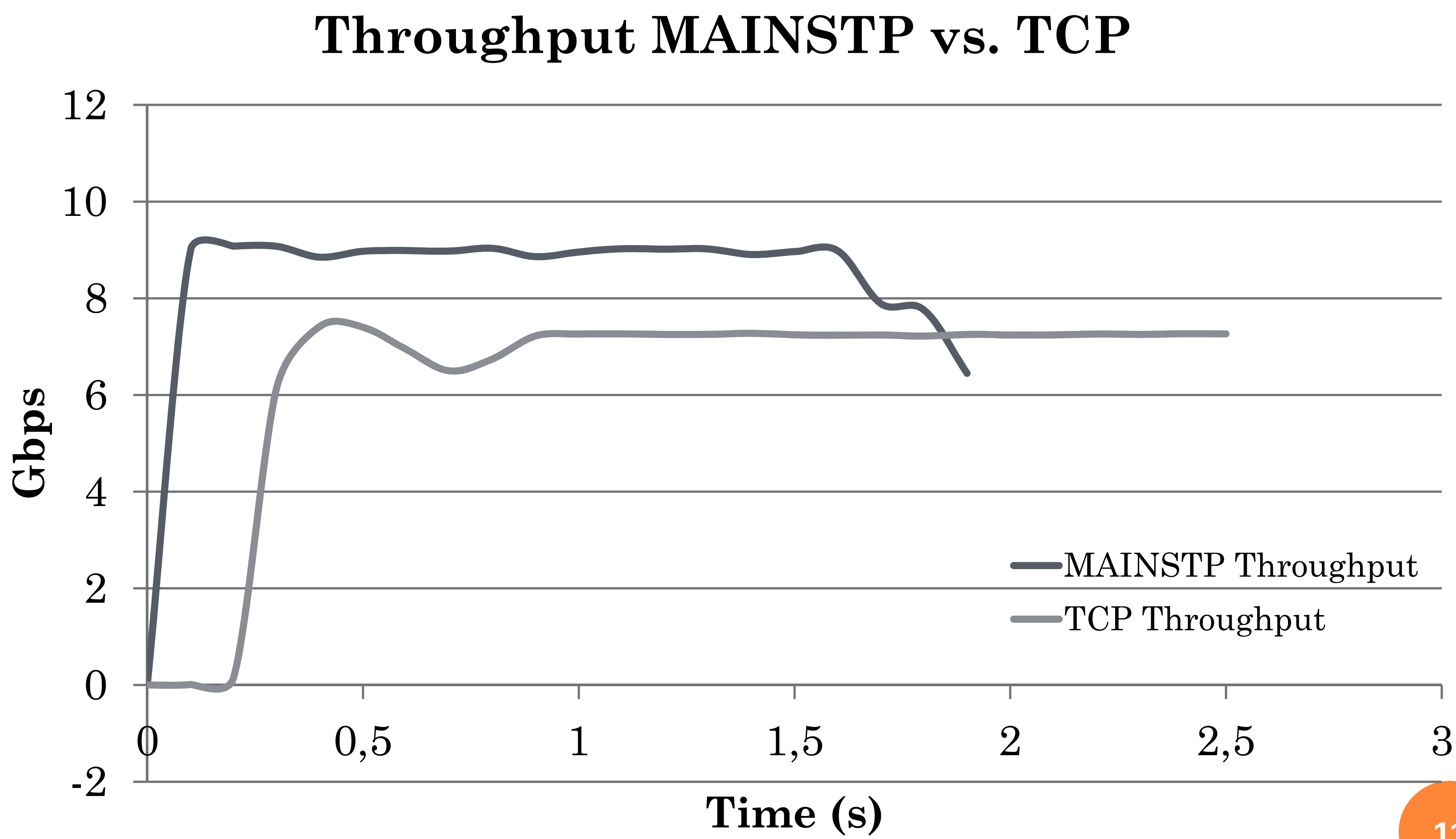
- 2 x Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz
- 6GB RAM
- 2 x 10GE (Intel 82598) NICs
- Jumbo Frame MTU: 9K
- Test file: 2GB iso (RAM loaded)
- Rates Achieved (averages over 30 file transfers):
  - TCP: ~**7.4 Gbps**
  - MAINS HiSpeed FTP: ~**8.5 Gbps**
- Current results would allow to transmit a 60GB iso in ~61 secs.
  - Your 60GB virtual machine can roam with you in just over a minute.

RESULTS



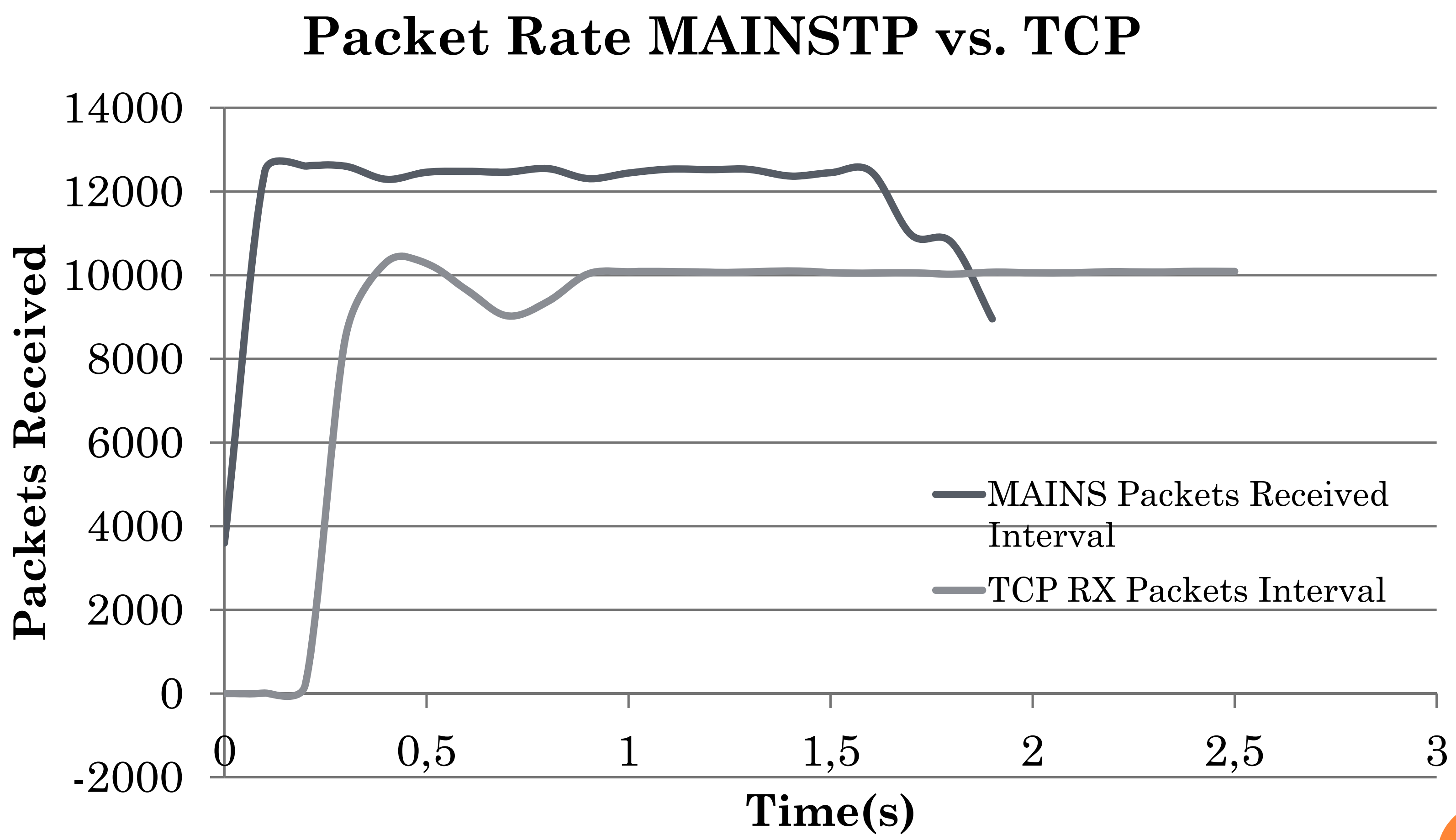
Throughput: 9000B Jumbo Frame MTU's
HiSpeed FTP
TCP
Gbps
6,5
7
7,5
8
8,5
9

RESULTS



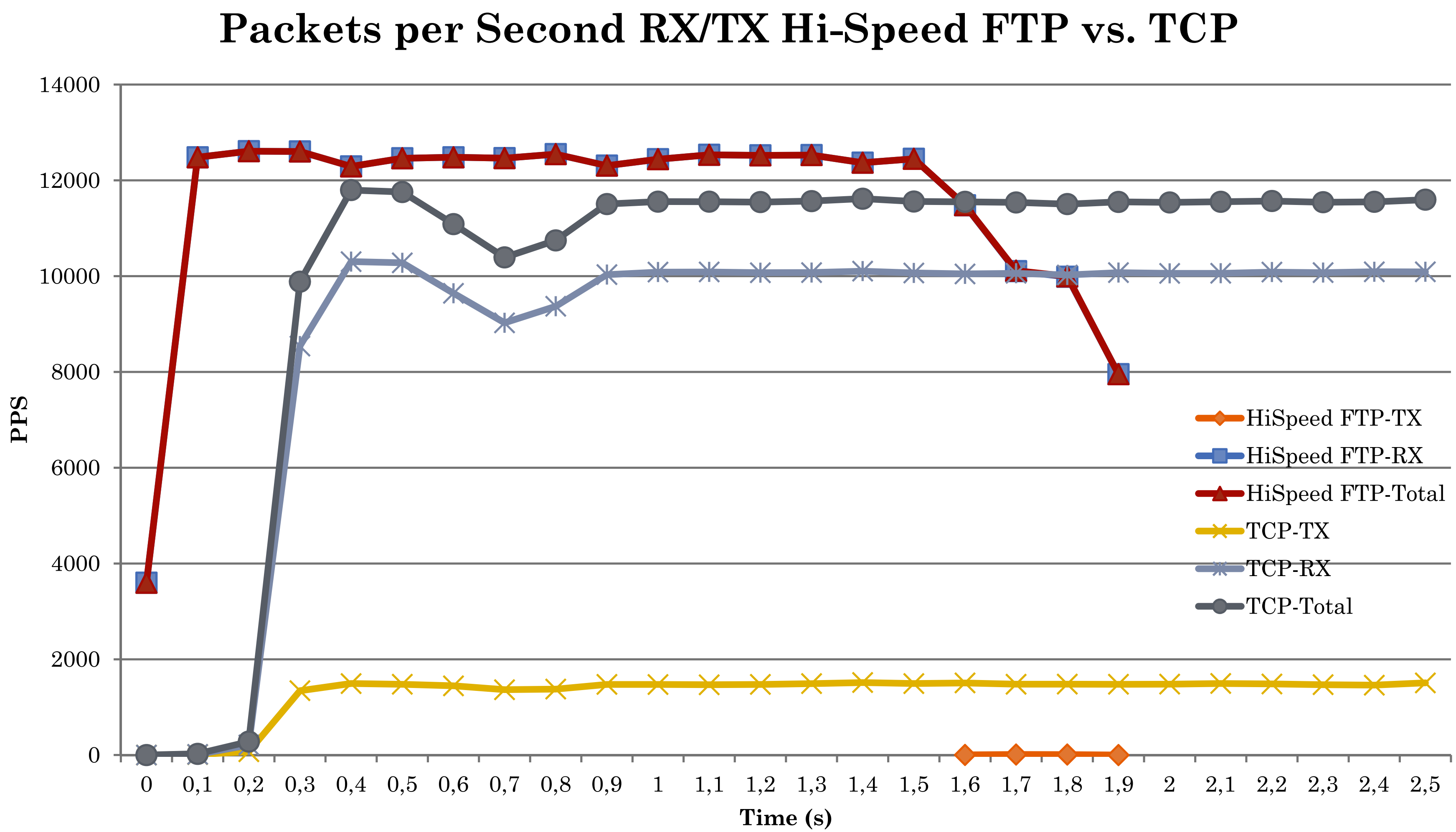
Throughput MAINSTP vs. TCP
12
10
8
6
4
2
0
-2
Gbps
0
0,5
1
1,5
2
2,5
3
Time (s)
MAINSTP Throughput
TCP Throughput

RESULTS

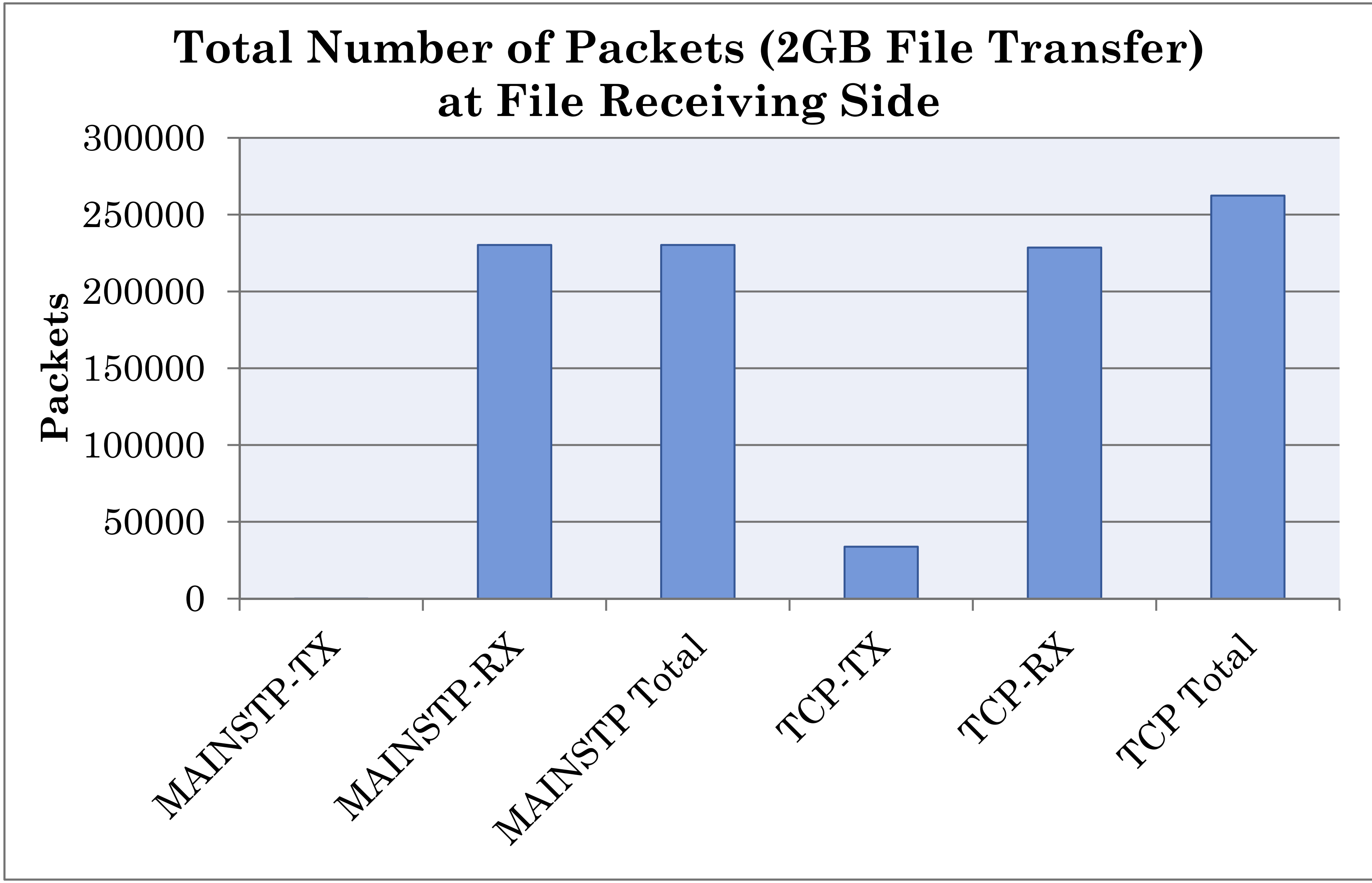


Packet Rate MAINSTP vs. TCP
Packets Received
14000
12000
10000
8000
6000
4000
2000
0
-2000
0
0,5
1
1,5
2
2,5
3
Time(s)
MAINS Packets Received Interval
TCP RX Packets Interval

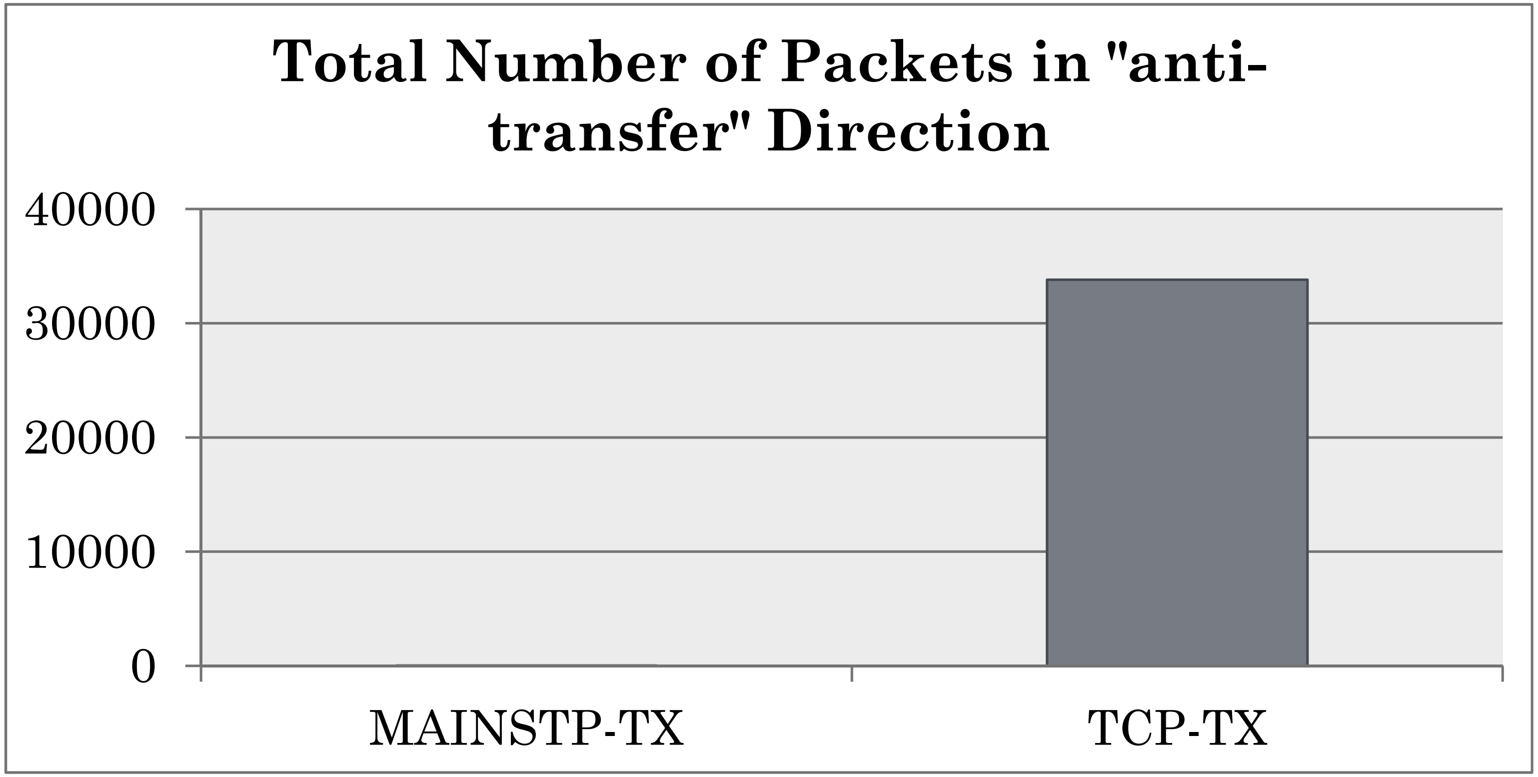
RESULTS


Packets per Second RX/TX Hi-Speed FTP vs. TCP
PPS
Time (s)
14000
12000
10000
8000
6000
4000
2000
0
0
0,1
0,2
0,3
0,4
0,5
0,6
0,7
0,8
0,9
1
1,1
1,2
1,3
1,4
1,5
1,6
1,7
1,8
1,9
2
2,1
2,2
2,3
2,4
2,5
HiSpeed FTP-TX
HiSpeed FTP-RX
HiSpeed FTP-Total
TCP-TX
TCP-RX
TCP-Total

RESULTS


Total Number of Packets (2GB File Transfer) at File Receiving Side
Packets
300000
250000
200000
150000
100000
50000
0
MAINSTP-TX
MAINSTP-RX
MAINSTP Total
TCP-TX
TCP-RX
TCP Total

RESULTS



- MAINSTP sends 25 packets of size MTU in anti-transfer direction requesting retransmissions.
- TCP (SACK enabled) sends ~35000 small packets in anti-transfer direction, for acknoledgements.

IMPLEMENTATION DETAILS

- Protocol Performance
  - Minimize packet loss during Phase 1.
    - Reduced I/O Path: the less times a packet is copied within the kernel, and the sooner it is available to the protocol implementation (userspace or kernelspace) the less frames will be dropped.
      - PACKET_MMAP: allows receiving traffic in userspace with ***zero*** packet copies. (done)
      - Protocol in kernelspace. (done: kernel module)
      - Modified drivers (ie. PacketShader by KAIST, South Korea).
  - Adjust transmission speed to available bandwidth.
    - Although protocol performs no active throttling, it initially configures to specified available bandwidth.
    - When the light path is established, if we are only assigned 4Gbps, ensure that we respect this to ***avoid overwhelming optical switches***.

Thank You. Questions?