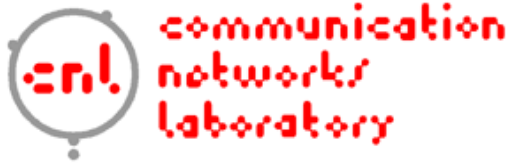
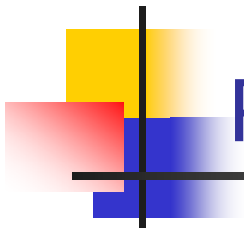
M-TCP+: using disconnection feedback to improve performance of TCP in wired/wireless networks

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

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





Roadmap

- Introduction
- Background and related work:
  - overview of TCP
  - M-TCP for mobile cellular networks
- M-TCP+ algorithm for wired/wireless network:
  - description
  - implementation
- Performance evaluation:
  - simulation scenarios and parameters
  - simulation results
- Conclusions

Introduction

- The Internet:
  - witnessed growth in wireless IP communications
  - incorporates wireless infrastructure to provide services for mobile and wireless hosts
- Transmission control protocol (TCP):
  - provides byte-stream transport for most Internet applications such as remote login, FTP, and HTTP
  - carries up to 90% of Internet traffic
  - originally designed for wired networks characterized by negligible bit error rates

M. Fomenkov, K. Keys, D. Moore, and K. Claffy, "Longitudinal study of Internet traffic in 1998-2003," in *Proc. ACM Winter Int. Symp. Inf. and Commun. Technologies*, Cancun, Mexico, Jan. 2004, pp. 1–6.

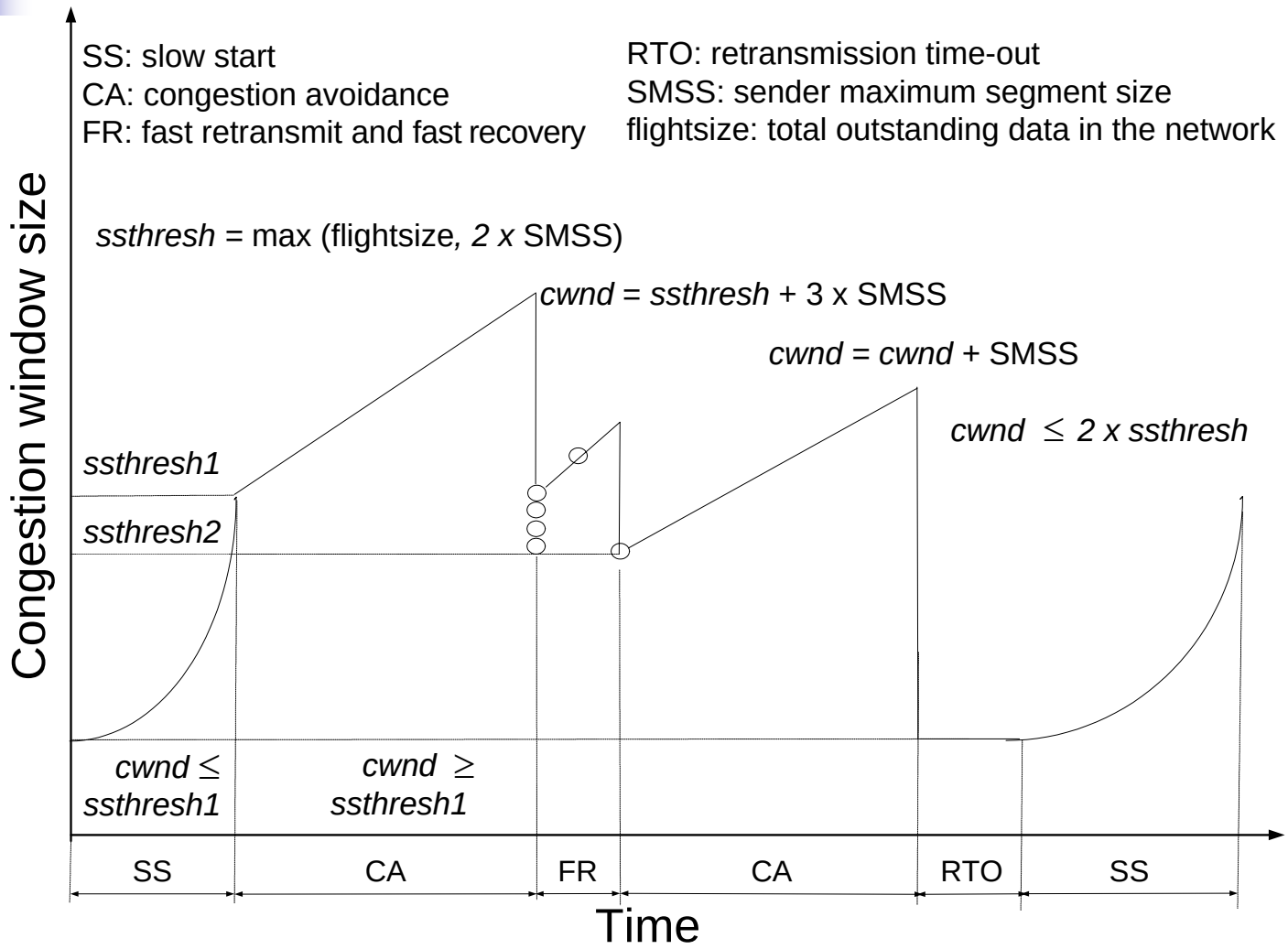
Roadmap

- Introduction
- Background and related work:
  - overview of TCP
  - M-TCP for mobile cellular networks
- M-TCP+ algorithm for wired/wireless network:
  - description
  - implementation
- Performance evaluation:
  - simulation scenarios and parameters
  - simulation results
- Conclusions

TCP congestion control algorithms

SS: slow start
CA: congestion avoidance
FR: fast retransmit and fast recovery

RTO: retransmission time-out
SMSS: sender maximum segment size
flightsize: total outstanding data in the network



TCP delayed acknowledgement option

- Allows TCP receivers to send an acknowledgement (ACK) for every second consecutive full-sized segment received
- Implemented by many TCP receivers in the Internet:
  - default interval period: 200 ms
  - maximum interval period: 500 ms
- Reduces protocol processing overhead
- Increases network efficiency and maximizes network bandwidth

J. Chen, Y. Z. Lee, M. Gerla, and M. Y. Sandidi, "TCP with delayed ACK for wireless networks," in *Proc. IEEE/CreateNet BROADNETS 2006*, San Jose, CA, USA, Oct. 2006, pp. 1–6.

W. Lilakiatsakun and A. Seneviratne, "TCP performances over wireless links deploying delayed ACK," in *Proc. 57th IEEE Veh. Technol. Conf.*, Jeju, Korea, Apr. 2003, vol. 3, pp. 1715–1719.

A full-sized segment is equivalent to the sender maximum segment size (SMSS)

TCP persist state

- TCP sender:
  - maintains an open connection by sending window probes periodically to receiver
  - refrains from transmitting data segments
  - enters persist state:
    - *rwnd* is reduced to zero
    - no outstanding unacknowledged data segments
  - employs the persist timer to maintain exchange of *rwnd* information
- TCP persist timer computes the expiration period similar to RTO timer

W. Stevens, *TCP Illustrated Volume 1: The Protocols*. Reading, MA: Addison-Wesley,1994.

TCP in wireless networks

- Heterogeneous wired/wireless networks:
  - designed to support a large number of mobile users
  - characterized by handoffs that cause disconnections and packet losses
- TCP in wireless networks:
  - misinterprets packet losses as an indication of congestion
  - reduces congestion window size and, consequently, throughput
  - induces serial timeouts from disconnections due to handoffs

Roadmap

- Introduction
- Background and related work:
  - overview of TCP
  - M-TCP for mobile cellular networks
- M-TCP+ algorithm for wired/wireless network:
  - description
  - implementation
- Performance evaluation:
  - simulation scenarios and parameters
  - simulation results
- Conclusions

M-TCP characteristics

- M-TCP network consists of a three-level hierarchy:
  - mobile hosts MHs
  - base stations BSs
  - supervisor hosts SHs:
    - are connected to fixed hosts FHs through a high-speed wired network
    - handle routing and connection management of MHs through BSs
- M-TCP:
  - maintains end-to-end semantics of TCP
  - employs split connection approach

K. Brown and S. Singh, "M-TCP: TCP for mobile cellular networks," *ACM Comput. Commun. Rev.*, vol. 27, no. 5, pp. 19–43, Oct. 1997.

M-TCP algorithm: supervisor host

- Connection is split at the SH
- Receives data segment FH
- Forwards received segment to MH through BS
- Receives new ACK from MH and forwards to FH ACK with ACK number reduced by one
- When disconnection is detected:
  - forces the FH into persist state with ACK number acknowledging the last byte and *rwnd* set to zero
  - assumes TCP sender will enter persist state with ACK number *ack_no* acknowledging the last (unacknowledged) byte

FH: fixed host
SH: supervisor host
BS: base station
MH: mobile host

Related work

- V-TCP mitigates adverse effect of host mobility on TCP performance, with TCP sender being either FH or MH
- Freeze-TCP:
  - improves TCP performance in the presence of frequent disconnections
  - sends zero window advertisement to force FH into persist state

D. Nagamalai, B. C. Dhinnakaran, B.-S. Choi, and J.-K. Lee, "V-TCP: a novel TCP enhancement technique," in *Networking-ICN, Lecture Notes in Comput. Science.* Springer, Berlin: vol. 3421, pp. 125–132, Mar. 2005.

T. Goff, J. Moronski, and V. Gupta, "Freeze-TCP: a true end-to-end TCP enhancement mechanism for mobile environments," in *Proc. IEEE INFOCOM*, Tel-Aviv, Israel, Mar. 2000, pp. 1537–1545.

FH: fixed host
MH: mobile host

Related work

- M-TCP, V-TCP, and Freeze-TCP:
  - all transmitted data segment have ACKs associated with them
  - last ACK number acknowledging the last byte forces TCP sender into persist state
- Delayed ACK enabled:
  - last ACK may not acknowledge the last outstanding data segment
  - TCP sender:
    - refrains from sending additional data segments
    - deflates *cwnd* when RTO timer expires

Roadmap

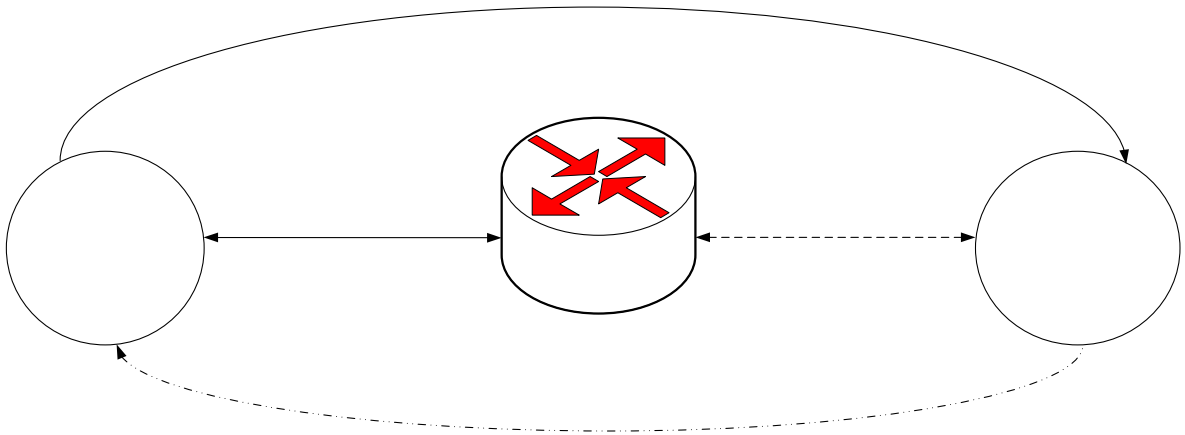
- Introduction
- Background and related work:
  - overview of TCP
  - M-TCP for mobile cellular networks
- M-TCP+ algorithm for wired/wireless network:
  - description
  - implementation
- Performance evaluation:
  - simulation scenarios and parameters
  - simulation results
- Conclusions

M-TCP+ algorithm

- Proposed for heterogeneous wired/wireless network with FH, intermediate host IH, and wireless host WH
- Considers cases with and without delayed ACK enabled
- Maintains end-to-end semantics of TCP in the absence of disconnections
- Employs disconnection feedback signal to force FH into persist state when disconnection occurs

FH: fixed host
IH: intermediate host
WH: wireless host

Heterogeneous wired/wireless network



- Ethernet link between FH and IH (router) is full-duplex with a data rate of 100 Mb/s
- IEEE 802.11b WLAN between IH (router) and WH (client) is half-duplex with data rate of 11 Mb/s

Roadmap

- Introduction
- Background and related work:
  - overview of TCP
  - M-TCP for mobile cellular networks
- M-TCP+ algorithm for wired/wireless network:
  - description
  - implementation
- Performance evaluation:
  - simulation scenarios and parameters
  - simulation results
- Conclusions

M-TCP+ implementation

- Extension to TCP NewReno in OPNET simulation tool
- Modification required at IH
- TCP port numbers employed to differentiate between TCP connections
- IH receives TCP data segment from FH:
  - checks TCP data segment's sequence number *seq_no*:
    - stores the number
    - keeps copy of TCP data segment in FIFO queue
  - forwards TCP data segment to WH

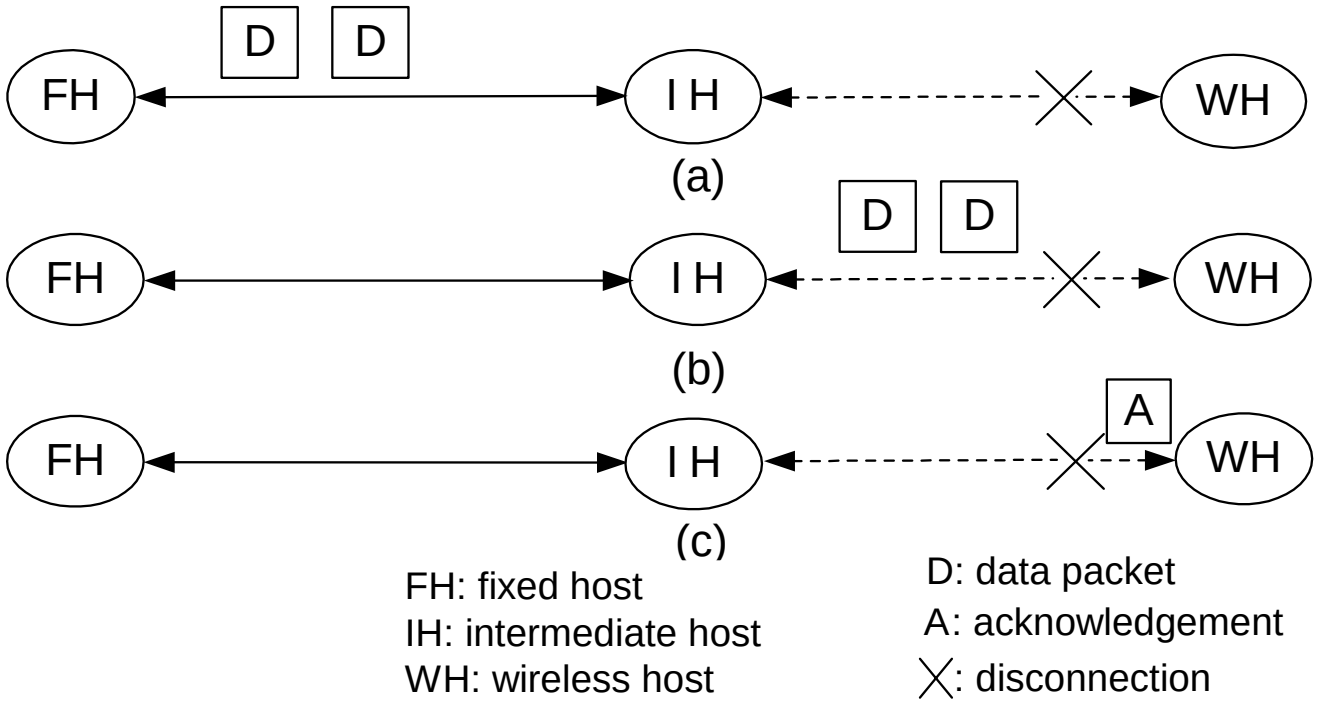
FH: fixed host
IH: intermediate host
WH: wireless host
FIFO: first-in-first-out

M-TCP+ implementation

- IH receives ACK segment from WH:
  - checks ACK number *ack_no* of ACK segment:
    - compares *ack_no* with *seq_no*
    - purges any segment in its FIFO queue preceding the newly acknowledged segment
  - forwards ACK segment to FH
- M-TCP+ network contains no SH:
  - IH forwards all ACKs as received from WH
  - *ack_no* of ACK forwarded to FH is not reduced by one

FH: fixed host
IH: intermediate host
WH: wireless host
SH: supervisor host

Cases of WH disconnection



- Unacknowledged data packets in flight:
  - wired link
  - wireless link
- ACK packet in flight in the wireless link

M-TCP+ implementation: disconnection

- IH:
  - receives disconnection feedback signal from WH
  - sends ACK of the last received TCP segment with *rwnd* set to zero to FH
  - acknowledges all outstanding data segments
- FH:
  - enters into persist state
  - refrains from sending additional data segments
  - resumes sending data with *cwnd* value previous to disconnection when *rwnd* is non-zero

FH: fixed host
IH: intermediate host
WH: wireless host

Roadmap

- Introduction
- Background and related work:
  - overview of TCP
  - M-TCP for mobile cellular networks
- M-TCP+ algorithm for wired/wireless network:
  - description
  - implementation
- Performance evaluation:
  - simulation scenarios and parameters
  - simulation results
- Conclusions

Simulation scenarios and parameters

- Simulation scenarios:
  - absence of disconnections and congestion
  - presence of disconnections only
  - presence of both disconnections and congestion
- Parameters:
  - 100 MB FTP file
  - 1,000 s simulated time
  - disconnection periods:
    - frequent: 5 s, 10 s, and 30 s in one-minute cycle
    - lengthy: 180 s in five-minute cycle
  - TCP parameters with delayed ACK enabled and with delayed ACK disabled

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	0.2 s
Maximum ACK segment	2
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

Simulation parameters

- TCP variants:
  - M-TCP+
  - M-TCP
  - TCP NewReno
  - TCP SACK
- Performance metrics:
  - FTP download response time: elapsed between the instant the FTP request is sent and the complete file is downloaded
  - TCP goodput: indicated by the received segment sequence number
  - retransmission ratio calculated as:

$$retransmission\ ratio = \frac{\#\ retransmitted\ segments}{total\ number\ of\ segments - retransmitted\ segments} \times 100$$

Roadmap

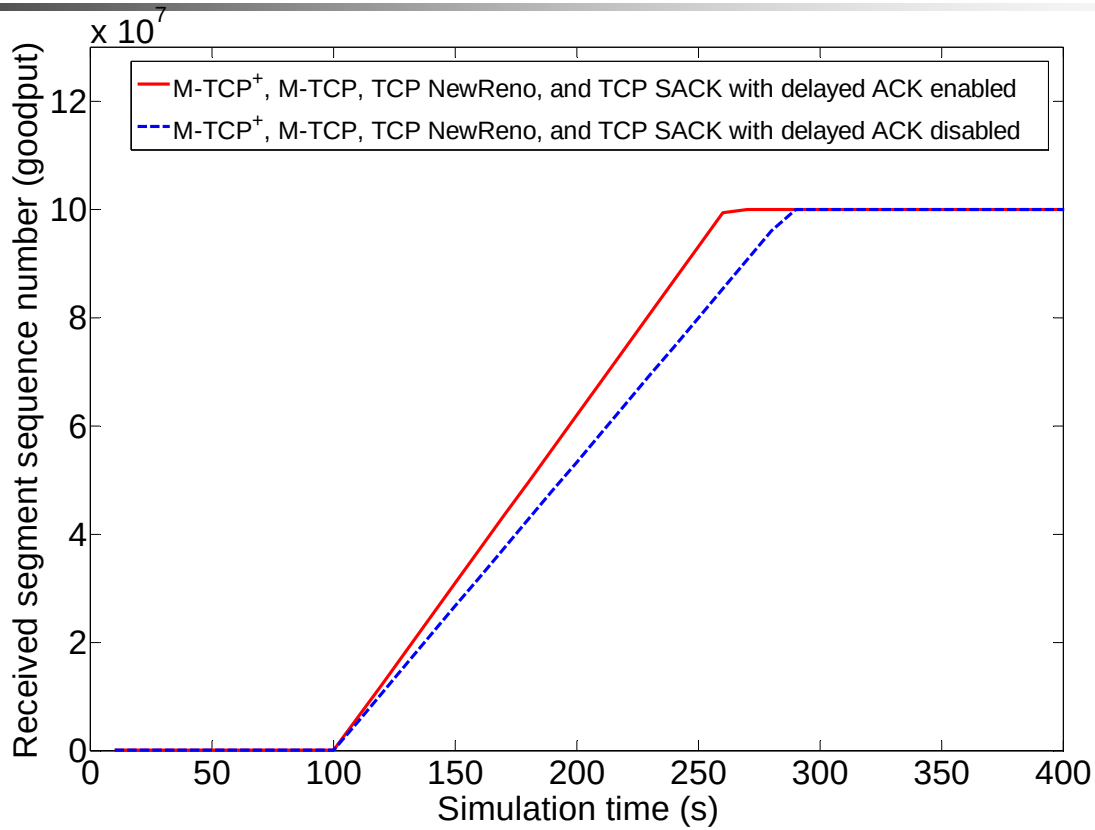
- Introduction
- Background and related work:
  - overview of TCP
  - M-TCP for mobile cellular networks
- M-TCP+ algorithm for wired/wireless network:
  - description
  - implementation
- Performance evaluation:
  - simulation scenarios and parameters
  - simulation results: absence of disconnections and congestion
- Conclusions

Absence of disconnections and congestion: download response time

Delayed ACK option	TCP Variant			
	M-TCP+	M-TCP	TCP NewReno	TCP SACK
Enabled	261.0	261.1	261.0	261.0
Disabled	287.4	287.8	287.4	287.4

- Download response time is measured in seconds
- The four TCP variants exhibit similar performance in all cases
- M-TCP+ algorithm does not introduce additional processing delay to the file download
- Download response times for the cases with delayed ACK enabled are lower than for the cases with delayed ACK disabled

Absence of disconnections and congestion: TCP goodput



- The four TCP variants have similar goodput performance in all cases
- TCP variants in cases with delayed ACK enabled show higher goodput than TCP variants in cases with delayed ACK disabled
- Received segment sequence number is used an indicator of goodput

Roadmap

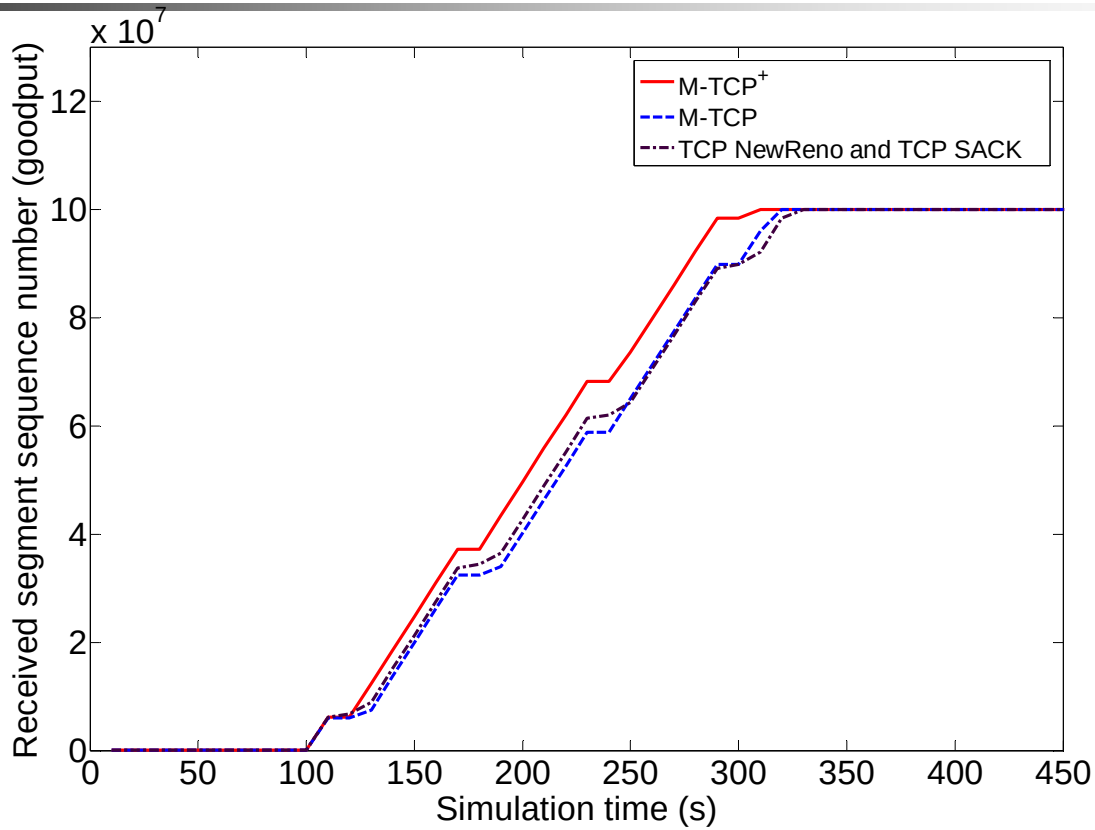
- Introduction
- Background and related work:
  - overview of TCP
  - M-TCP for mobile cellular networks
- M-TCP+ algorithm for wired/wireless network:
  - description
  - implementation
- Performance evaluation:
  - simulation scenarios and parameters
  - simulation results: presence of disconnections only
- Conclusions

Presence of disconnections only: download response time

Delayed ACK option	TCP variant	Disconnection period			
		5 s	10 s	30 s	180 s
Enabled	M-TCP+	277.9	302.4	435.6	622.4
	M-TCP	278.7	316.2	484.5	0.0
	NewReno	283.0	322.5	439.4	0.0
	SACK	283.0	322.5	439.4	0.0
Disabled	M-TCP+	311.4	327.9	489.3	647.8
	M-TCP	311.9	328.8	489.3	649.4
	NewReno	316.8	384.4	497.8	0.0
	SACK	316.8	384.4	497.8	0.0

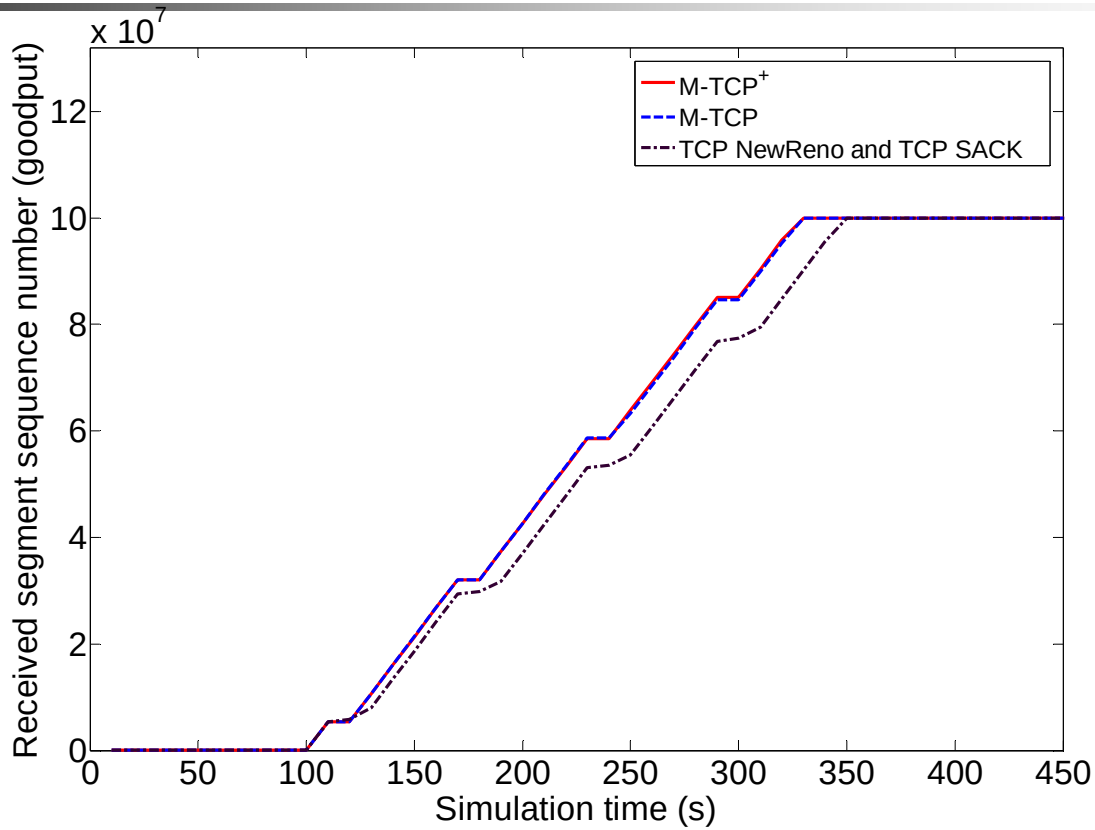
- The M-TCP+ algorithm exhibits the lowest download response time in both cases for all disconnection periods
- M-TCP+ algorithm reduces download response time by 2%–15%

Presence of disconnections only with delayed ACK enabled: TCP goodput



- Delayed ACK is enabled and the disconnection period is 10 s
- M-TCP+ outperforms M-TCP, TCP NewReno, and TCP SACK and improves goodput by ~ 12%

Presence of disconnections only with delayed ACK disabled: TCP goodput



- Delayed ACK is disabled and the disconnection period is 10 s
- M-TCP+ and M-TCP:
  - outperform TCP NewReno and TCP SACK
  - improve goodput by ~ 10%

Presence of disconnections only: retransmission ratio

Delayed ACK option	TCP variant	Disconnection period			
		5 s	10 s	30 s	180 s
Enabled	M-TCP+	0.4	0.5	1.2	0.3
	M-TCP	2.6	5.0	9.2	8.4
	NewReno	3.5	5.3	7.7	6.8
	SACK	3.5	5.3	7.7	6.8
Disabled	M-TCP+	0.6	0.6	1.3	0.3
	M-TCP	0.6	0.6	1.3	0.3
	NewReno	4.7	5.3	9.2	7.8
	SACK	4.7	5.3	9.2	7.8

- M-TCP+ exhibits the best performance with delayed ACK enabled
- M-TCP+ and M-TCP show similar performance with delayed ACK disabled

Roadmap

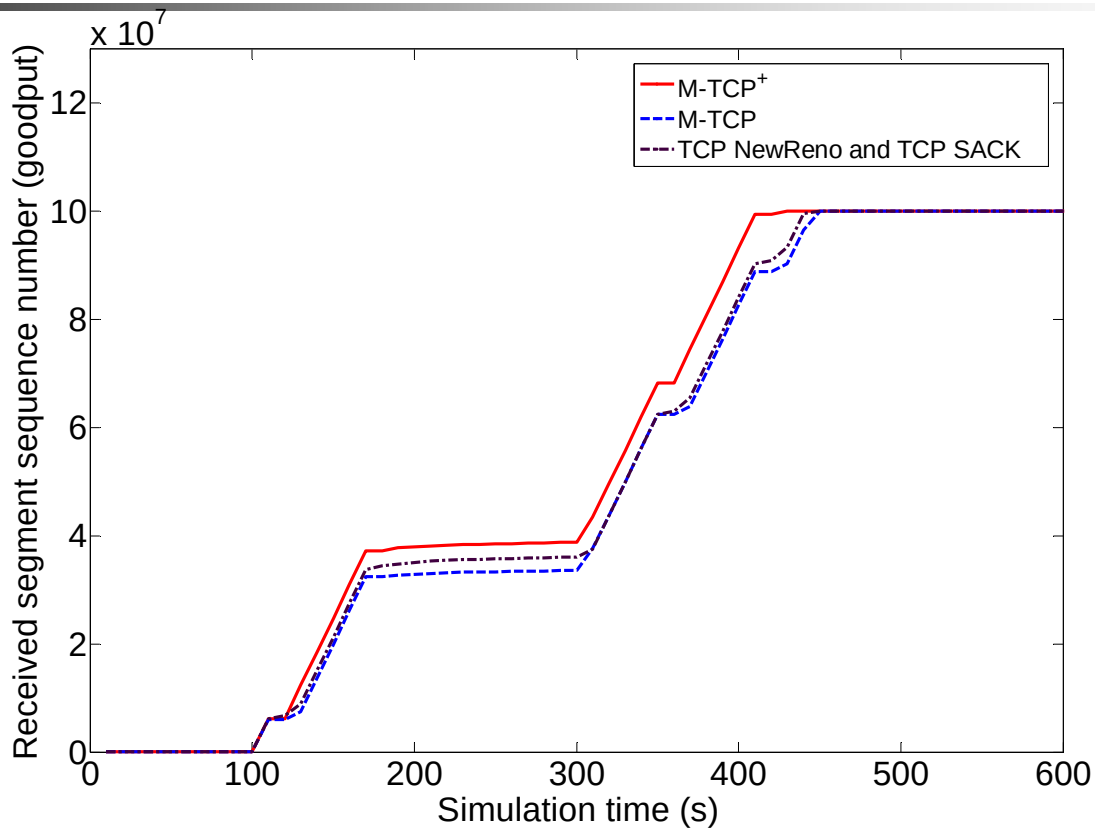
- Introduction
- Background and related work:
  - overview of TCP
  - M-TCP for mobile cellular networks
- M-TCP+ algorithm for wired/wireless network:
  - description
  - implementation
- Performance evaluation:
  - simulation scenarios and parameters
  - simulation results: presence of both disconnections and congestion
- Conclusions

Presence of both disconnections and congestion: download response time

Delayed ACK option	TCP Variant			
	M-TCP+	M-TCP	TCP NewReno	TCP SACK
Enabled	421.0	445.1	447.0	447.0
Disabled	445.5	447.4	464.3	464.3

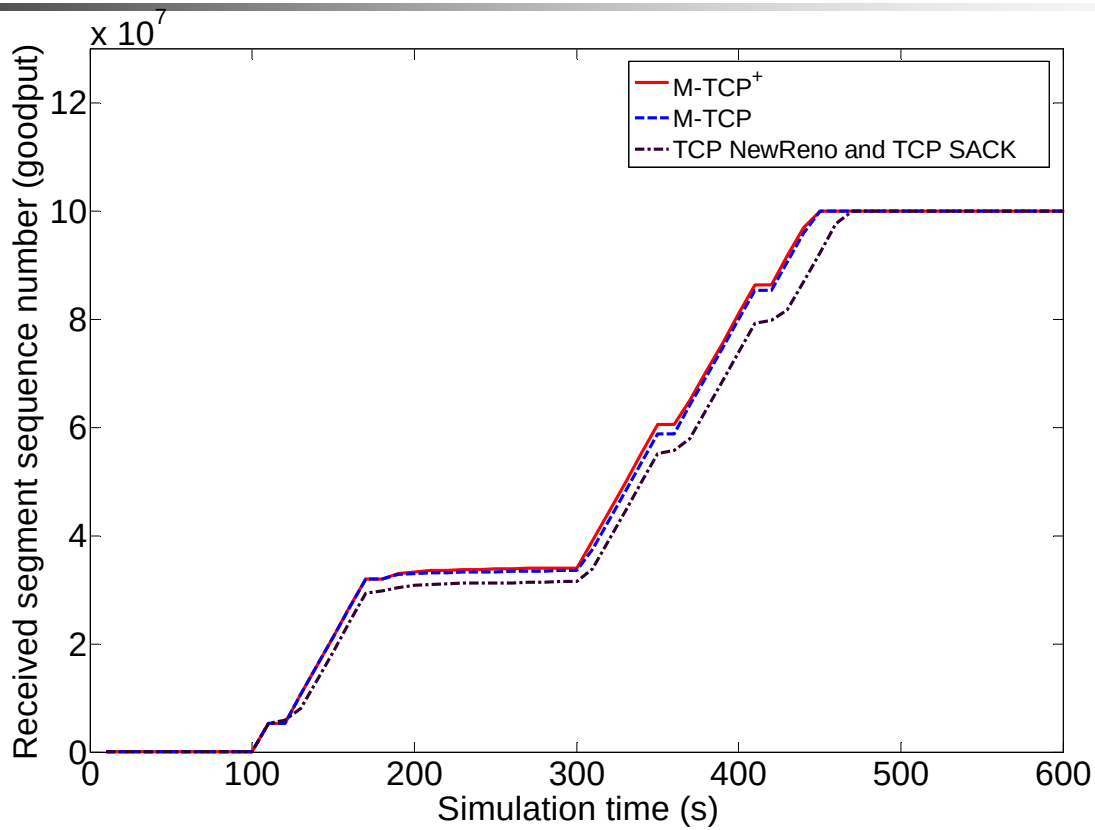
- Disconnection period is 10 s
- M-TCP+ outperforms M-TCP, TCP NewReno, and TCP SACK in the cases with delayed ACK enabled
- Download response times are higher for the same disconnection period (10 s) in both cases, with delayed ACK enabled and disabled
- Higher download response times indicate the impact of congestion

Presence of both disconnections and congestion with delayed ACK enabled: TCP goodput



- Delayed ACK option is enabled and the disconnection period is 10 s
- M-TCP+ outperforms M-TCP, TCP NewReno, and TCP SACK and improves goodput by ~ 8%

Presence of both disconnections and congestion with delayed ACK disabled: TCP goodput



- Delayed ACK is disabled and the disconnection period is 10 s
- M-TCP+ and M-TCP:
  - outperform TCP NewReno and TCP SACK
  - improve goodput by ~ 12%

Presence of both disconnections and congestion: retransmission ratio

Delayed ACK option	TCP Variant			
	M-TCP+	M-TCP	TCP NewReno	TCP SACK
Enabled	0.9	7.0	7.6	7.6
Disabled	0.7	1.6	7.6	7.6

- M-TCP+ shows lower retransmission ratio than M-TCP, TCP NewReno, and TCP SACK in the cases with delayed ACK enabled
- With delayed ACK disabled M-TCP+ and M-TCP:
  - have similar reduction of ~ 7% in terms of retransmission ratios
  - outperform TCP NewReno and TCP SACK
- Additional segment losses in the presence of congestion

Roadmap

- Introduction
- Background and related work:
  - overview of TCP
  - M-TCP for mobile cellular networks
- M-TCP+ algorithm for wired/wireless network:
  - description
  - implementation
- Performance evaluation:
  - simulation scenarios and parameters
  - simulation results
- Conclusions

Conclusions

- In the absence of disconnection and congestion, M-TCP+:
  - performs comparably to M-TCP, TCP NewReno, and TCP SACK for both cases with delayed ACK enabled and disabled
  - does not increase processing delay of the file download response time
- In the presence of disconnections only and both disconnections and congestion, M-TCP+ performs:
  - better than M-TCP, TCP NewReno, and TCP SACK with delayed ACK enabled
  - similar to M-TCP with delayed ACK disabled

Conclusions

- Deployment of M-TCP+ in existing networks requires:
  - modifications only at intermediate hosts (IHs) such as base stations, routers, and gateways
  - additional buffers at an IH to store packets that will be retransmitted to a wireless host (WH) when it reconnects to the network after disconnections
- M-TCP+ maintains the end-to-end semantics of TCP in the absence of disconnections

References

- C. Williamson, "Internet traffic measurement," *IEEE Internet Comput.,* vol. 5, no. 6, pp. 70–74, Nov./Dec. 2001.
- W. Stevens, *TCP Illustrated Volume 1: The Protocols*. Reading, MA: Addison-Wesley,1994.
- V. Jacobson, "Congestion avoidance and control," in *Proc. ACM SIGCOMM Symp. on Commun. Archit. and Protocols*, Stanford, CA, USA, Aug. 1988, pp. 314–329.
- M. Allman, V. Paxson, and W. Stevens, "TCP congestion control," *RFC 2581*, Apr. 1999.
- V. Paxson, "Automated packet trace analysis of TCP implementations," in *Proc. ACM SIGCOMM Conf. on Appl., Technol., Archit., and Protocols for Comput. Commun.*, Cannes, France, Sept. 1997, pp. 167–179.
- K. Brown and S. Singh, "M-TCP: TCP for mobile cellular networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 27, no. 5, pp. 19–42, Oct. 1997.
- W. G. Zeng, M. Zhan, Z. Lin, and Lj. Trajković, "Performance evaluation of M-TCP over wireless links with periodic disconnections," *OPNETWORK 2003*, Washington, DC, Aug. 2003.
- D. Nagamalai, B. C. Dhinnakaran, B.-S. Choi, and J.-K. Lee, "V-TCP: a novel TCP enhancement technique," in *Networking-ICN, Lecture Notes in Comput. Science.* Springer, Berlin: vol. 3421, pp. 125–132, Mar. 2005.
- T. Goff, J. Moronski, and V. Gupta, "Freeze-TCP: a true end-to-end TCP enhancement mechanism for mobile environments," in *Proc. IEEE INFOCOM*, Tel-Aviv, Israel, Mar. 2000, pp. 1537–1545.
- OPNET Modeler software [Online]. Available: http://www.opnet.com/products/modeler/home.html.