University of Helsinki - Department of Computer Science

HTTP traffic performance in wireless environment

Jarno Saarto

jarno.saarto@helsinki.fi

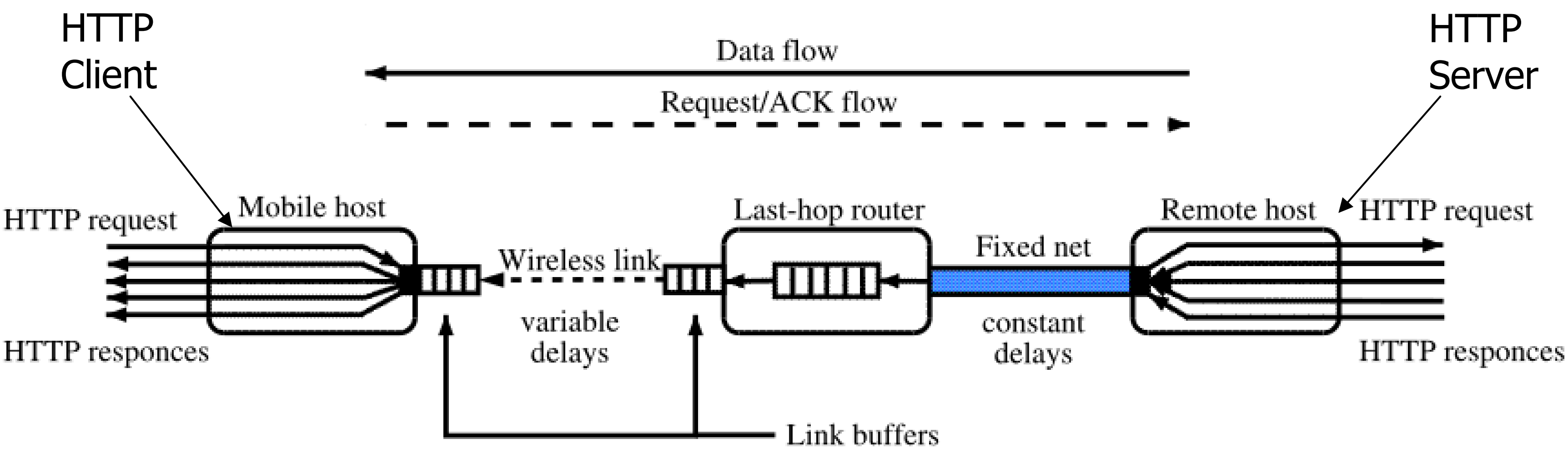
Objectives and methods:

- Objectives
  - Evaluate HTTP 1.0 traffic performance in wireless environment
  - Compare different TCP enhancements
  - Compare different types of link
- Methods:
  - The wireless link and last-hop router were emulated with Seawind
    - Parameters were chosen to mimic roughly a GPRS / UMTS type wireless link
  - HTTP Traffic was generated with a HTTP traffic generator tool

Link parameters:

Parameter	Value
Router buffer	20 packets
Queue drop policy	Tail-drop
Link send buffer	9600 bytes
Link receive buffer	9600 bytes
Propagation delay	300 ms
MTU	576 bytes
Bandwidth	64000 bps

Seawind parameters used in tests



Wireless network properties:

Parameter	**Good state**	**Bad state**
Error rate	0 %	63 %
Length distribution	Exponential	Uniform
Min value	0 ms	200 ms
Max value	9 s	1.9
Mean	6 s	-

Properties of the good and bad states

Link type	Max number of retransmissions
Optimal	-
Lossy link with low ARQ persistency	1
Lossy link with medium ARQ persistency	2
Lossy link with high ARQ persistency	3

Link types

Workload:

Object type	Small size	Medium size	Large size
Main object	6288 bytes	12576 bytes	71788 bytes
In-line object	2096	8384	-

Label	Main object size	In-line object size	# of in-line objects
`s+2s`	**Small**	**Small**	**2**
`s+2m`	**Small**	**Medium**	**2**
`s+8m`	**Small**	**Medium**	**8**
`m+2m`	**Medium**	**Medium**	**2**
`m+8m`	**Medium**	**Medium**	**8**
`l+8m`	**Large**	**Medium**	**8**

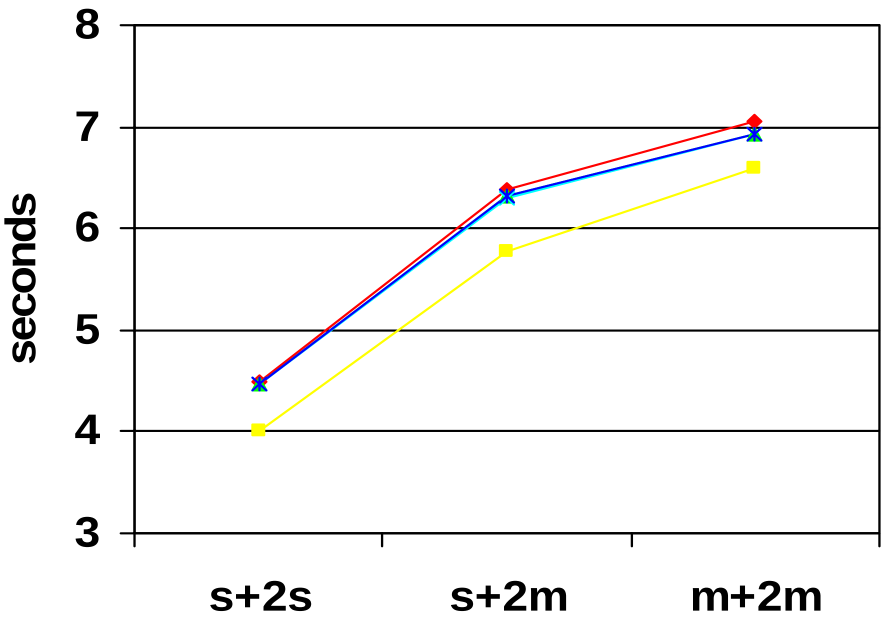
Labels: main object size + *n* in-line object size, where *n* is the number of in-line objects

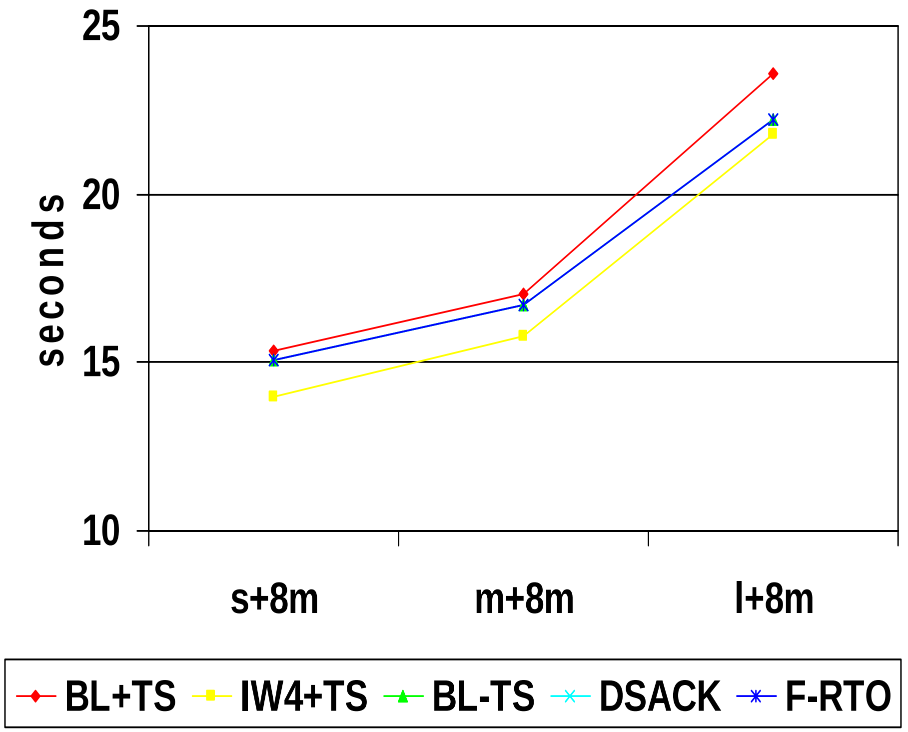
TCP variants:

Baseline TCP	Increased Initial Window: 2 segments Limited Transmit SACK TCP Timestamps (with and without)
Enhanced TCP	Increased Initial Window: 4 segments (IW4) D-SACK F-RTO

TCP enhancements included in baseline and enhanced TCP implementations

Results: optimal link





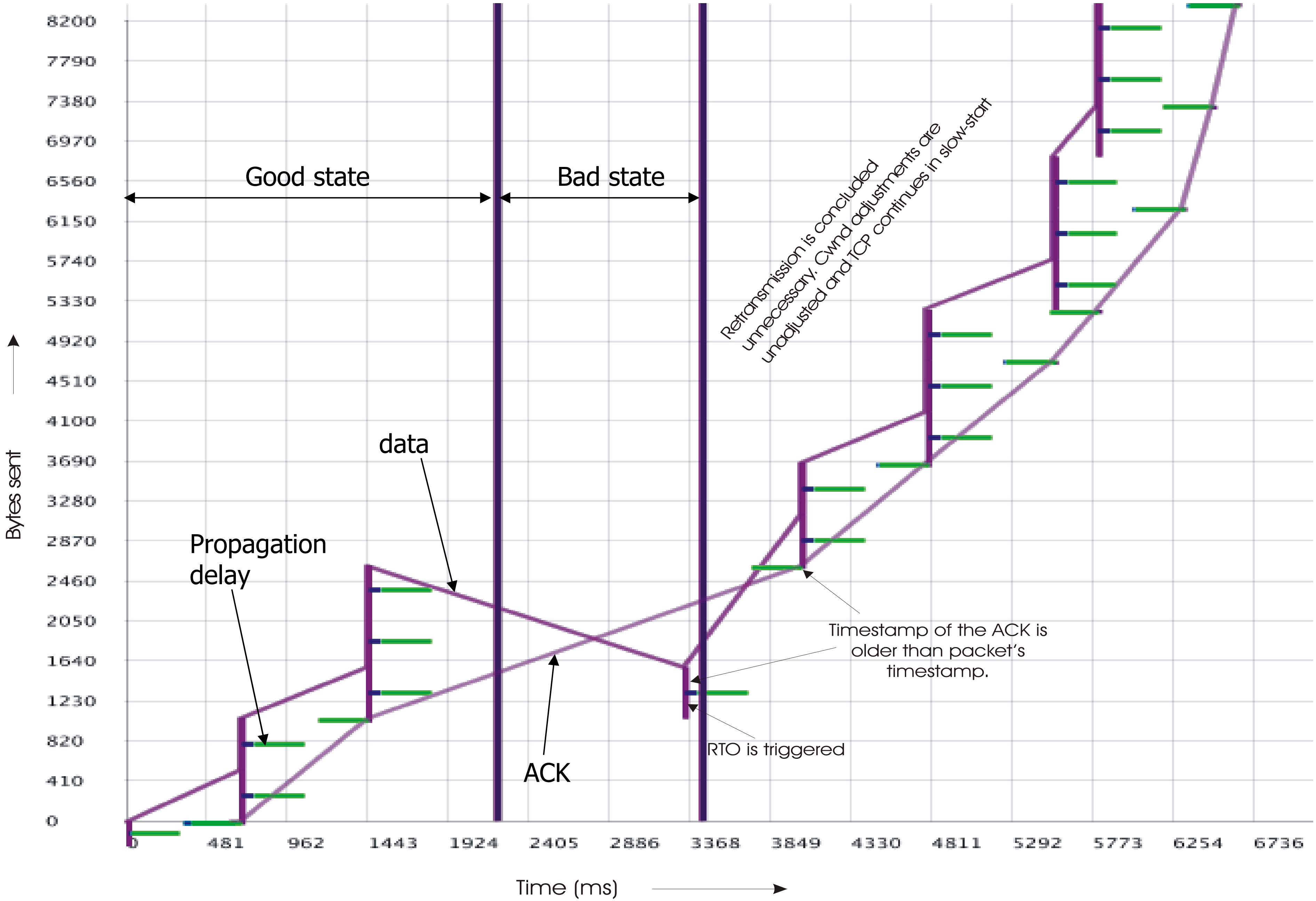
Results - lossy link with low ARQ persistency

WL	s+2s	s+2m	s+8m	m+2m	m+8m	l+8m
Response time (s)	6.45	8.82	20.13	10.73	22.88	34.98
Med th. (m) B/s	1208	1256	1271	1614	1559	2231
Slowest th. (i) B/s	619	1326	786	1070	681	530
Fastest th. (i) B/s	693	1470	1695	1400	1715	1482
Drops (c) median	0	0	0	0	0	0
Drops (e) median	0	0	2	0	2	6
Σ maximum drops	11	14	47	19	49	89

Summary of lossy link with low ARQ persistency
test results of Baseline TCP with TCP Timestamps

Low ARQ persistency: Baseline TCP with TCP Timestamps

- Packet losses increased response times
  - By delaying the TCP handshake or the initial HTTP request
  - By causing retransmission timeouts
  - By causing fast retransmissions of packets
    - TCP continues in congestion avoidance



Low ARQ persistency: Baseline TCP with TCP Timestamps
TCP transfer of a medium in-line object

Differences in response times with lossy link with low ARQ persistency

Workload	BL+TS	IW4+TS	BL-TS	F-RTO
s+2s	6.45 s	-28.8%	+3.9%	+2.2%
s+2m	8.82 s	-3.2%	+2.0%	+15.4%
s+8m	20.13 s	-5.8%	+8.5%	+1.7%
m+2m	10.73 s	-19.9%	-0.6%	-14.9%
m+8m	22.88 s	-11.8%	-8.6%	-5.8%
l+8m	34.98 s	-2.3%	-8.1%	+5.7%

Differences in response times with different TCP variants

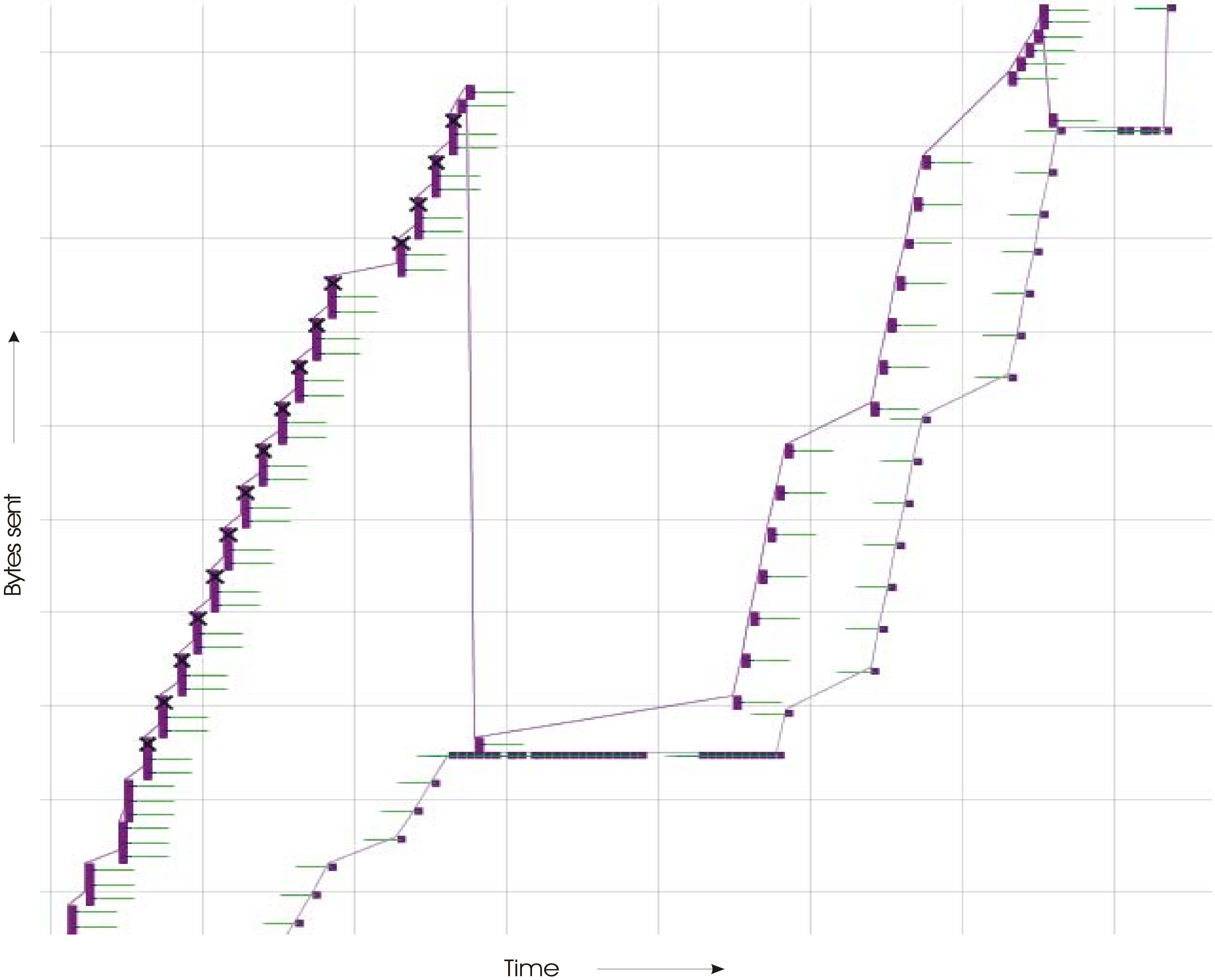
Low ARQ persistency: TCP behaviors

- Baseline TCP without TCP Timestamps
  - Similar TCP behaviors to baseline TCP with TCP Timestamps
  - Recovering from packet losses was efficient
  - Differences resulted from different network conditions
- F-RTO did not generally affect performance, because there was generally no spurious retransmission timeouts
- Differences were due to different network conditions

Results - lossy link with medium ARQ persistency:

WL	s+2s	s+2m	s+8m	m+2m	m+8m	l+8m
Response time (s)	5.90	8.44	20.16	8.61	20.57	29.72
Med th. (m) B/s	1196	1169	1113	1804	1804	2977
Slowest th. (i) B/s	575	1438	1027	1315	969	516
Fastest th. (i) B/s	669	1472	1606	1365	1554	1311
Drops (c) median	0	0	0	0	0	15
Drops (e) median	0	0	0	0	0	0
Σ maximum drops	7	7	23	8	20	53

Summary of lossy link with medium ARQ persistency test results of Baseline TCP with TCP Timestamps



Lossy link with medium ARQ persistency:

TCP Transfer of a large main object

Differences in response times with lossy link with medium ARQ persistency

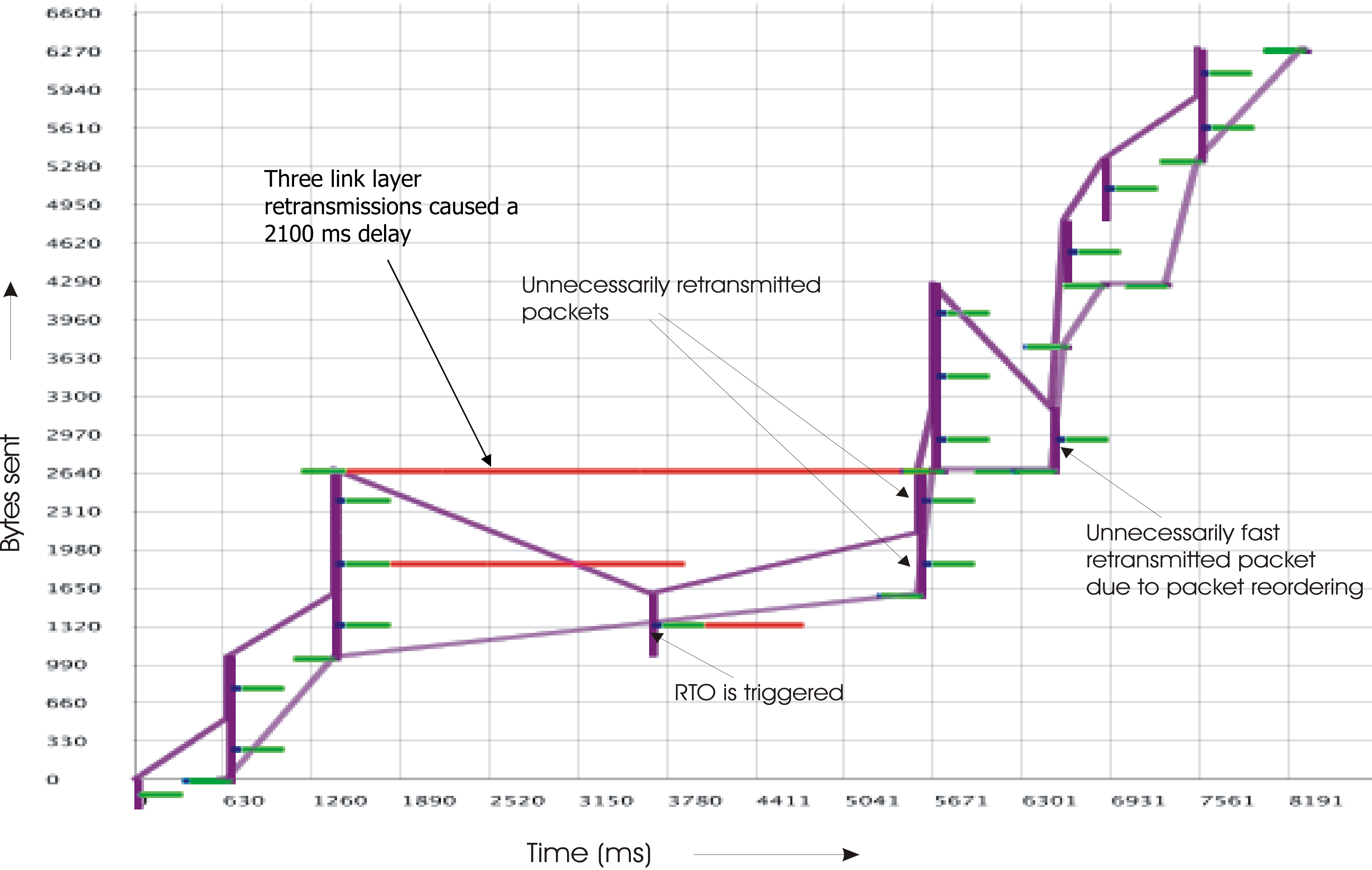
Workload	BL+TS	IW4+TS
s+2s	5.90 s	-18.6%
s+2m	8.44 s	-18.6%
s+8m	20.16 s	-15.0%
m+2m	8.61 s	-15.9%
m+8m	20.57 s	-8.3%
l+8m	29.72 s	-10.9%

Differences in response times with different TCP variants

Results - lossy link with high ARQ persistency:

WL	s+2s	s+2m	s+8m	m+2m	m+8m	l+8m
Response time (s)	5.87	7.72	19.68	9.29	21.06	30.83
Med th. (m) B/s	1334	1182	1050	1649	1526	2932
Slowest th. (i) B/s	730	1476	1034	1262	982	540
Fastest th. (i) B/s	796	1618	1608	1317	1579	1382
Drops (c) median	0	0	0	0	0	17
Drops (e) median	0	0	0	0	0	0
Σ maximum drops	2	0	5	3	4	46

Summary of link with high ARQ persistency test results of Baseline TCP with TCP Timestamps



Lossy link with high ARQ persistency: TCP transfer of a small main object with BL-TS

Differences in response times with link with high ARQ persistency

Workload	BL-TS	IW4-TS	D-SACK	F-RTO	BL+TS
s+2s	6.34 s	-18.8%	-3.9%	-7.4%	-5.2%
s+2m	8.40 s	-12.4%	+0.1%	-3.4%	+7.1%
s+8m	20.79 s	-3.0%	-4.7%	-1.1%	+6.4%
m+2m	9.20 s	-10.9%	-6.5%	+5.2%	+6.0%
m+8m	21.91 s	-8.0%	-1.4%	-2.7%	+1.0%
l+8m	30.09 s	-6.8%	-1.3%	+3.3%	+3.0%

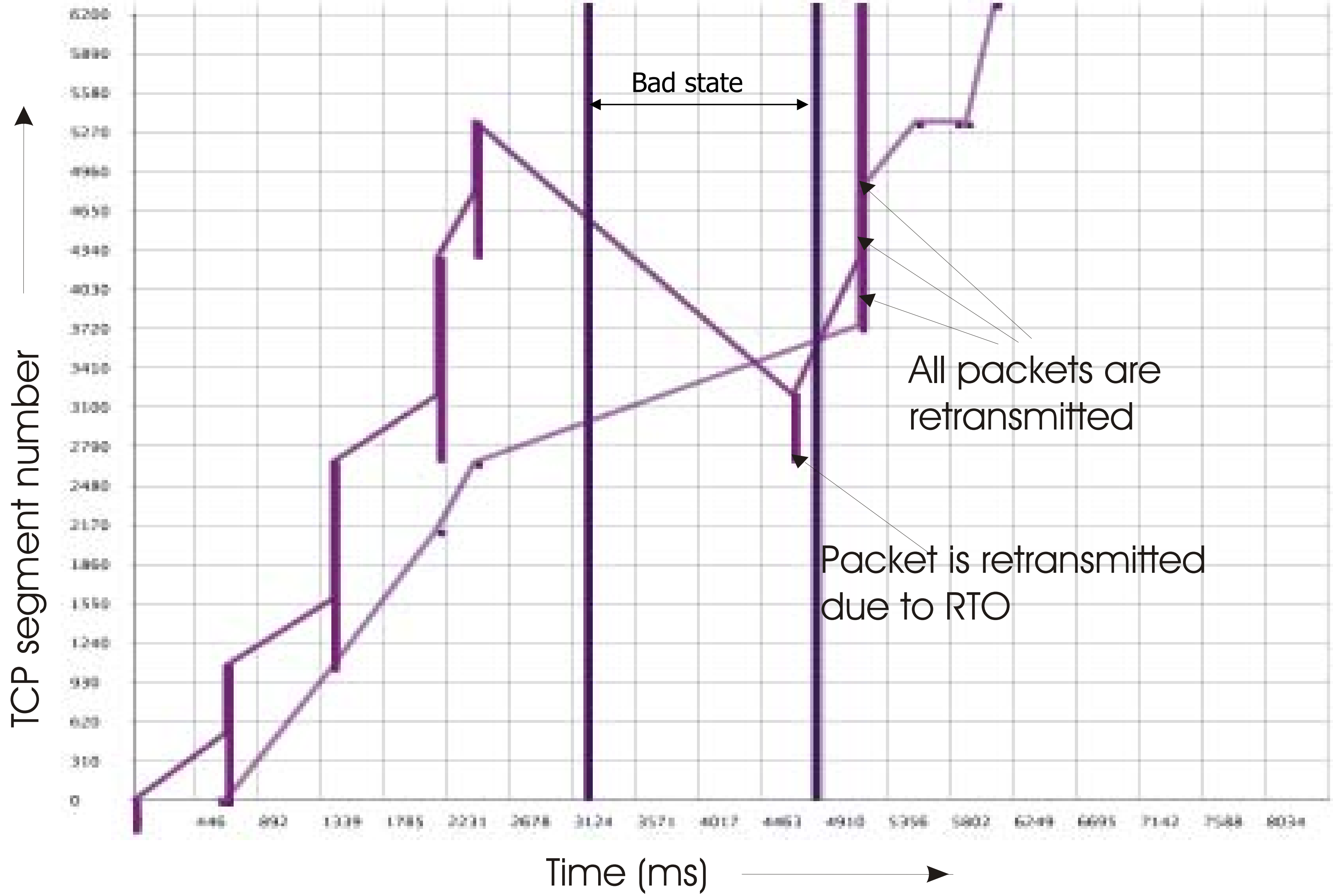
Differences in response times with different TCP variants

High ARQ persistency: Spurious retransmission timeouts

- The differences between response times and elapsed times of connections varied most with BL-TS
  - Inefficient recoveries from spurious retransmission timeouts
- D-SACK was not able to improve performance in our tests
  - Packets were unnecessarily retransmitted

High ARQ persistency: Spurious retransmission timeouts (cont)

- F-RTO recovered efficiently from majority of spurious RTOs
  - No packets other than the one whose RTO was triggered were retransmitted unnecessarily in those cases
  - In some cases F-RTO failed to conclude an RTO unnecessary
  - With WL l+8m the F-RTO algorithm was more conservative than the conventional algorithm



Lossy link with high ARQ persistency: TCP transfer of a small main object with D-SACK

High ARQ persistency: Spurious retransmission timeouts (cont)

- BL+TS was typically as efficient as F-RTO
  - It was able to detect all unnecessary retransmissions of packets
  - Best recoveries in cases of spurious retransmission timeouts
  - Stability was not as good as with F-RTO in some test cases
    - Workload l+8m

Conclusion

- Link with low ARQ persistency:
  - No spurious retransmission timeouts
  - A lot of error related losses
  - Response times were not stabile
  - IW 4 increased performance and stability

Conclusion (cont)

- Link with medium ARQ persistency:
  - Number of error related losses was decreased notably
  - No spurious retransmission timeouts
  - Response times were stabile
  - IW4 increased performance and stability in all tests

Conclusion (cont)

- Link with high ARQ persistency:
  - Nearly all error related losses were avoided
  - Spurious retransmission timeouts existed
  - Congestion at the last-hop router was increased
  - IW 4 improved the response times in general
  - Great variation on response times
    - F-RTO and TCP Timestamps ameliorated the situation
    - D-SACK was not able to improve the response times