



*Was:*

*Evaluation of different Flow Control Algorithms*

*Is:*

Evaluation of an Enhanced Fairness  
Algorithm that Avoids HOL Blocking

**version 0.1**

Stein Gjessing

Simula Research Lab. / U. Oslo

Oslo, NORWAY

[steing@ifi.uio.no](mailto:steing@ifi.uio.no) [www.ifi.uio.no/~steing](http://www.ifi.uio.no/~steing)

# An Enhanced Fairness Algorithm for avoiding HOL blocking

version 0.1

- Presented previously
- Assume Non-HOL Blocking Queues
- All stations have  $2(N-1)$  counters that for each link records
  - the "owner" usage of the link (sent in a control packet)
  - this stations usage of the link
- Send on a link if you have used less than the "owner"

(The "owner" of a link is the station immediately upstream)

# An RPR model in Java

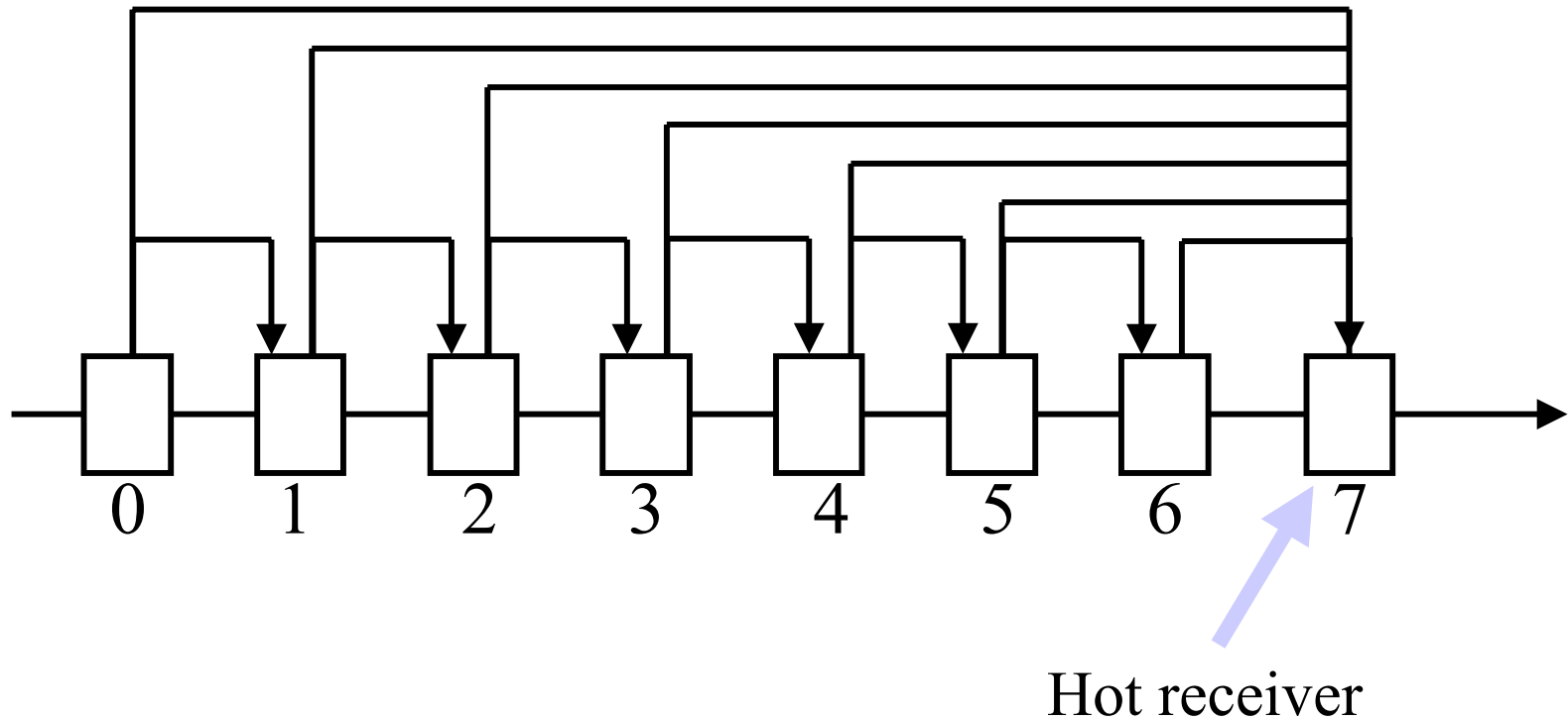
- Two priority levels - Separate buffers for the two
- One tick is one byte transfer
  - OC-48 is about 3ns per tick (8/3 Gbit/sec)
- 16 stations per ring
  - Also run some of the experiments with 32, 64 and 128 stations – nothing special to report
- 2500 ticks cables (also run with 50 000 tick cables)
  - have not noticed any difference on results from longer cables (but have not looked close, parameters should be tuned)
- The model is a program of more than 10 lines, hence it contains errors

~ 2 thousand lines

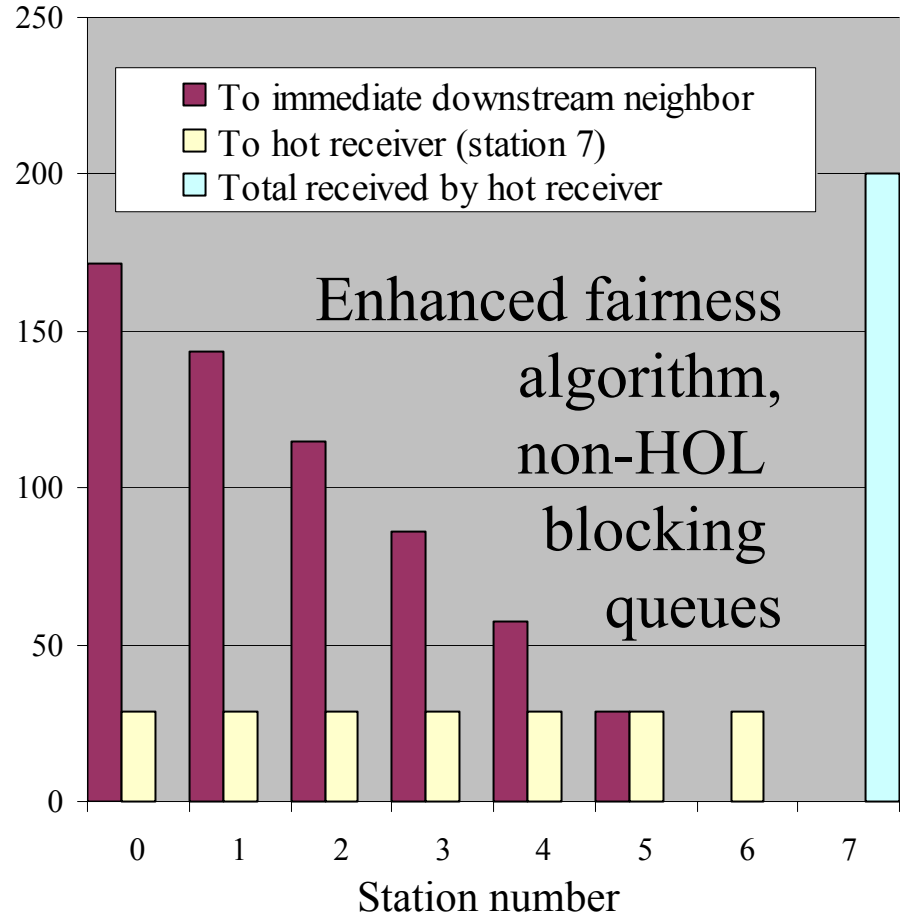
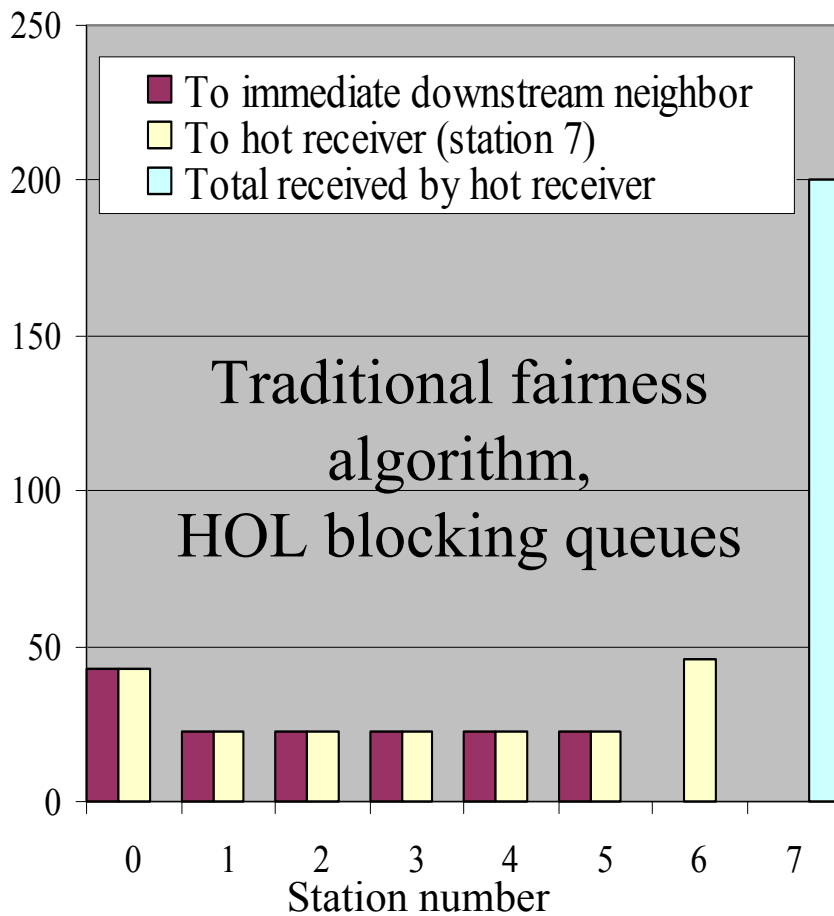
# All experiments

- 500 byte data packet low priority, fairness
- 32 byte control packets, high priority, no fairness
- New fairness algorithm compared with "Traditional fairness"
  - "Traditional fairness" algorithm is my implementation of Ciscos RFC 2892
- All (except "Random startup" at end) are stable situations
  - 99% confidence interval for mean almost not visible
- All maximum load

# Hot receiver and local traffic

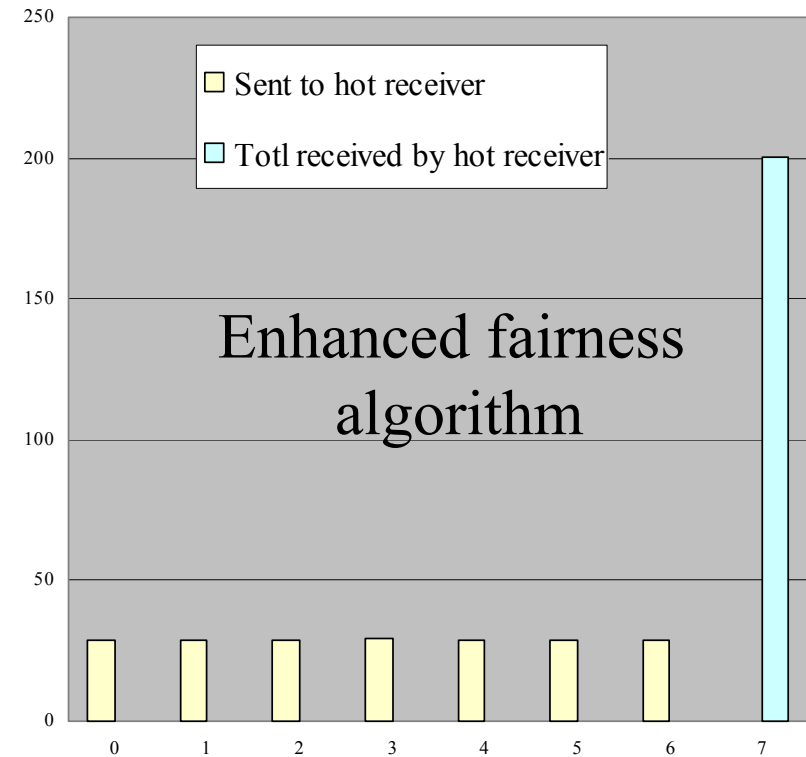
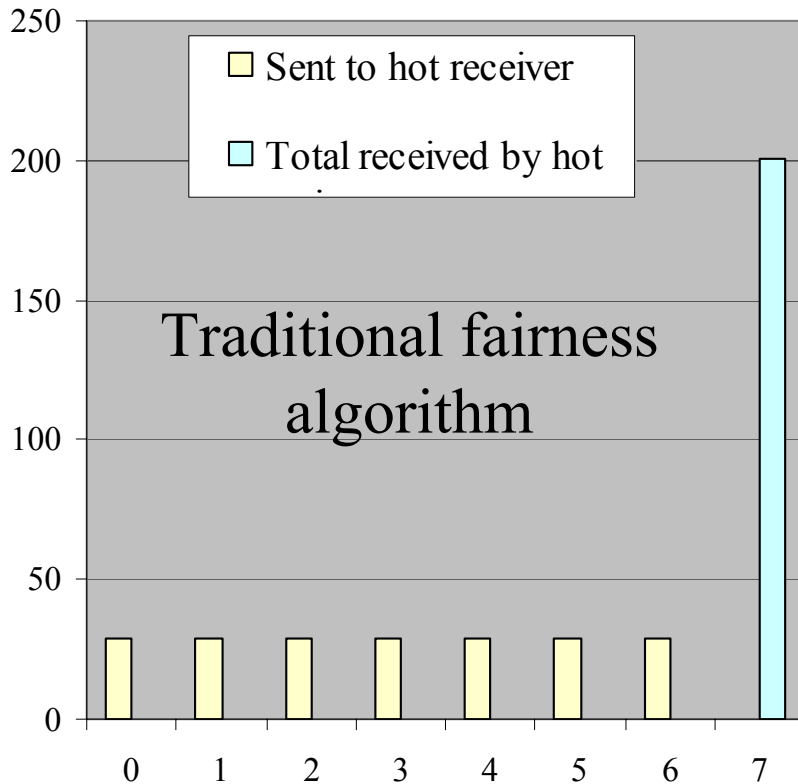


# Hot receiver and local traffic



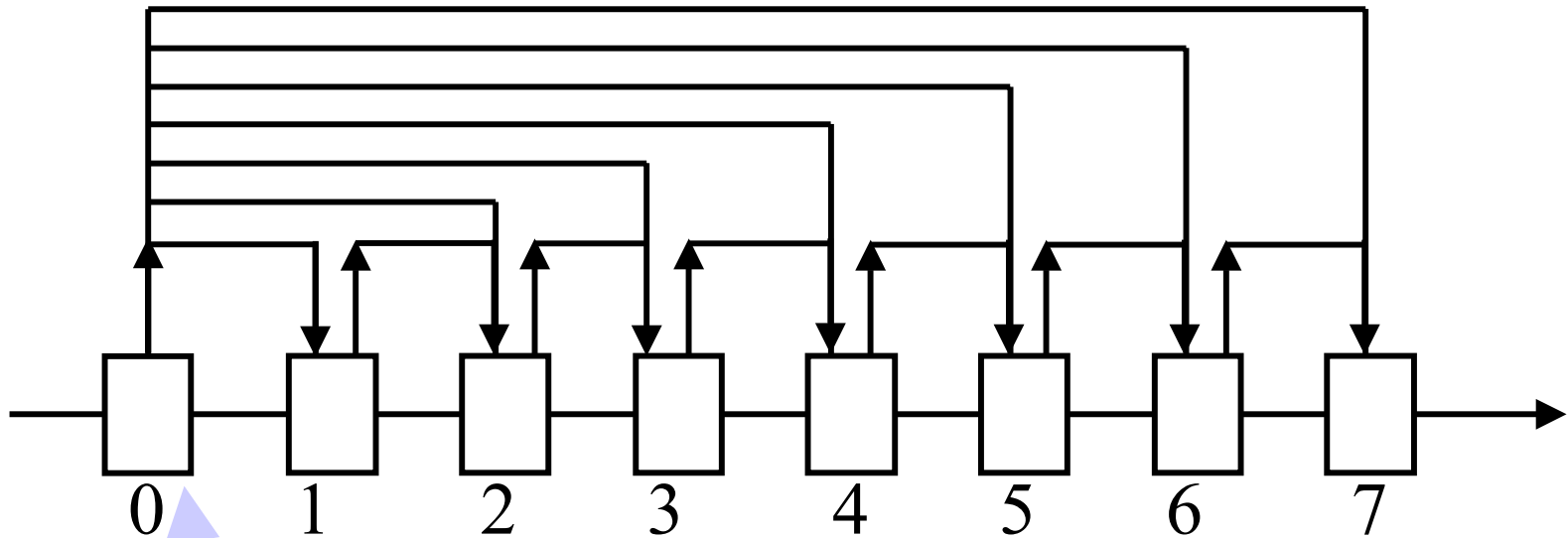
Number of packets per 100 000 ticks (really per 100 160 ticks)

# Hot receiver – no local traffic



No HOL blocking - no difference

# Hot sender and local traffic

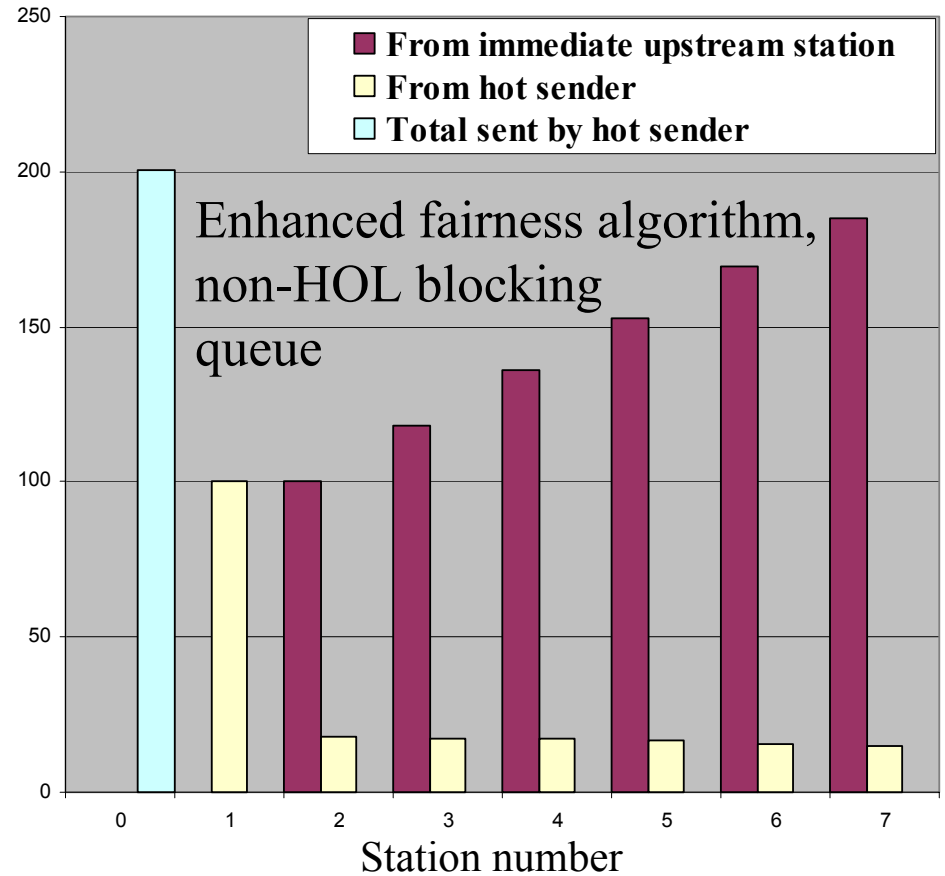
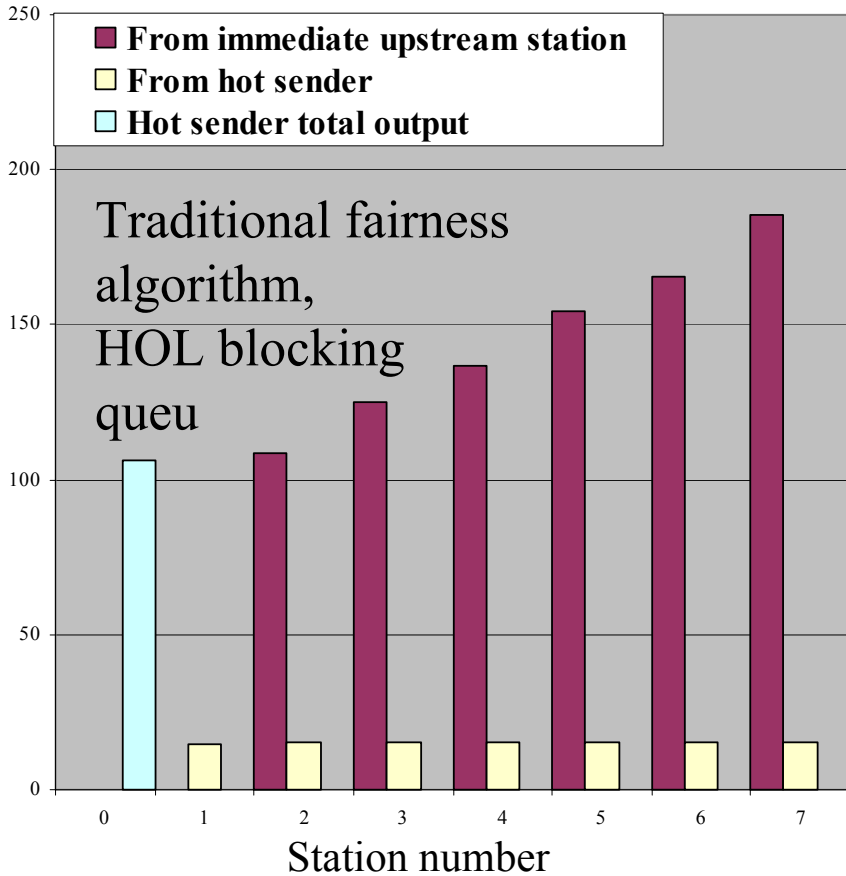


Hot sender

HOL Blocking an issue  
in station 0 only



# Hot sender and local traffic



Number of packets per 100 000 ticks (really pr 100 160 ticks)

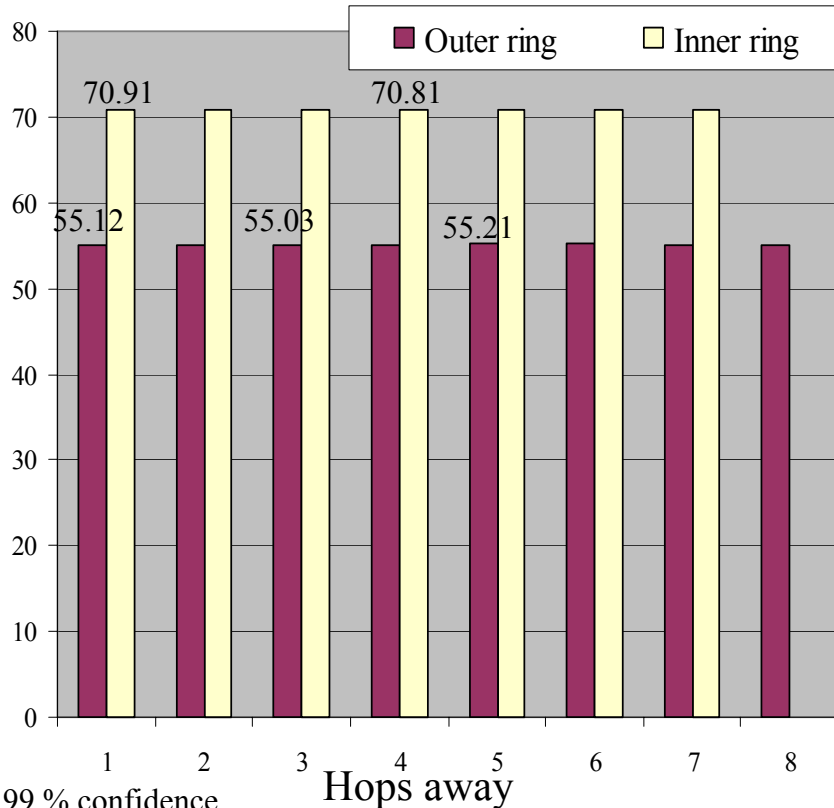
# Random traffic

- All send to all
- **Fully loaded** ring in both directions
  - Control and data packets compete on the wires
- Send shortest path
  - when a tie , use outer ring
  - Hence, outer ring is more congested, and gets fewer packets thru

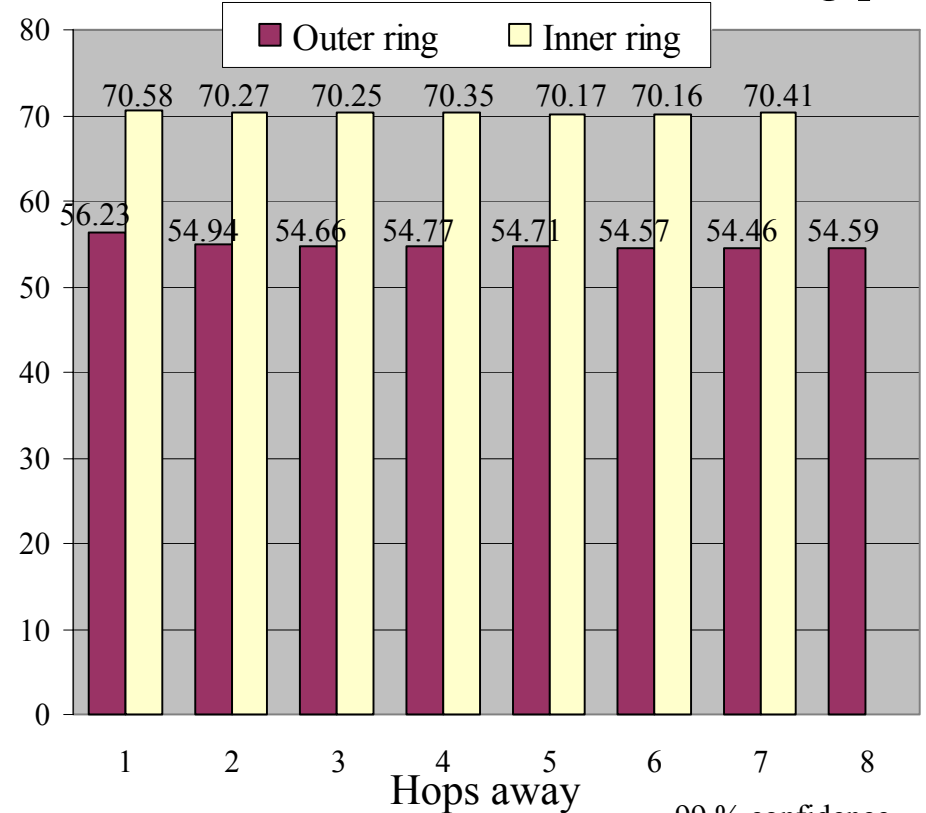
Traditional fairness algorithm,  
HOL blocking queues

# Random traffic

Enhanced fairness algorithm,  
non-HOL blocking queue



99 % confidence interval for the mean is ~0.15 packets

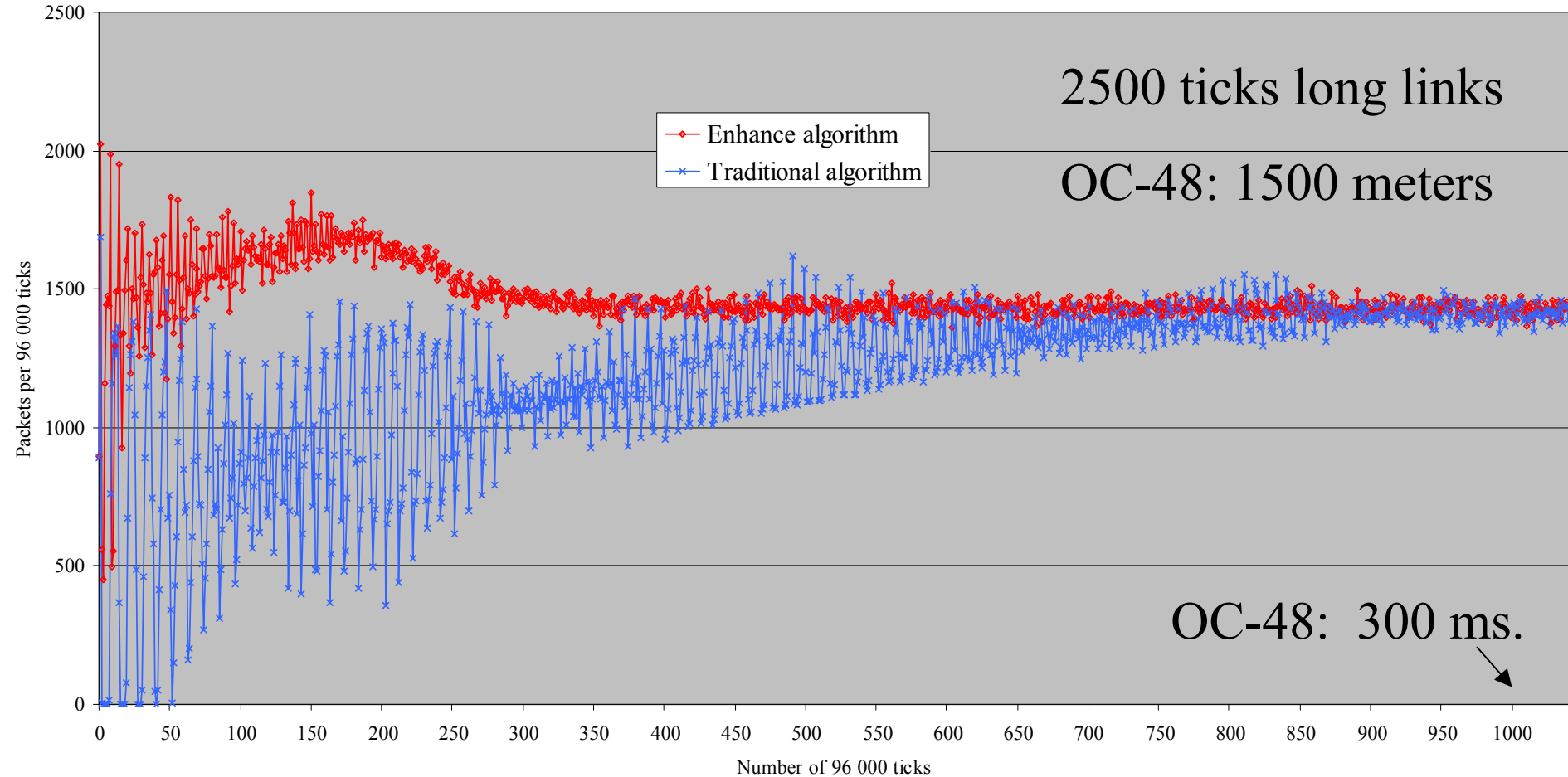


99 % confidence interval for the mean is ~0.3 packets

## Packets per station per million ticks

# Startup of Random traffic

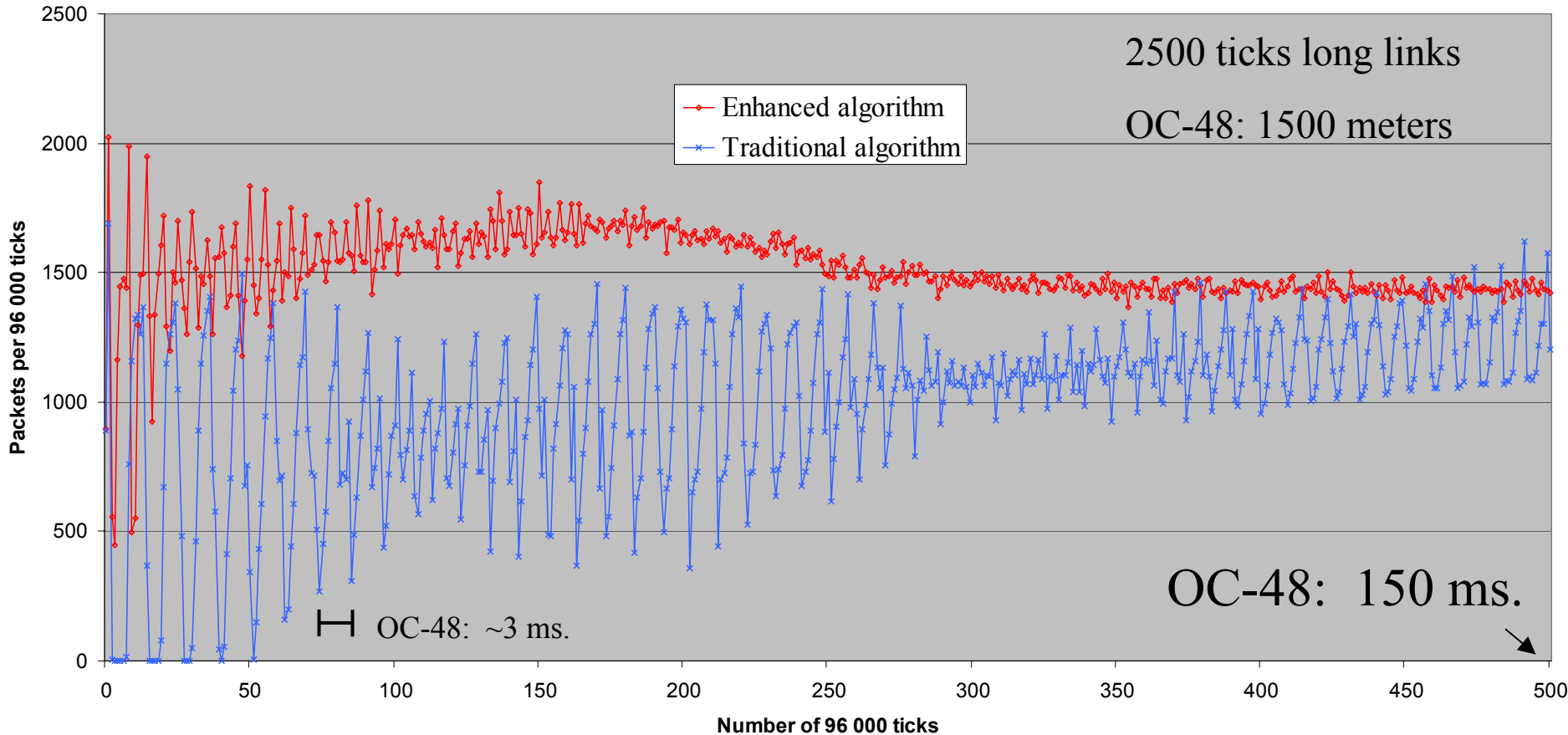
- short links (remember: full load)



Total number of packets received on 16 node ring

# Random traffic – startup

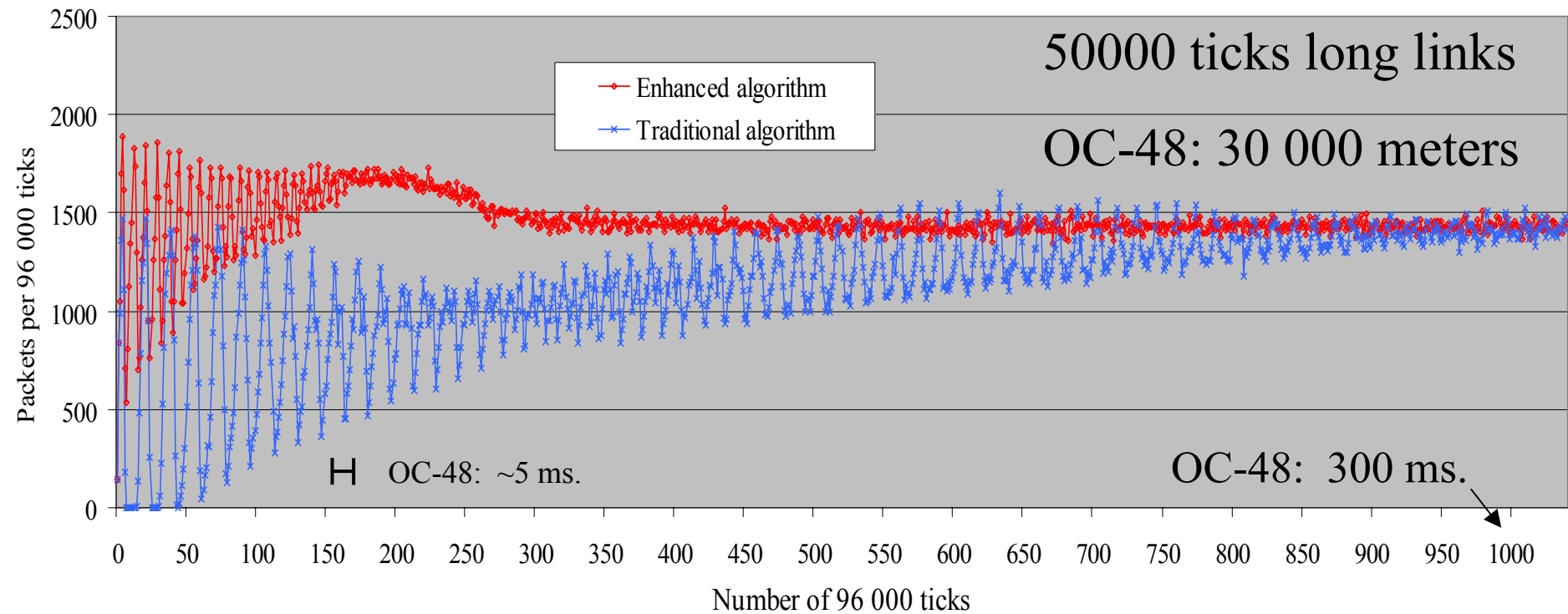
(short links - enlarged first 1/2 of previous)



Total number of packets received on 16 node ring

# Startup of Random traffic

- long links (remember: full load)



Total number of packets received on 16 node ring

# Recorded Passthru buffer depth

## - short / long links

- All Passthru buffer depth recorded at every measure point ( each 96 000 ticks, ~1040 measure points)
- Mean:
  - Traditional: 2764 / 2889 (~half of the buffers empty at any time)
  - Enhanced: 21052 / 21101 (most buffers non-empty at any time)
- Max (threshold is 25 000 bytes) :
  - Traditional: 25 000 / 25 500 bytes
  - Enhanced: 26 000 / 25 500 bytes

# Conclusion

## The Enhanced Fairness Algorithm

version 0.1

- Avoids HOL blocking almost perfectly
- Is dynamic
  - No pre-set loads
- Is aggressive
  - Might be too aggressive ?
- Simple algorithm
  - Few counters
  - Small control packets broadcasted
  - Or can aggregate status to larger point-to-point packets  
(not shown)