Effect of Transfer File Size on TCP-ADaLR Performance: a Simulation Study

Modupe Omueti and Ljiljana Trajković
{momueti, ljilja}@cs.sfu.ca

Communication Networks Laboratory
http://www.ensc.sfu.ca/research/cnl
Simon Fraser University
Vancouver, Canada

communication networks laboratory

Roadmap

- Introduction
- Overview of TCP
- Background and related work
- TCP with adaptive delay and loss response (TCP-ADaLR) algorithm
- Simulation scenarios and performance evaluation
- Conclusions

Introduction

- File transfer protocol FTP:
  - is the primary protocol used for file transfers
  - relies on the transmission control protocol TCP for reliable transport
- Current Internet trend:
  - growth in IP communications
  - increasing demand in multimedia and data applications
- Transmission control protocol:
  - provides reliable byte-stream transport for most Internet applications such as FTP
  - carries up to 90% of Internet traffic

IP: Internet Protocol

Roadmap

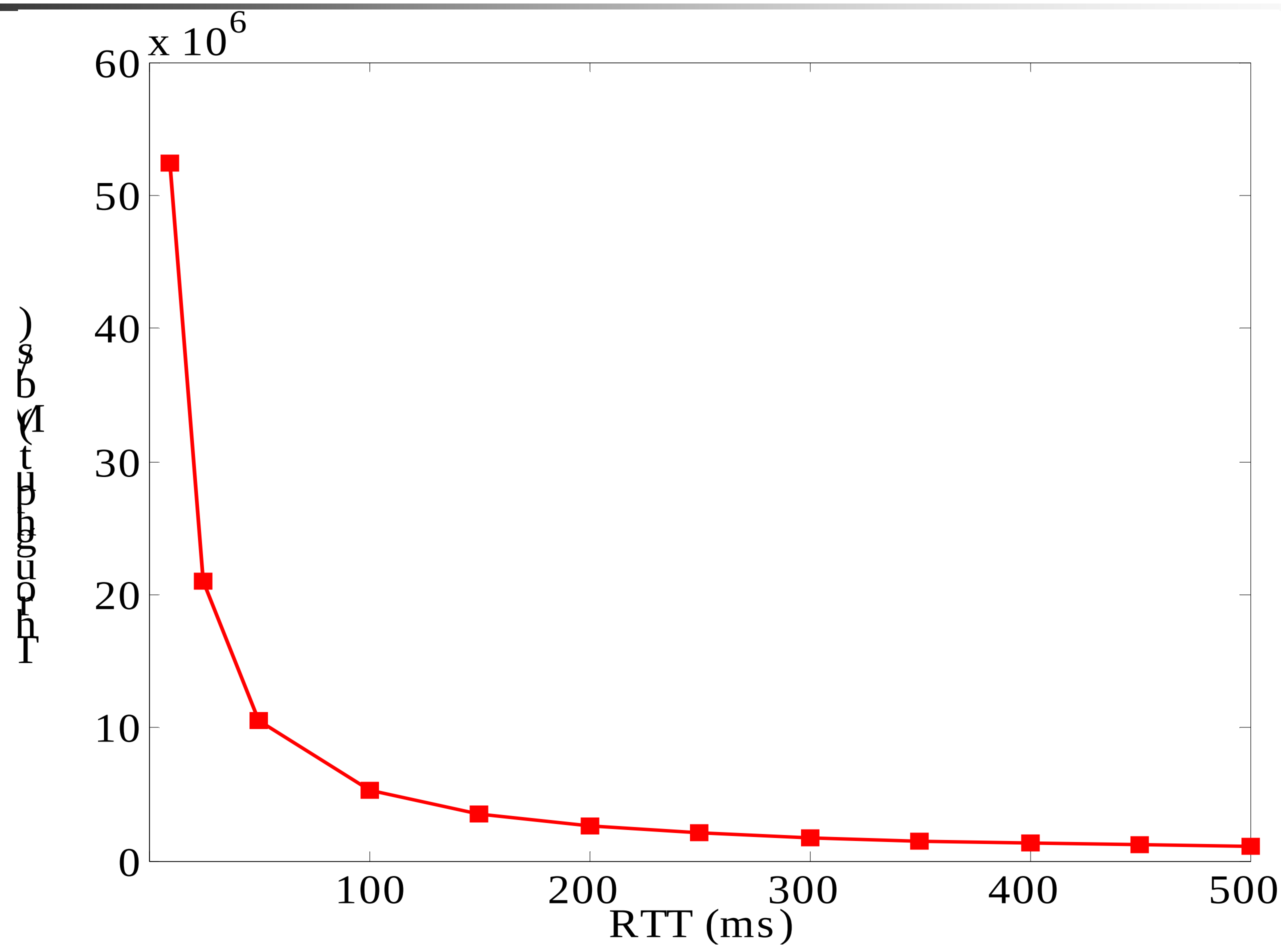
- Introduction
- Overview of TCP
- Background and related work
- TCP with adaptive delay and loss response (TCP-ADaLR) algorithm
- Simulation scenarios and performance evaluation
- Conclusions

Overview of TCP

- Transmission control protocol TCP:
  - employs acknowledgement ACK mechanism for flow and congestion control
  - increases the congestion window cwnd size based on the value of receiver's advertised window rwnd and the number of segments or bytes acknowledged
  - performs poorly in the presence of large propagation delays
- Large bandwidth-delay product:
  - is the product of the link capacity and the RTT
  - characterizes networks with large propagation delays

RTT: round trip time

Overview of TCP



Achievable throughput is inversely proportional with RTT and, hence, links with larger RTTs are underutilized

Roadmap

- Introduction
- Overview of TCP
- Background and related work
- TCP with adaptive delay and loss response (TCP-ADaLR) algorithm
- Simulation scenarios and performance evaluation
- Conclusions

Related work

- HighSpeed TCP (HSTCP) and scalable TCP :
  - proposed for large bandwidth-delay product networks
  - adjust the cwnd adaptively based on the current cwnd when:
    - ACKs are received
    - TCP segments are lost
  - have cwnd adjustment parameters that are specifically optimized for Gigabit Ethernet wide area networks
- User Datagram Protocol UDP:
  - proposed for accelerating large file transfers in networks with large RTTs
  - is an unreliable connectionless protocol

Related work

- Reliable blast UDP (RBUDP):
  - employs TCP for exchanging control information
  - employs UDP for transferring data
  - requires a priori knowledge of the rate of the bottleneck link between the sender and receiver
- Simple available bandwidth utilization library (SABUL) combines:
  - UDP with rate-based control
  - TCP-like window-based control

Roadmap

- Introduction
- Overview of TCP
- Background and related work
- TCP with adaptive delay and loss response (TCP-ADaLR) algorithm
- Simulation scenarios and performance evaluation
- Conclusions

TCP-ADaLR: TCP with adaptive delay and loss response

- End-to-end solution for improving TCP performance:
  - mitigates the effect of large propagation delays, error losses, and the limited TCP rwnd on TCP throughput
  - introduces modifications only at the TCP sender
    - scaling component ρ
    - adaptive cwnd increase mechanism
    - adaptive rwnd increase mechanism
    - loss recovery mechanism
- TCP-ADaLR modifications are applicable to:
  - TCP SACK
  - TCP NewReno

TCP-ADaLR: TCP with adaptive delay and loss response

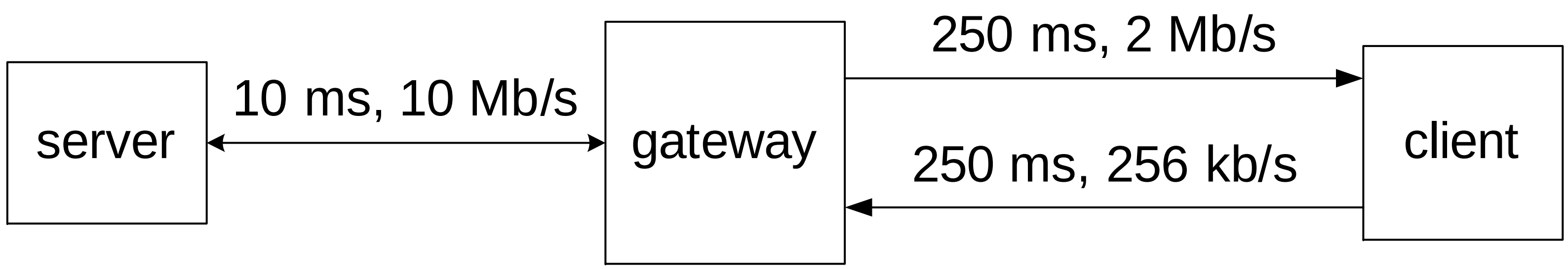
- Improves TCP performance by taking the effect of RTT on the window increments:
  - computes scaling component ρ from measured RTT
  - adjusts the cwnd and rwnd based on the value of scaling component ρ
- Allows faster transmission of additional segments in the presence and absence of losses
- Limits the number of retransmissions in order to prevent a large number of unnecessary retransmissions

RTT: round trip time
cwnd: congestion window
rwnd: receiver's advertised window

Roadmap

- Introduction
- Overview of TCP
- Background and related work
- TCP with adaptive delay and loss response (TCP-ADaLR) algorithm
- Simulation scenarios and performance evaluation
- Conclusion

Network topology



- Propagation delays are one-way
- Ethernet link between the gateway and the server is full-duplex with a data rate of 10 Mb/s
- Link between the gateway and the client is asymmetric with data rates of 2 Mb/s downlink and 256 kb/s uplink
- Links are lossless

Simulation scenarios and performance metrics

- Simulation scenarios:
  - 50 MB file download for various link propagation delays 10 ms, 25 ms, 50 ms, 100 ms, 200 ms, 250 ms, 300 ms, 350 ms, 400 ms, 450 ms, and 500 ms
  - FTP file download for various file sizes 500 kB, 50 MB, 100 MB, 200 MB, 300 MB, 400 MB, and 500 MB
- Performance metrics:
  - FTP download response time
  - TCP throughput
  - link throughput and utilization

TCP simulation parameters

TCP Parameter	Value
Initial RTO	3.0 s
Minimum RTO	1.0 s
Maximum RTO	64.0 s
Timer granularity	0.5 s
Persistent timeout	1.0 s
Maximum ACK delay	0.0 s
Maximum ACK segment	1
Duplicate ACK threshold	3
Sender maximum segment size (SMSS)	1,460 bytes
Slow start initial count	2
Receiver's advertised window (*rwnd*)	65,535 bytes
Retransmission threshold	6
RTT gain	0.125
RTT deviation gain	0.25
RTT deviation coefficient	4

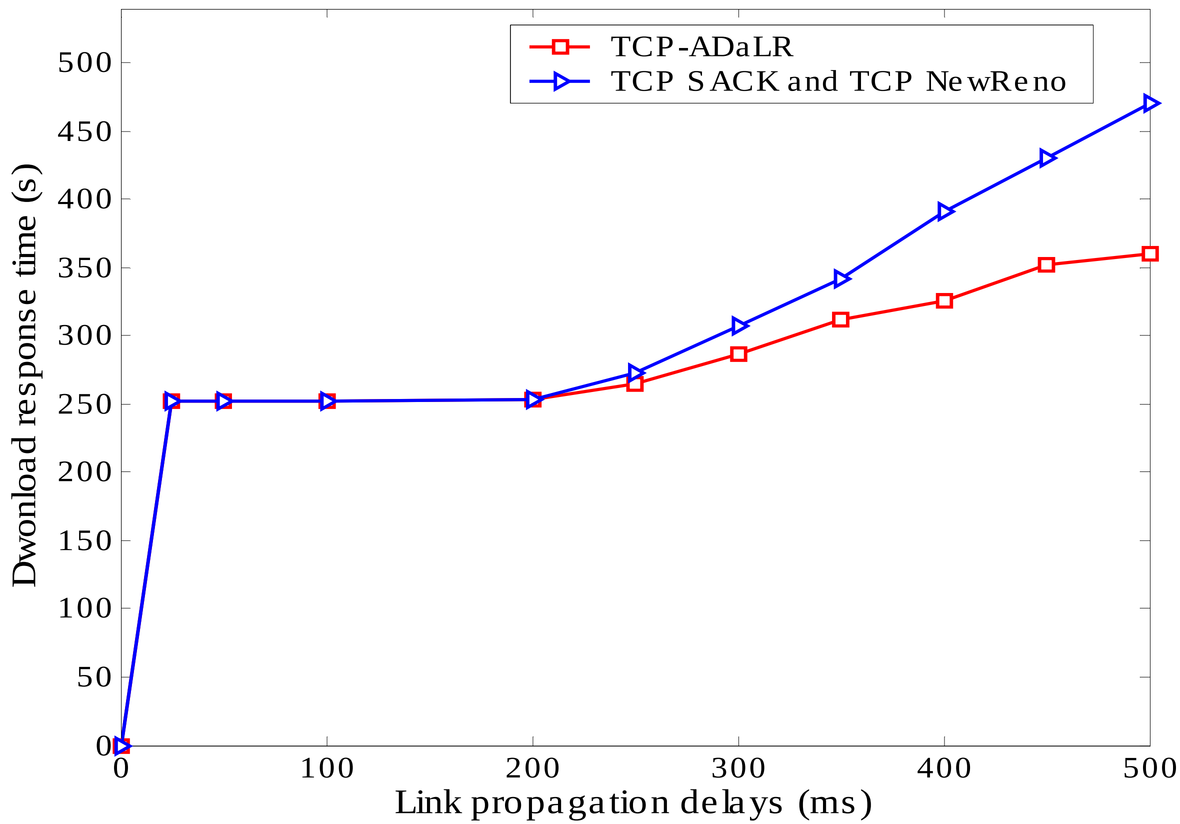
FTP file application parameters

Attribute	Value
File inter-request time (s)	18,000
File inter-request time distribution	constant
File size (MB)	50
File size distribution	constant
Simulated time (hours)	5

Roadmap

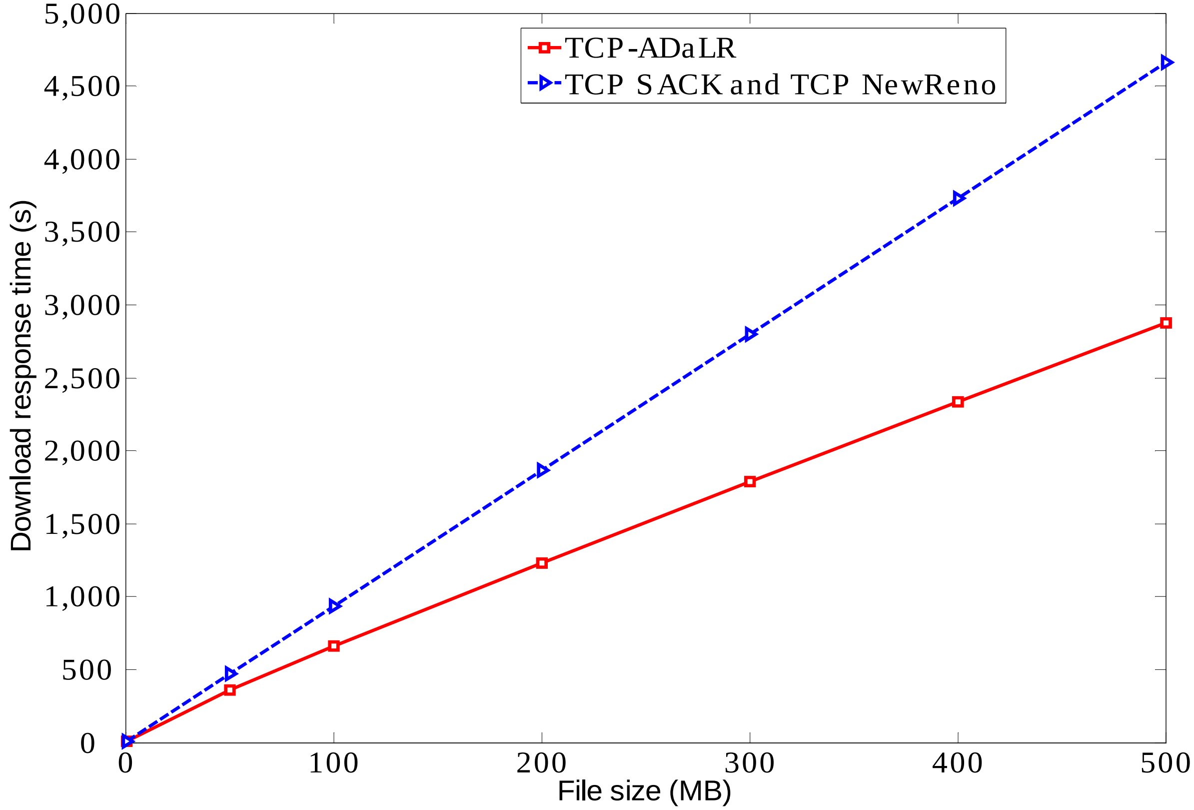
- Introduction
- Overview of TCP
- Background and related work
- TCP with adaptive delay and loss response (TCP-ADaLR) algorithm
- Simulation scenarios and performance evaluation:
  - 50 MB file download for various link propagation delays
  - FTP file download for various file sizes
- Conclusions

Scenario with various link propagation delays: download response time



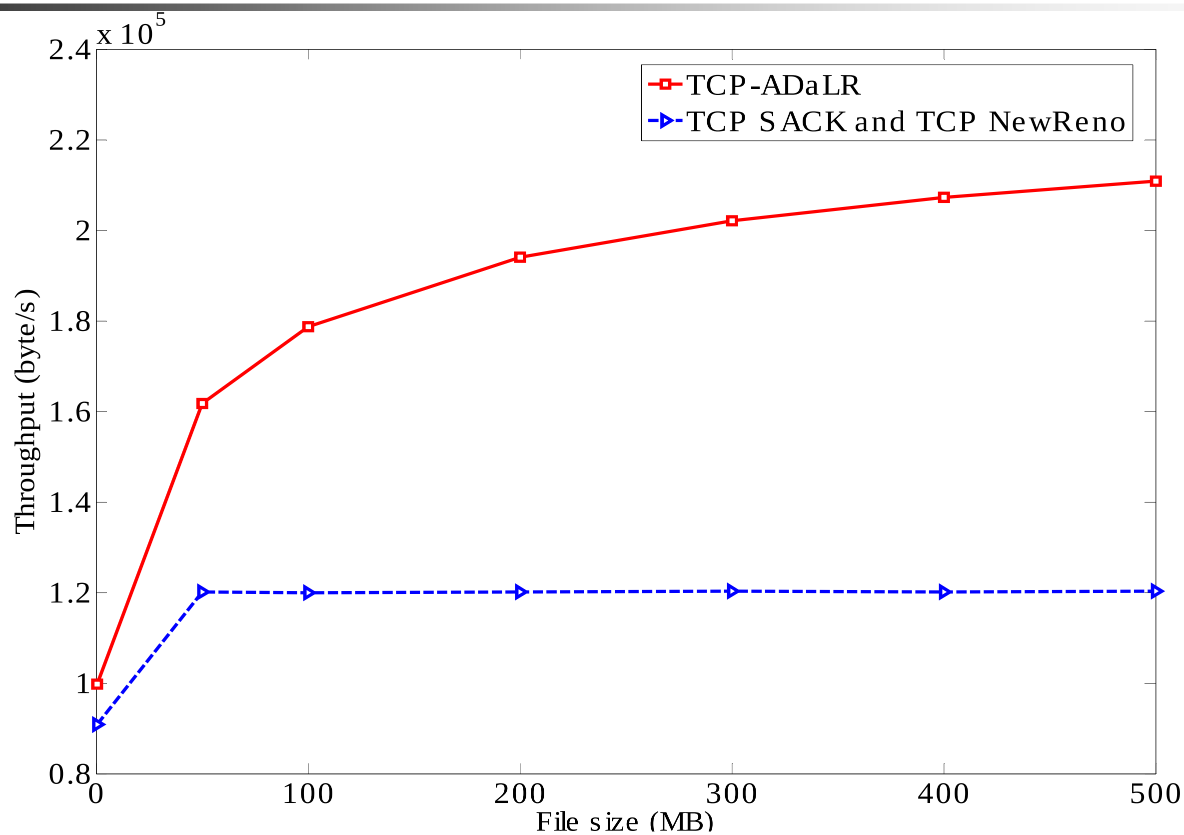
- TCP-ADaLR, TCP SACK, and TCP NewReno exhibit comparable performance for RTTs < 200 ms
- TCP-ADaLR exhibits 5.5%–75% shorter download response time than TCP SACK and TCP NewReno for RTTs > 200 ms

Scenario with various FTP file sizes: download response time



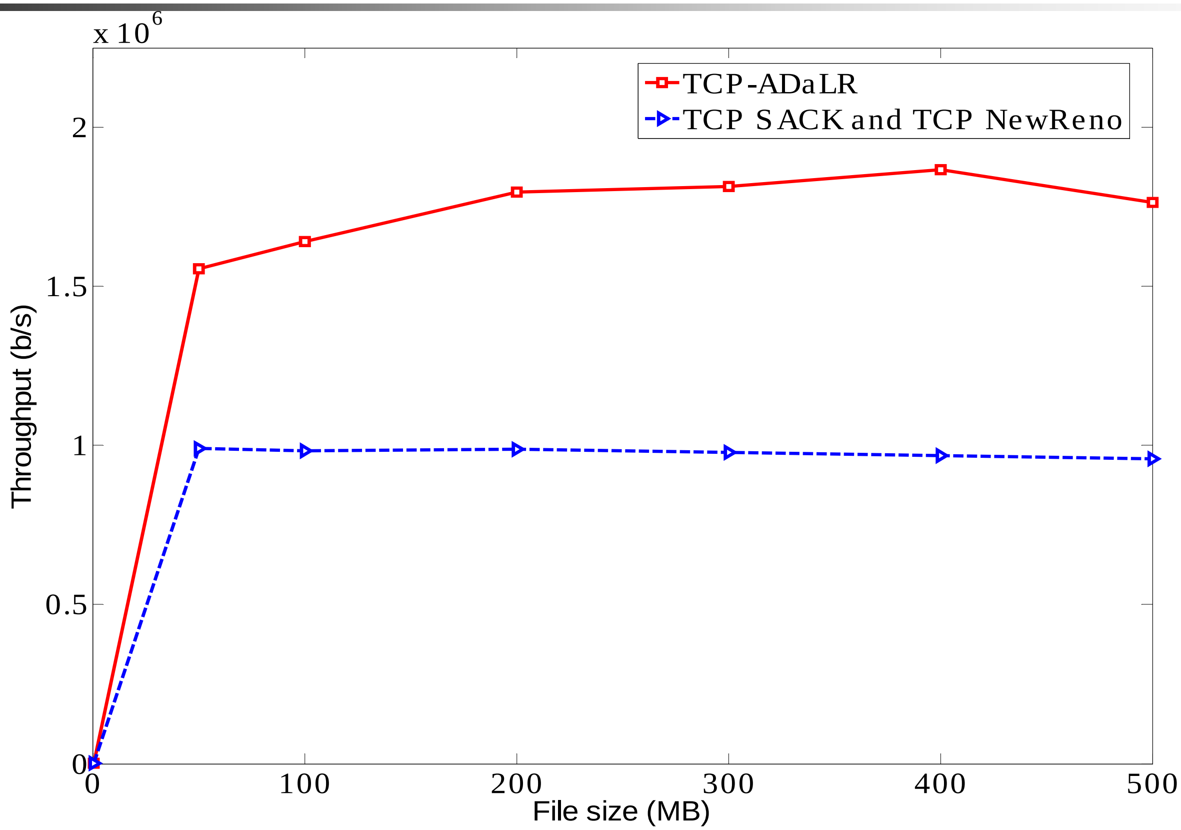
- As file size increases, FTP file download response time increases for TCP-ADaLR TCP SACK, and TCP NewReno connections
- TCP-ADaLR shows 9%–38% shorter download response times than TCP SACK, and TCP NewReno

Scenario with various FTP file sizes: TCP throughput



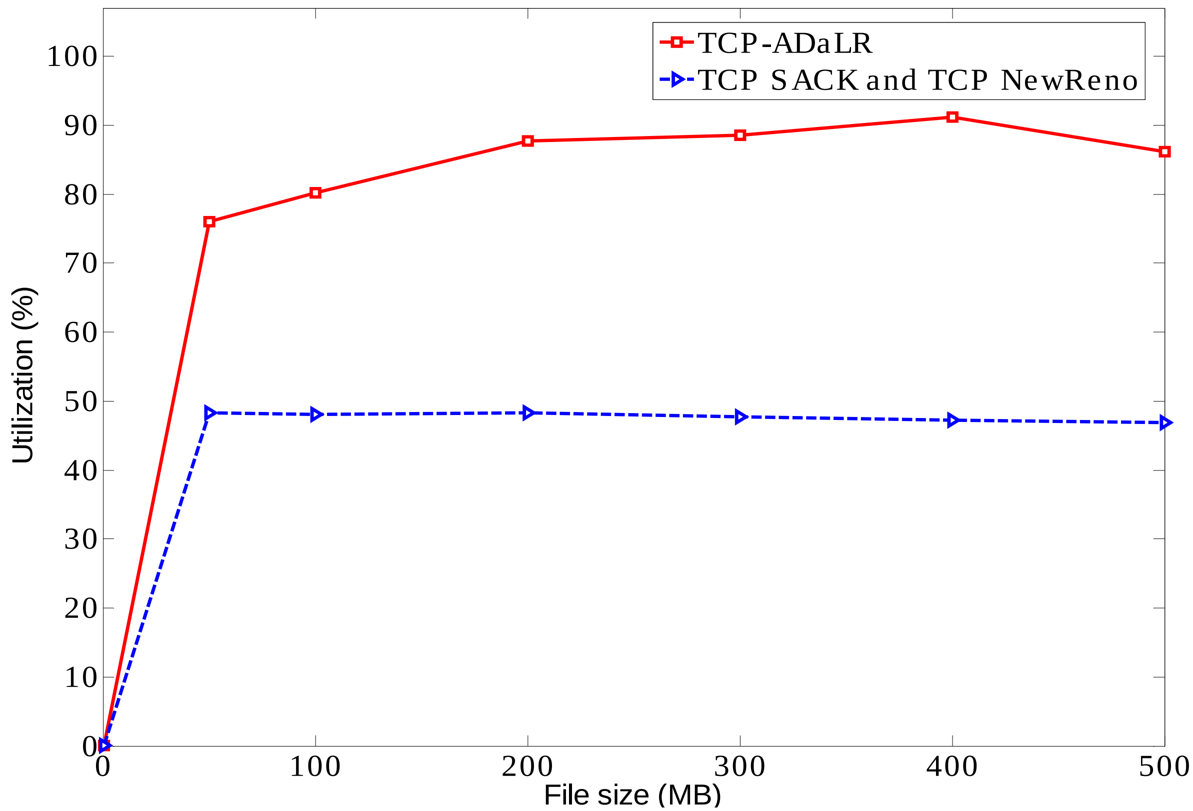
- As file size increases, TCP-ADaLR shows 9%–75% higher TCP throughput compared to TCP SACK, and TCP NewReno
- TCP throughput of TCP SACK and TCP NewReno remains unchanged when the transfer file size exceeds 50 MB

Scenario with various FTP file sizes: link throughput



- As file size increases, TCP-ADaLR exhibits up to 81% higher link throughput than TCP SACK, and TCP NewReno
- For TCP SACK and TCP NewReno, the value of rwnd limits the throughput to ~ 1 Mb/s

Scenario with various FTP file sizes: link utilization



- As file size increases, TCP-ADaLR exhibits 57%-90% higher link utilization than TCP SACK, and TCP NewReno
- For TCP SACK and TCP NewReno, the value of rwnd limits the throughput to ~ 50% of the available 2 Mb/s link capacity

Roadmap

- Introduction
- Overview of TCP
- Background and related work
- TCP with adaptive delay and loss response (TCP-ADaLR) algorithm
- Simulation scenarios and performance evaluation
- Conclusions

Conclusions

- TCP-ADaLR algorithm improved TCP performance for transfers of small to moderate and large files in high latency lossless networks
- TCP-ADaLR utilized link capacities better, accelerated file transfers, and exhibited higher TCP throughput than TCP SACK and TCP NewReno
- Deployment of TCP-ADaLR in existing networks requires modifications only at the TCP sender, with minimal processing and memory overhead
- Simulation results indicate that TCP-ADaLR is a viable option for large file transfers in existing networks that employ TCP as the transport layer protocol

References

- J. Postel and J. Reynolds, "File transfer protocol," *RFC 959*, Oct. 1985.
- J. Postel, Ed., "Transmission control protocol," *RFC 793*, Sept. 1981.
- W. Stevens, *TCP Illustrated Volume 1: The Protocols*. Reading, MA: Addison-Wesley, 1994.
- M. Allman, V. Paxson, and W. Stevens, "TCP congestion control," *RFC 2581*, Apr. 1999.
- M. Omueti and Lj. Trajković, "TCP with adaptive delay and loss response for heterogeneous networks," in *Proc. Third Annual International Wireless Internet Conference (WICON 2007)*, Austin, TX, Oct. 2007.
- M. Omueti and Lj. Trajković, "OPNET model of TCP with adaptive delay and loss response for broadband GEO satellite networks," *OPNETWORK 2007*, Washington, DC, Aug. 2007.
- S. Floyd, "HighSpeed TCP for large congestion windows," *RFC 3649*, Dec. 2003.
- T. Kelly, "Scalable TCP: improving performance in high speed wide area networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 2, pp. 83–91, Apr. 2003.
- J. Postel., "User datagram protocol," *RFC 793*, Aug. 1980.
- E. He, J. Leigh, O. Yu, and T. DeFanti, "Reliable blast UDP: predictable high performance bulk data transfer," in *Proc. 5th Int. Conf. on Cluster Computing*, Chicago, IL, Sept. 2002, pp. 317–324.
- Y. Gu and R. Grossman, "SABUL: A transport protocol for grid computing," *J. of Grid Computing*, vol. 1, no. 4, pp. 377–386, Dec. 2003.
- OPNET Modeler software [Online]. Available:http://www.opnet.com/products/modeler/home.html .