An Approach to Flexible QoS Routing with Active Networks

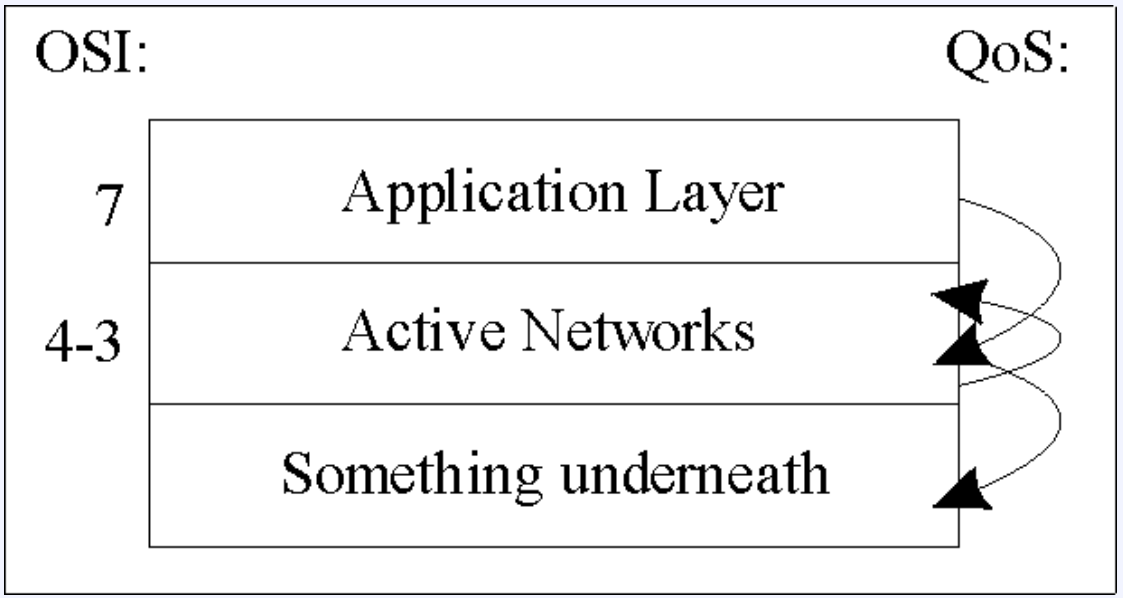
Michael Welzl Leopold Franzens University Innsbruck
Alfred Cihal Johannes Kepler University Linz
Max Mühlhäuser TU Darmstadt

Outline

- QoS with Active Networks: why / how?
- The mechanism
- Simulation: two scenarios
- Evaluation
- Conclusion / future work

Active Networks and QoS

- AN break with OSI tradition:

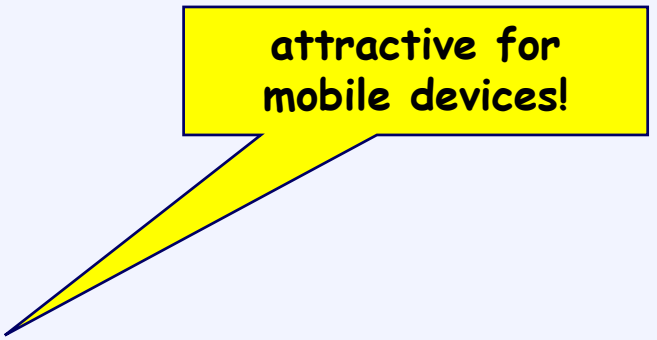


- quality scaling
- ACC, ... AQR
- ARP

- Advantage: flexibility
  (e.g., who will standardize proprietary quality scaling?)

Active Networks and MANETs

- MANETs:
  - Mobile Devices, also as routers
  - Memory and CPU restrictions
  - Flexible environment, changing topology



- Active Networks:
  - Lightweight software (app + net stack): only relevant code necessary
  - Flexible (customizable, programmable) networking infrastructure

- Active QoS Routing:
  - Active Networks -> source influences the network
  - Source routing often needed for MANET (changing topology)
  - Source routing <=> routing control + information at the source -> QoS

AQR: Requirements and Assumptions

- channel properties readable
  (we need QoS information - exception: delay)
- demand loading
  (we send a lot of packets ("capsules") - enhance scalability)
- link state routing table available?
  Yes: enhance scalability
- Assumption: flexibility
  - routing mechanism should be sender- or receiver-initiated
  - no standardized routing metrics

AQR: How it works

- Starting from the source, at regular intervals (e.g., 2 RTT)...
- Is this the destination?
- no...
  - Check QoS minimum requirements (drop?), update QoS data
  - Add this node's address to the visited list
  - Calculate all non-cyclic paths to dest. from the link state routing table (altered depth search alg.)
  - Send this code to all first hops on these paths (but not visited nodes)
- yes...
  - remember this packet's visited list

+ scalability

If there is none, simply flood

- Destination:
  - builds a list of paths from received packets for at least 1 RTT
  - reports the best path back to the source

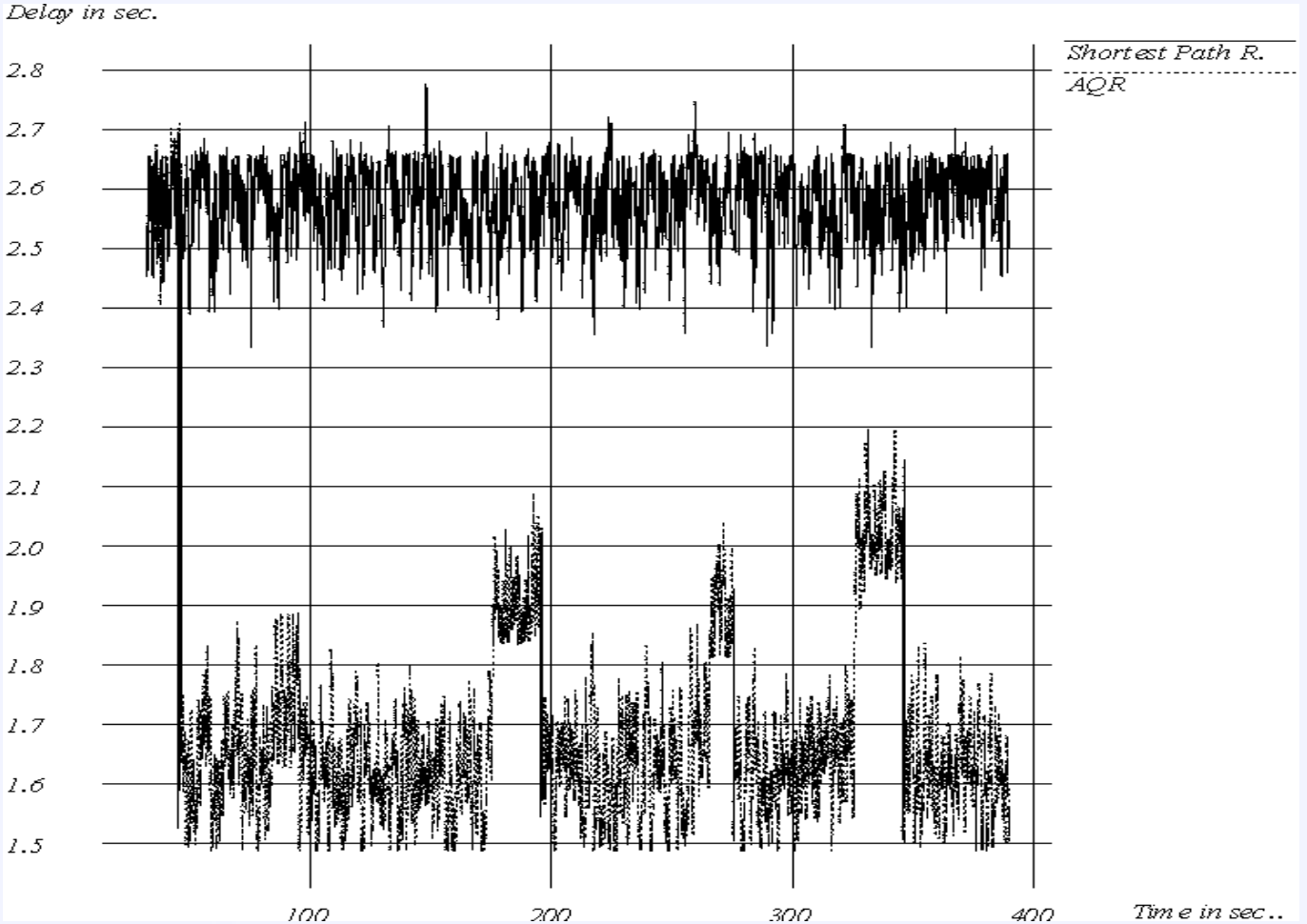
MANET Routing Classification

- Proactive routing
  - continuously make routing decisions
  - examples: DBF, DSDV, WRP, ..
- Reactive routing
  - determine routes when needed
  - examples: TORA, DSR, AODV, ABR, RDMAR, ..
- AQR:
  - reactive - but: continuously refresh to maintain QoS
  - may be combined with existing proactive routing
  - related mechanisms:
    - DSR
    - "On-Demand QoS Routing in Multihop Mobile Networks" [Lin, Infocom 2001]
  - main difference: QoS spec. flexibility through Active Networks

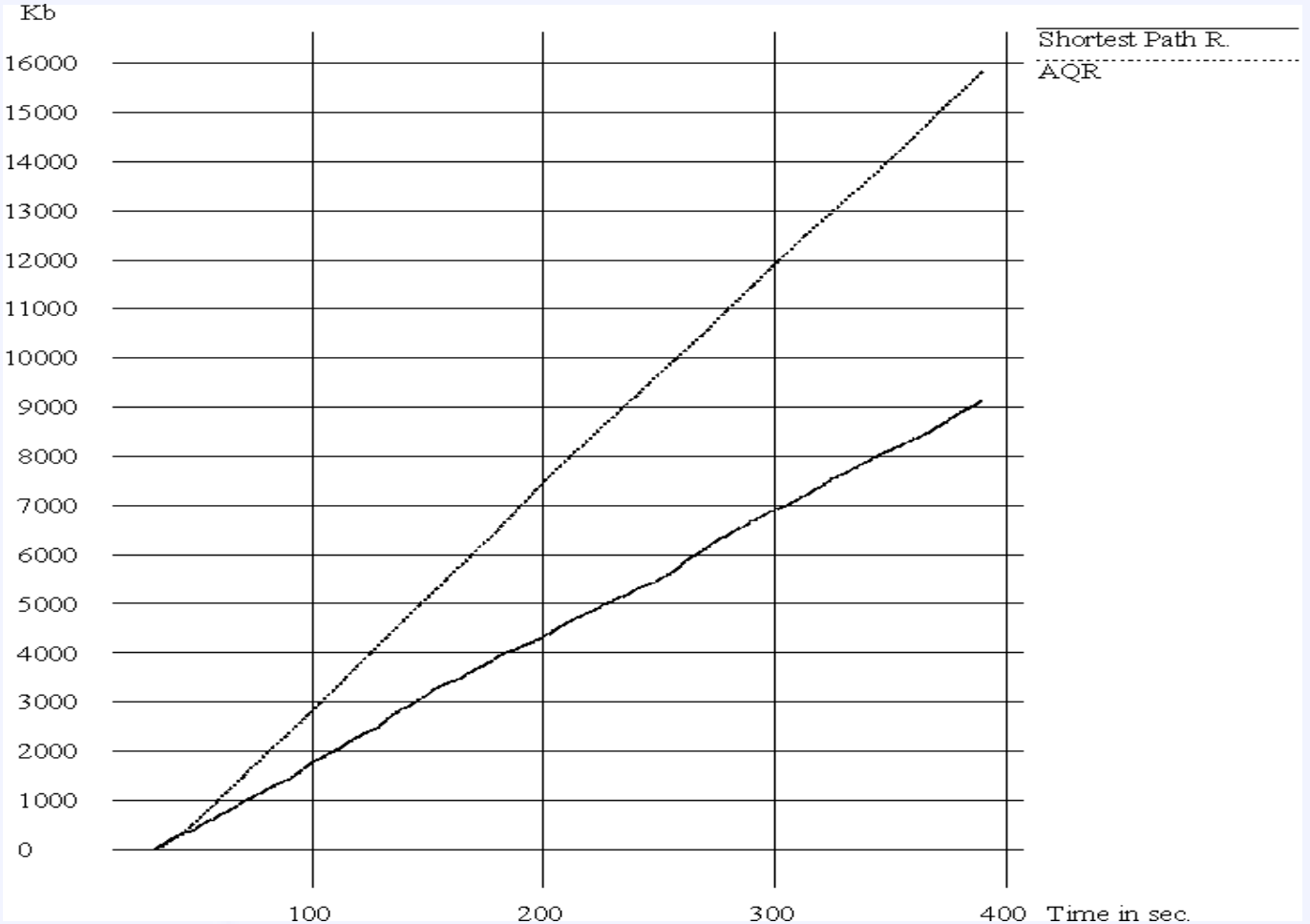
Simulation: Scenario 1

- ns-2 simulator
- Topology: One domain, 50 nodes
- Constant parameters:
  Drop delay threshhold: 5s
  Probing frequency: 10s
  (RTT: approx. 4,88 s)
- Background traffic: UDP, Pareto distrib. ON-OFF sources
  packet size: 500 bytes, burst time 100ms, idle time 150ms, rate 1000 kb/s
- QoS parameter: delay

Delay


Delay in sec.
2.8
2.7
2.6
2.5
2.4
2.3
2.2
2.1
2.0
1.9
1.8
1.7
1.6
1.5
100
200
300
400
Time in sec..
Shortest Path R.
AQR

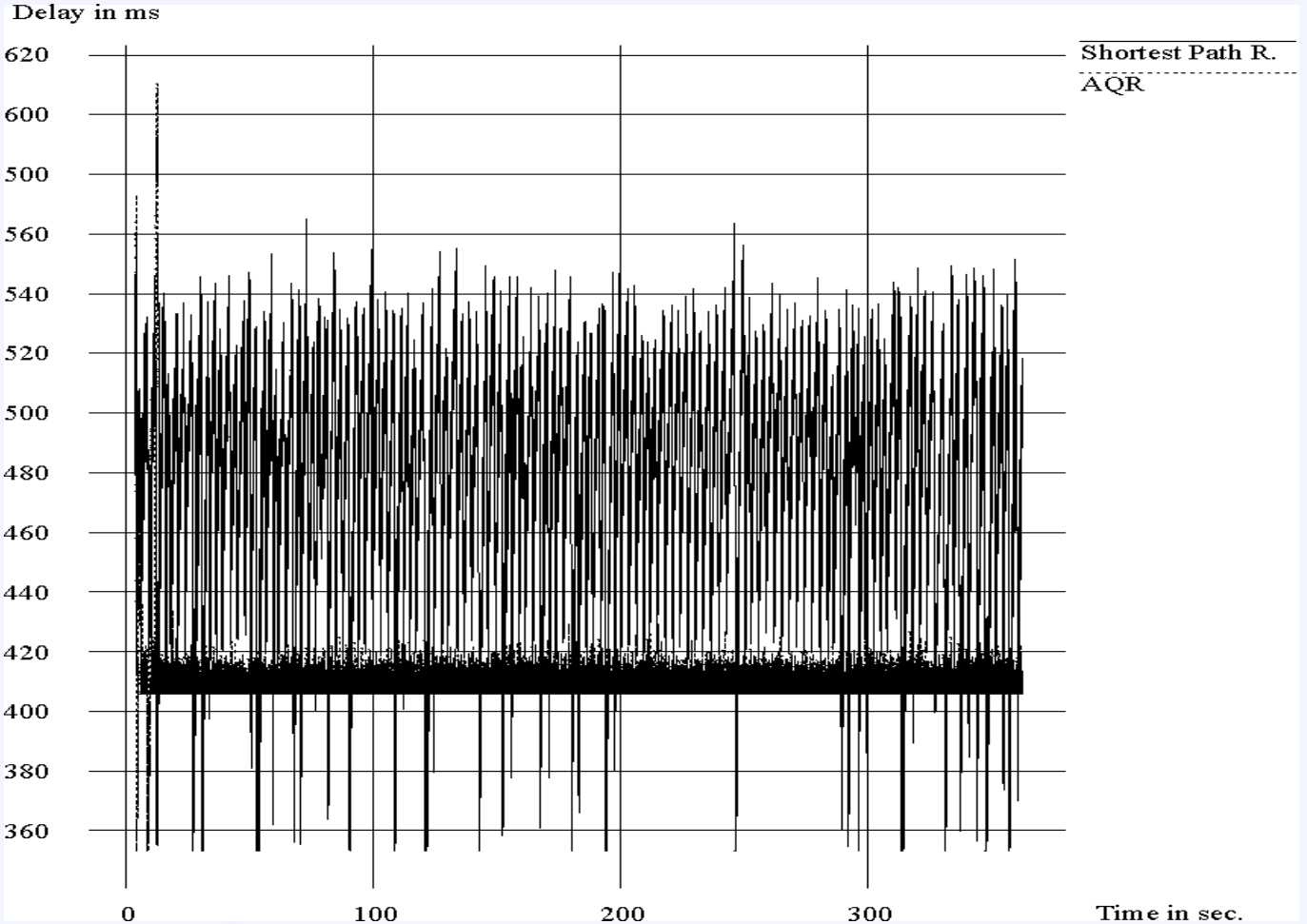
Throughput


Kb
16000
15000
14000
13000
12000
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
100
200
300
400
Time in sec.
Shortest Path R.
AQR

Scenario 2

- Topology: One domain, 15 nodes
- Constant parameters:
  Drop delay threshhold: 1.5s
  Probing frequency: 1.5s
  (RTT: approx. 776 ms)
- Background traffic: UDP / TCP mixture, Pareto distrib.
  ON-OFF sources, packet size: 300 bytes, burst time 130ms, idle time 200ms, rate 1400 kb/s
- QoS parameter: delay

Delay


Delay in ms
620
600
500
560
540
520
500
480
460
440
420
400
380
360
0
100
200
300
Time in sec.
Shortest Path R.
AQR

Simulation Results

- Scenario 1:

	AQR	*Shortest Path*
avg. latency	1696 ms	2574 ms
min. latency	1489 ms	2335 ms
max. latency	2712 ms	2779 ms
max. jitter	1224 ms	445 ms
throughput	43.03 kb/s	24.86 kb/s

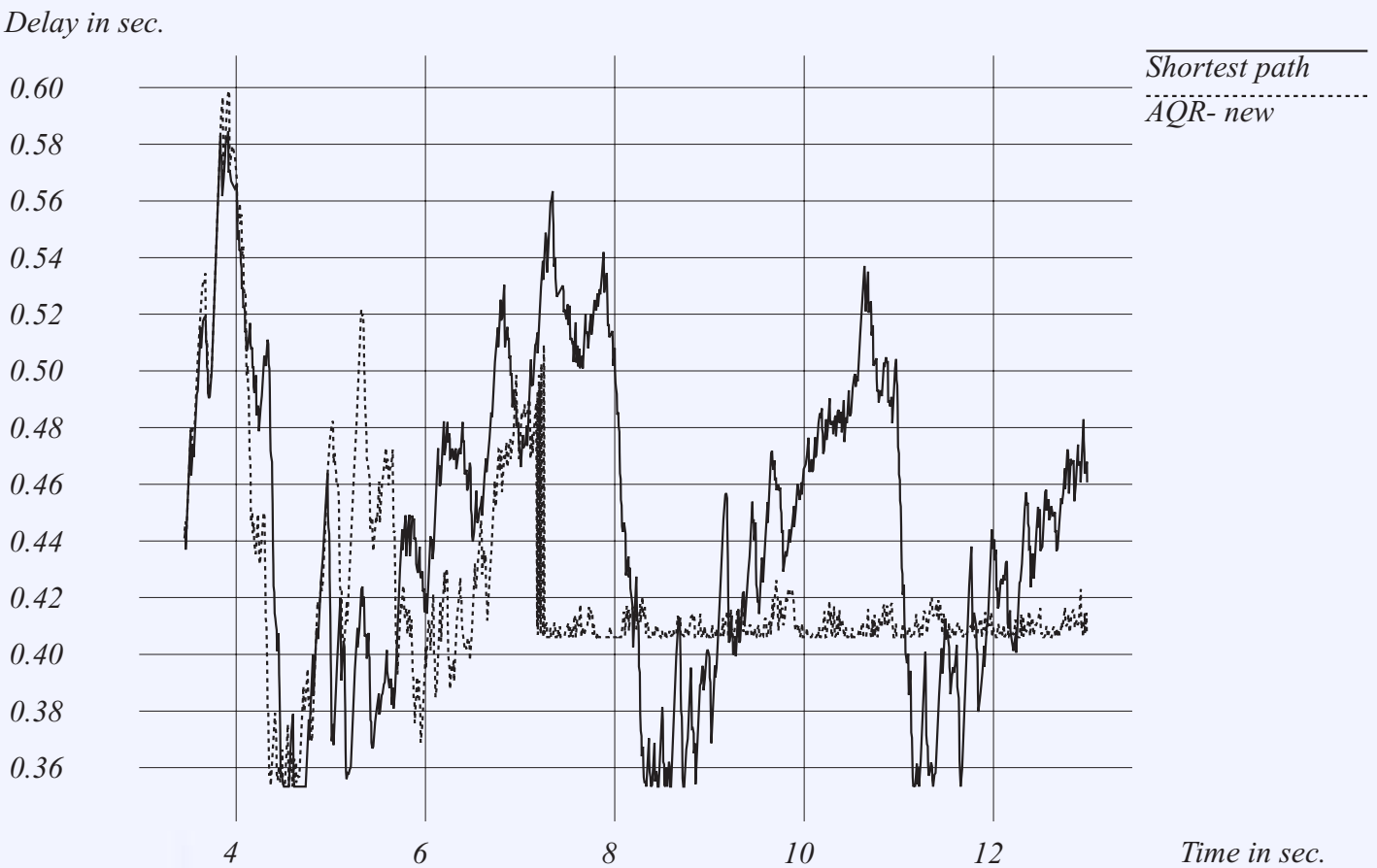
- Scenario 2:

	AQR	*Shortest Path*
avg. latency	410 ms	476 ms
min. latency	353 ms	353 ms
max. latency	574 ms	565 ms
max. jitter	220 ms	212 ms
throughput	48.76 kb/s	48.46 kb/s

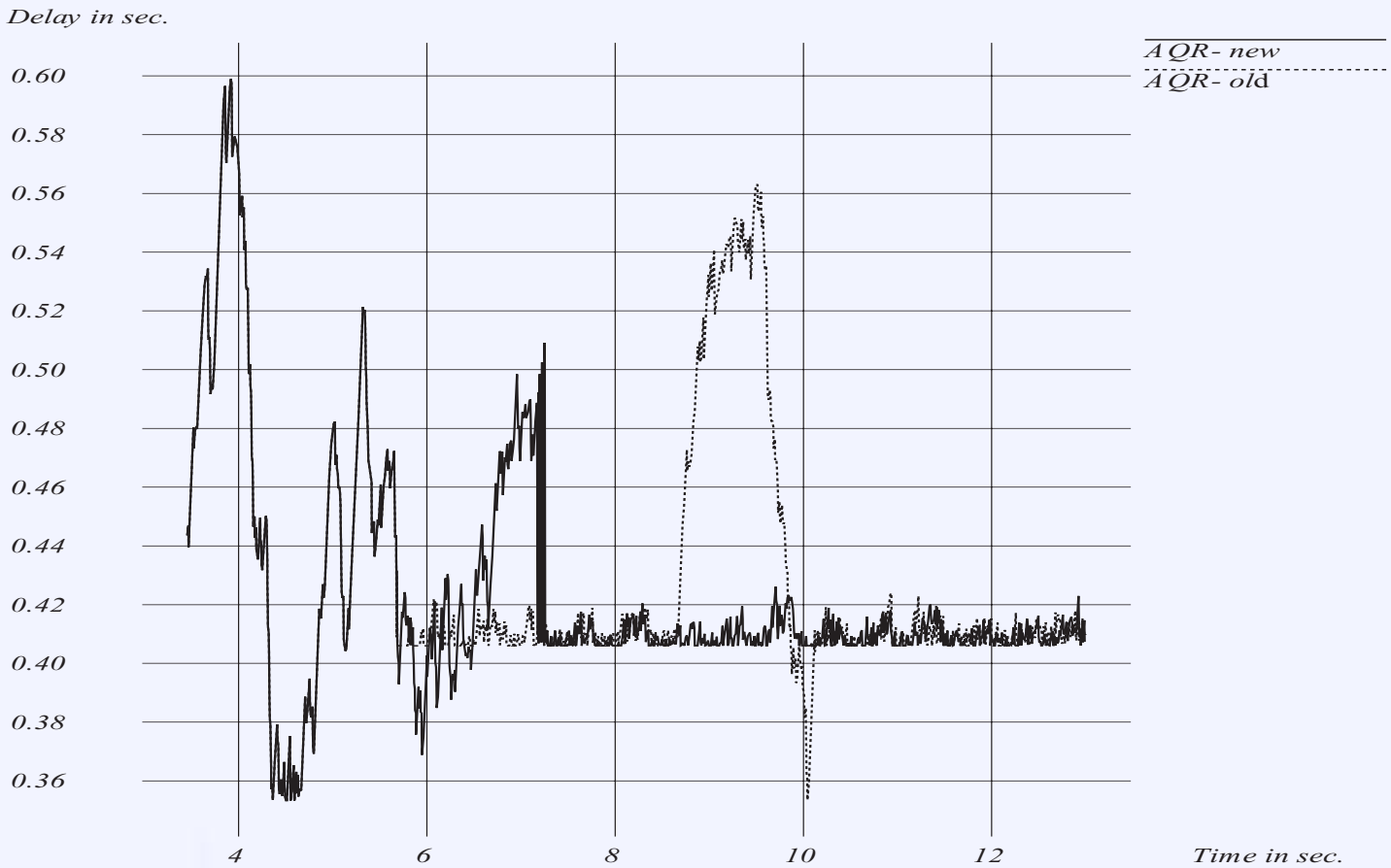
Other QoS Parameters

- Bandwidth:
  - disappointing results with responsive and unresponsive bg traffic
  - note: links had similar nominal bandwidth, only load varied
  - AQR chooses long paths
  - even additional jitter
- Bandwidth / Delay combination:
  - smoother throughput (reduced jitter)
  - faster convergence
- Other possibilities:
  - choose route based on error rate, bandwidth fluctuation, packet loss ratio, cost, ..

Bandwidth / Delay Mixture

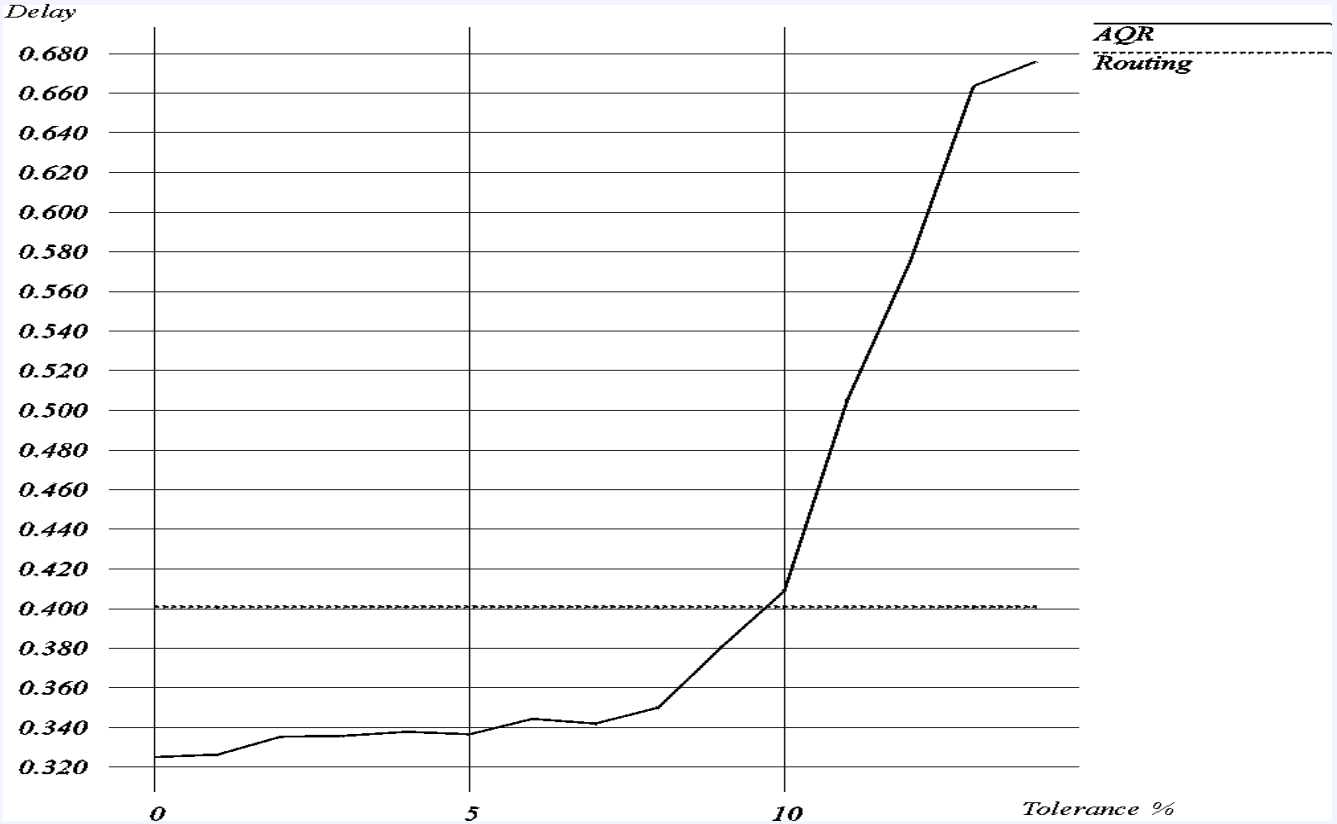


Bandwidth / Delay Mixture /2



Additional Idea: Tolerance Factor

- Split traffic when paths are within tolerance range
  - better resource utilization



Conclusion / Future Work

- Delay: reduced
- Bandwidth / delay: reduced delay + smoother throughput
- Path splitting increases jitter / delay
  - useful for bandwidth, but scalable?
- Work to do: multi-domain routing, problems with lost probing packets, more detailed examination of scalability
- Possible extension (easy?!): multicast

Further Documentation

http://come.to/michael.welzl/AQR/

(Simulator source code, example scripts)