Assessing Interactions among Legacy and High-Speed TCPs

HIDEyuki Shimonishi

Tutomu Murase

h-shimonishi@cd.jp.nec.com

NEC

Medy Sanadidi

UCLA CSD

Motivation

Study statistical behavior of various high-speed protocols

-In arbitrary topology networks, with multiple bottleneck links on a flow path, with plenty of short sized flows

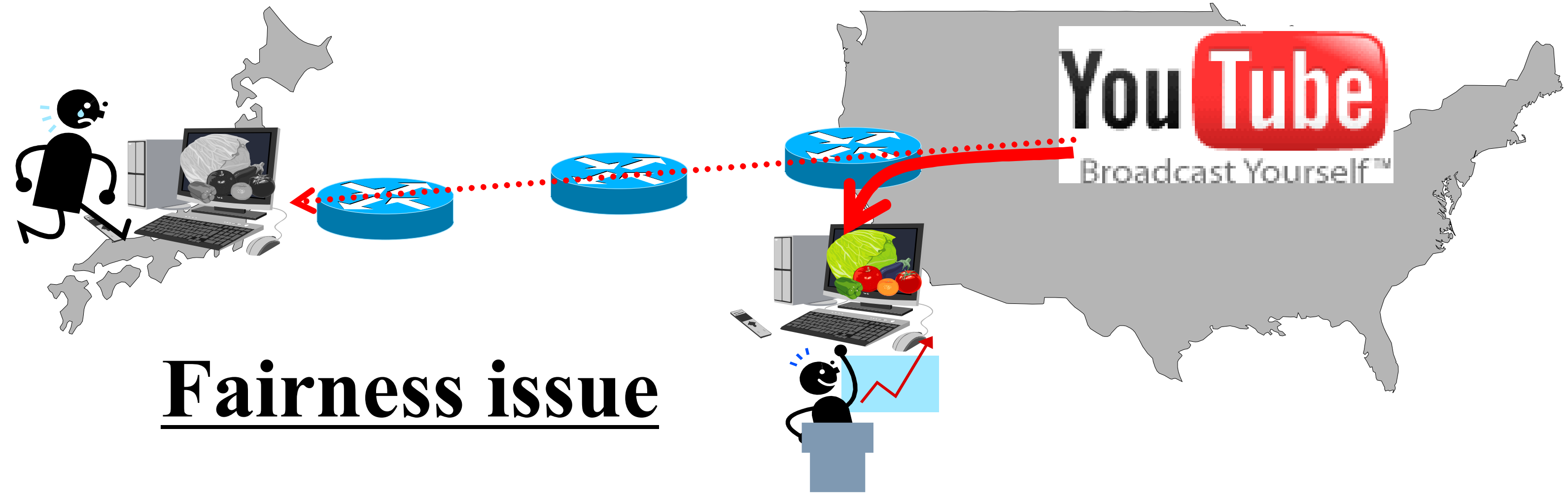
How do they behave differently ?

Fairness and friendliness, as well as efficiency and throughput

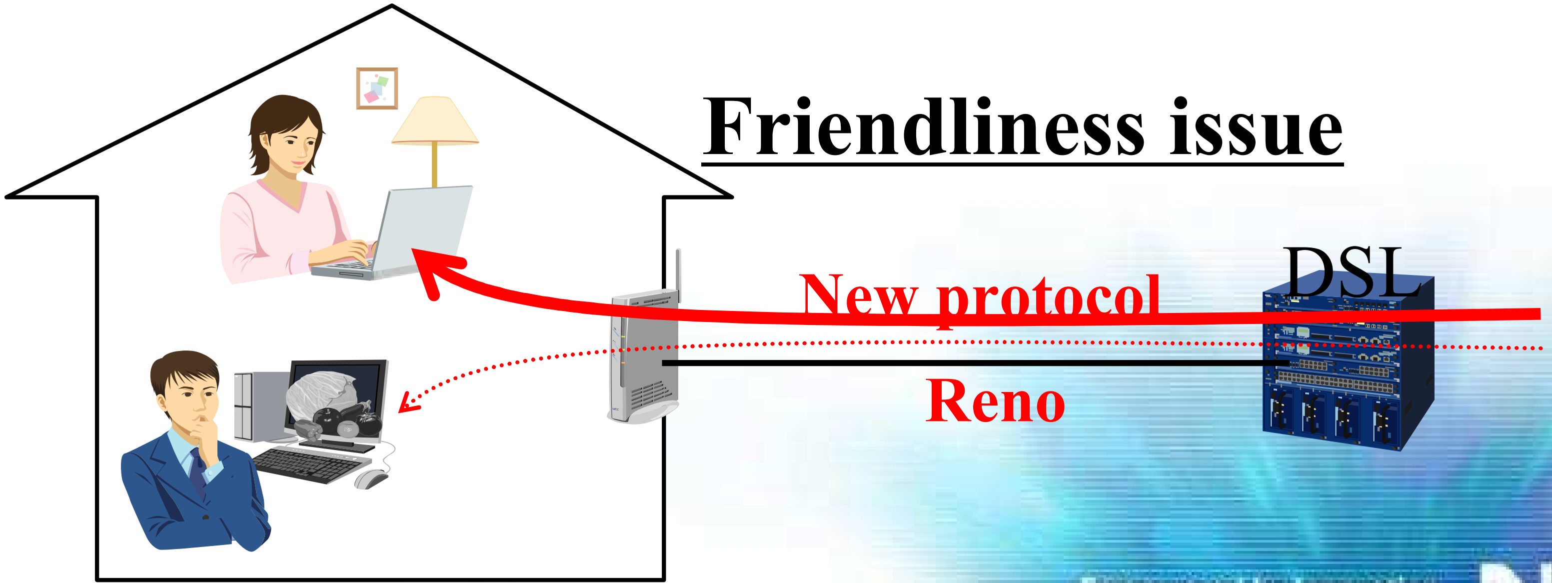
How do they co-exist with Reno ?

Can we have reasonable scenario for migration ?

continued



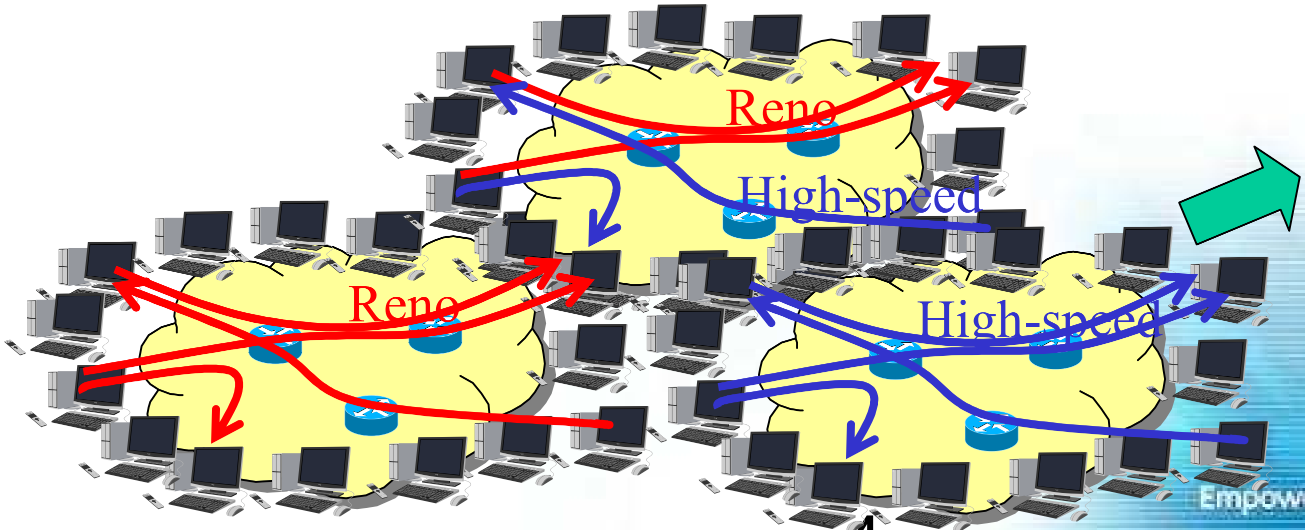
You Tube
Broadcast Yourself™
Fairness issue



Friendliness issue
DSL
New protocol
Reno

Comparative study
Same configuration but different protocols

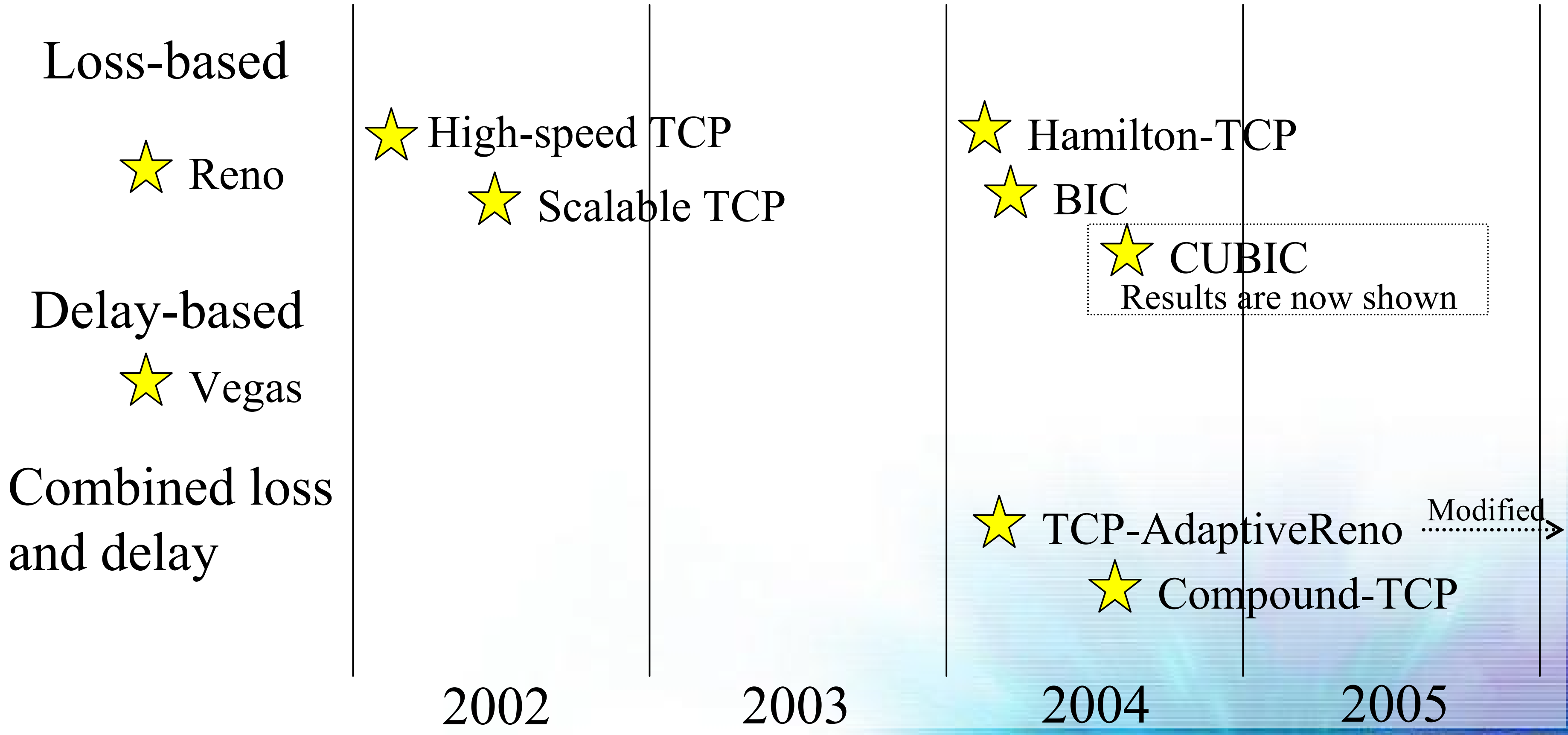
- Run multiple experiments/simulations using the same configuration
  - Same topology and link configurations
  - Same set of flows
  - Same object creation of each flow
- But different protocols
  - Reno, high-speed, and their mixture



Flow-by-flow, file-by-file comparison

Protocols compared

- Used NS2 patch for Linux TCP congestion control modules



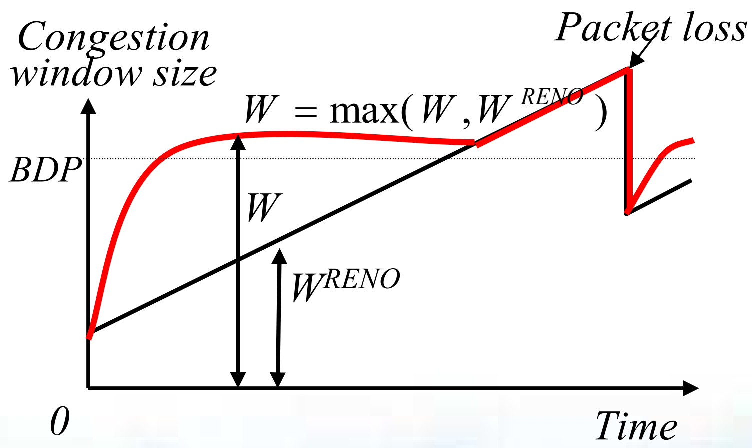
TCP-AdaptiveReno (AReno)

- Loss-based AIMD mechanism + adaptive window increase using delay information
  - During congestion avoidance

$$W += \left(\alpha \frac{B}{R} RTTe^{c} - \beta W c \right) / W$$

$$W^{RENO} += 1 / W$$

$$W = \max(W, W^{RENO})$$



α, β; control parameter
R; Achieved rate (=W/RTT)
B; Estimated link capacity

c; delay-based congestion estimation

$$c = \frac{RTT - RTT_{\min}}{RTT_{cong} - RTT_{\min}}$$

  - Upon packet loss

$$W = \frac{1}{1+c} W, \quad W^{RENO} = \frac{1}{1+c} W^{RENO}$$

TCP-AdaptiveReno (AReno) –cont'd

- More attention on transient state, rather than steady state

$$W \leftarrow W + \left(\alpha \frac{B}{R} RTTe^{c} - \beta Wc \right) / W$$

α, β; control parameter
R; Achieved rate (=W/RTT)
B; Estimated link capacity

c; delay-based congestion estimation

$$c = \frac{RTT - RTT_{\min}}{RTT_{cong} - RTT_{\min}}$$

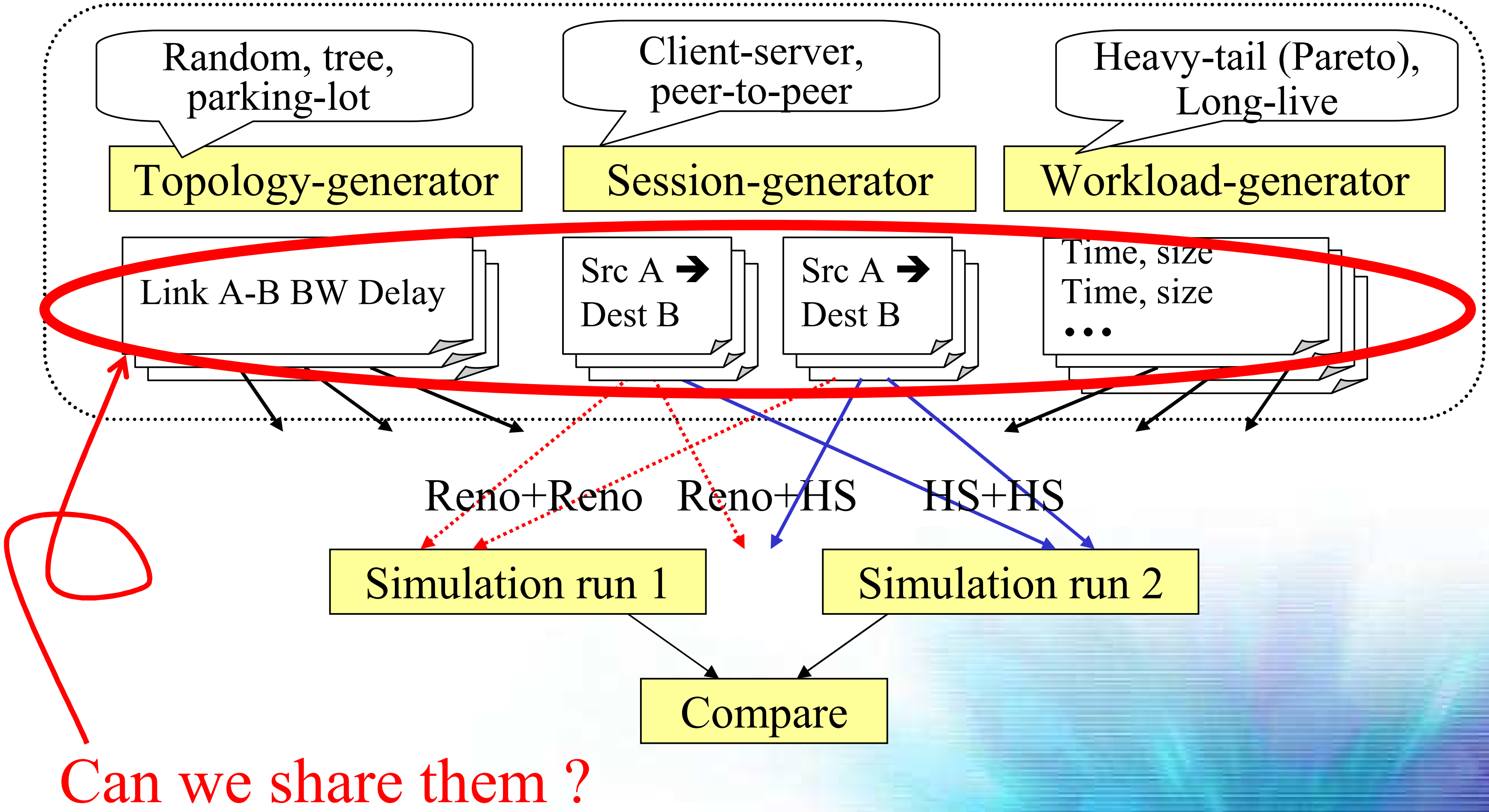
- Improve RTT-fairness
  - ➔multiply *RTT*
- Improve friendliness to Reno
  - ➔multiply e^c and *c*
- Scalable to high-speed network
  - ➔multiply *B/R*

- Steady state equilibrium; $\alpha (B/R) RTTe^{c} = \beta Wc$
  - No RTT factor, but bottleneck link capacity and delay

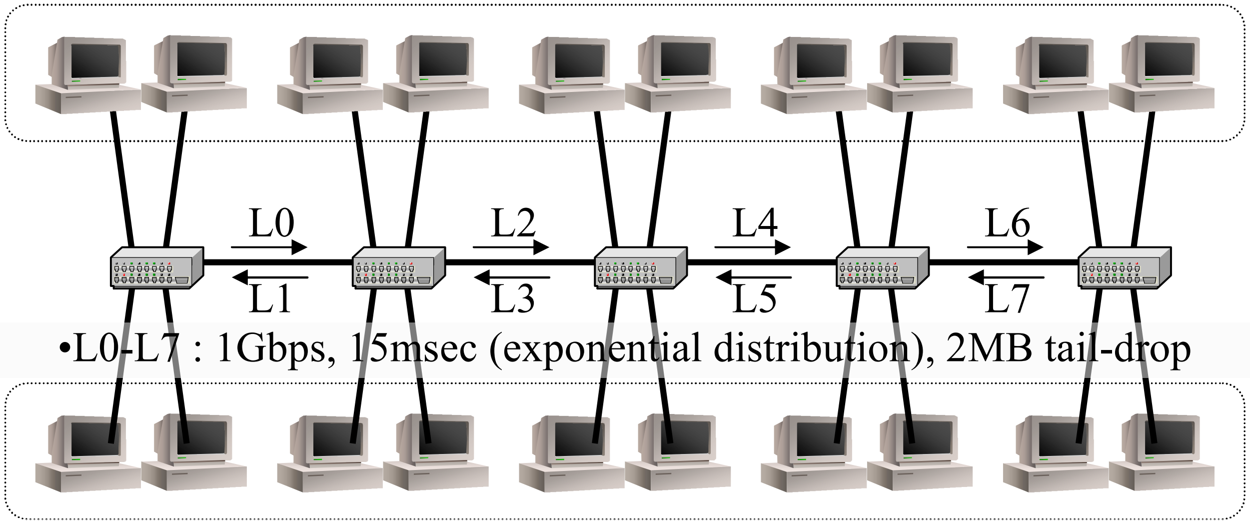
$$SendingRate \quad R = \sqrt{B \frac{\alpha}{\beta} \frac{e^{c}}{c}}$$

Empowered by Innovation NEC

Our methodology



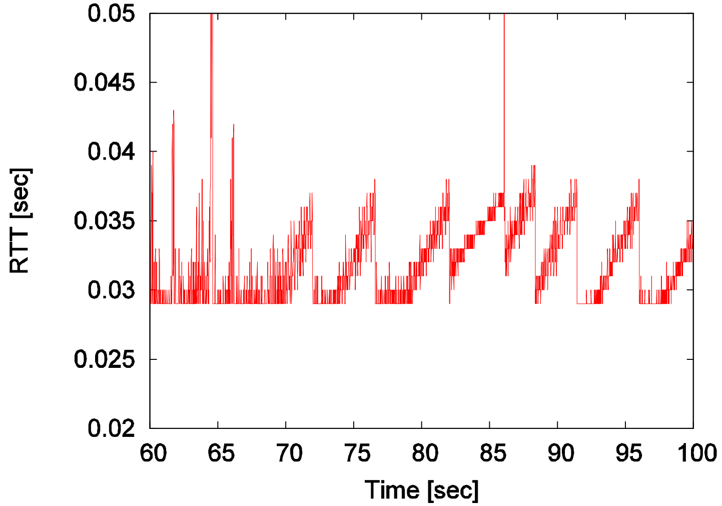
Simulation configuration



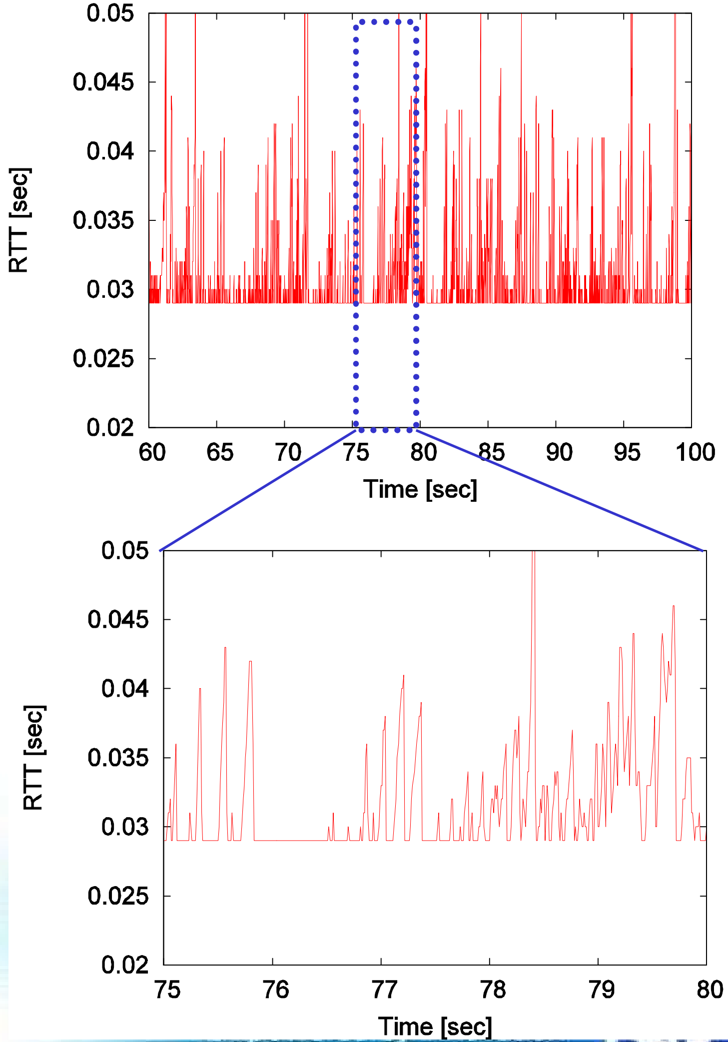
- Topologies
  - Parking lot with 5 routers, 1Gbps links with 2MB buffer (15msec)
  - Average round trip delay of a flow = 130msec (exponential)
- Sessions
  - 100 short-lived flows, 1-40 long-lived flows
- Workloads
  - Short-lived: 1MB file (Pareto), 1sec inter-arrival time (exponential)
  - Long-lived: 4.7GB file (fixed), 2min inter-arrival time (exponential)

RTT of a flow; an example

Without short-lived flows



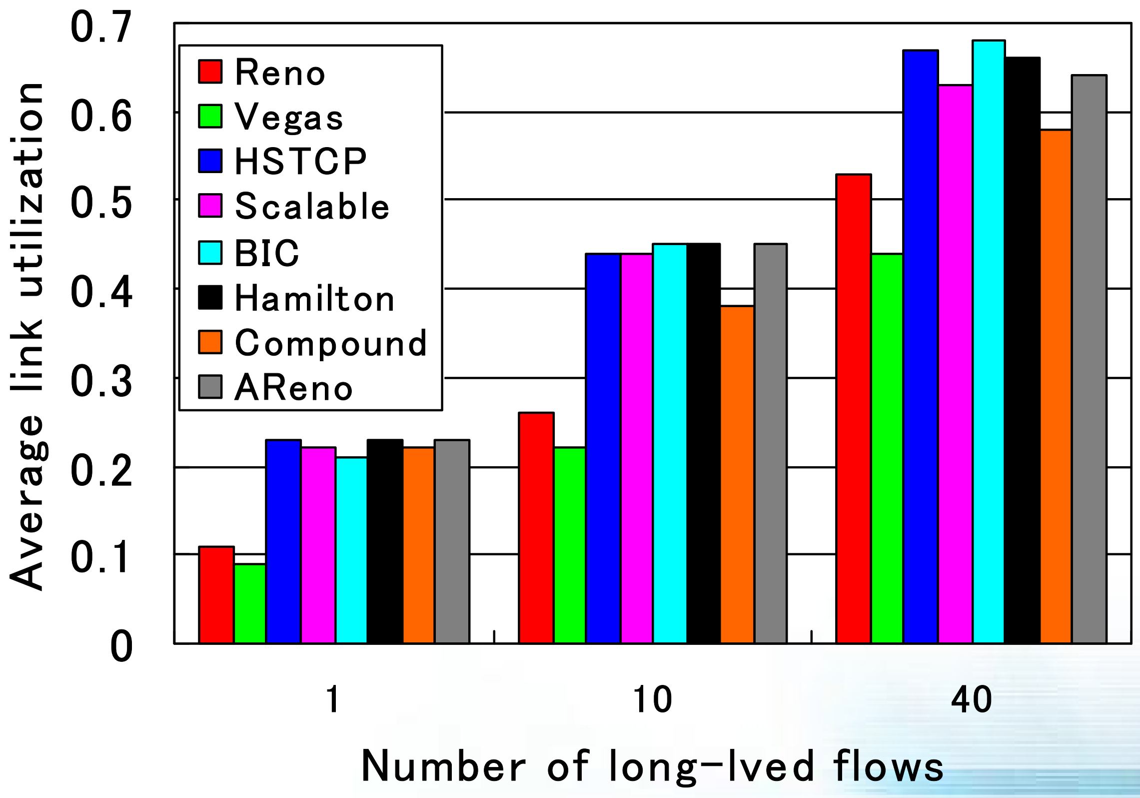
With short-lived flows



- Many spikes due to slow-start of short-lived flows
- Not very good situation for delay-based protocols

Efficiency improvement (1)
Overall link utilization

- Overall utilization of 8 backbone links



- High-speed TCPs improve efficiency
  - Compound is bit milder

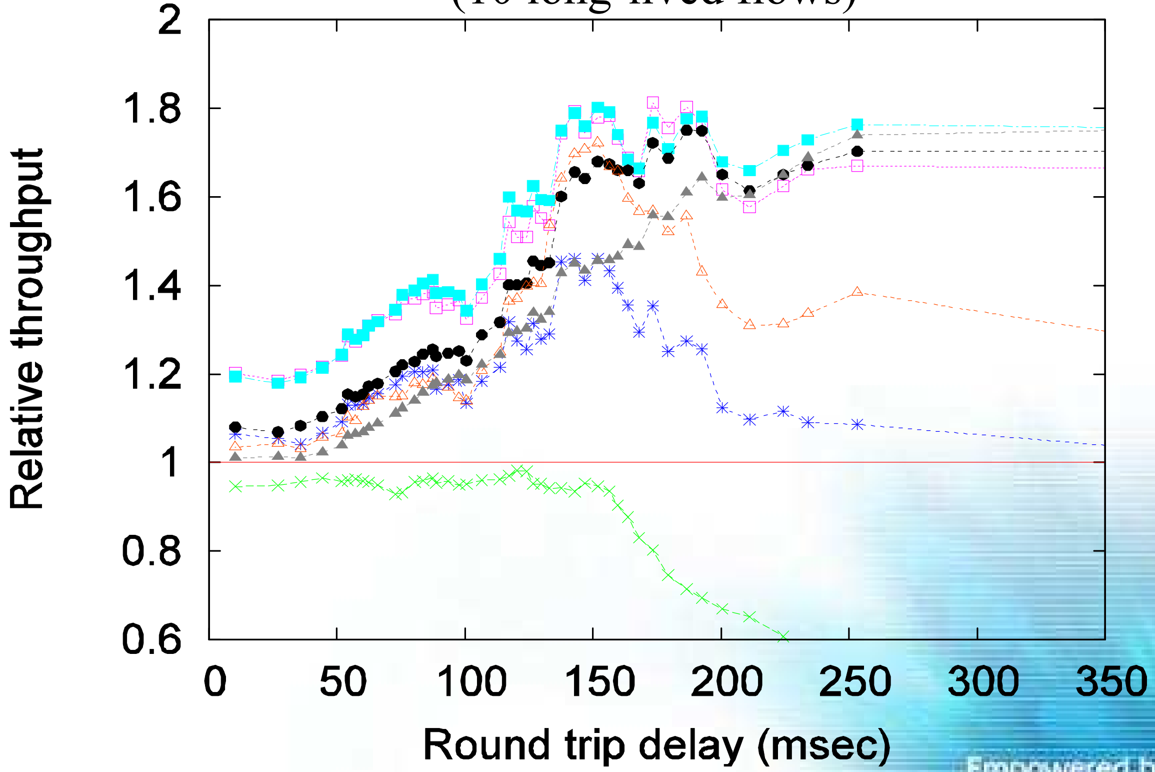
Efficiency improvement (2)
Throughput improvement vs. RTT

- Per-flow throughput improvement vs. RTT

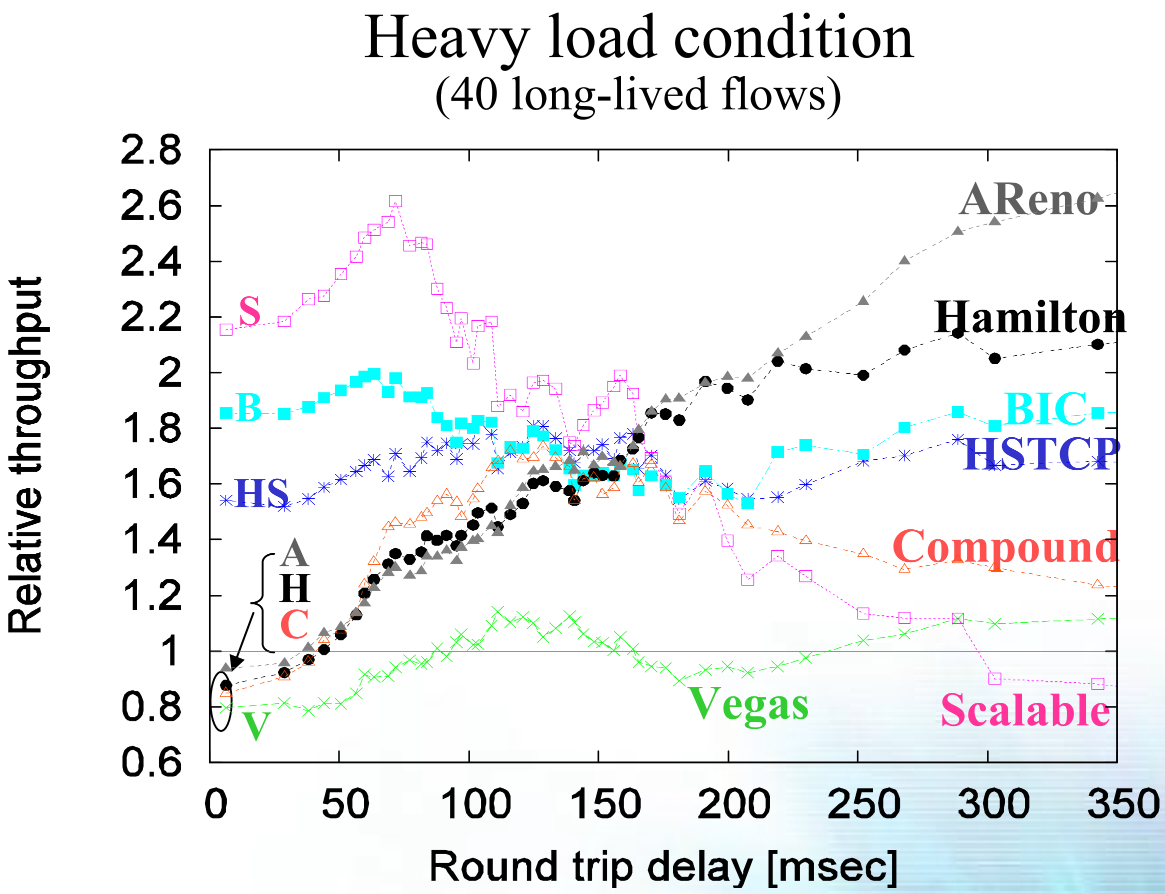
$$\text{Relative throughput} = \Sigma_{i<N} \left(\frac{\text{Throughput of flow i (using high-speed)}}{\text{Throughput of flow i (using Reno)}} \right) / N$$

Light load condition

(10 long-lived flows)

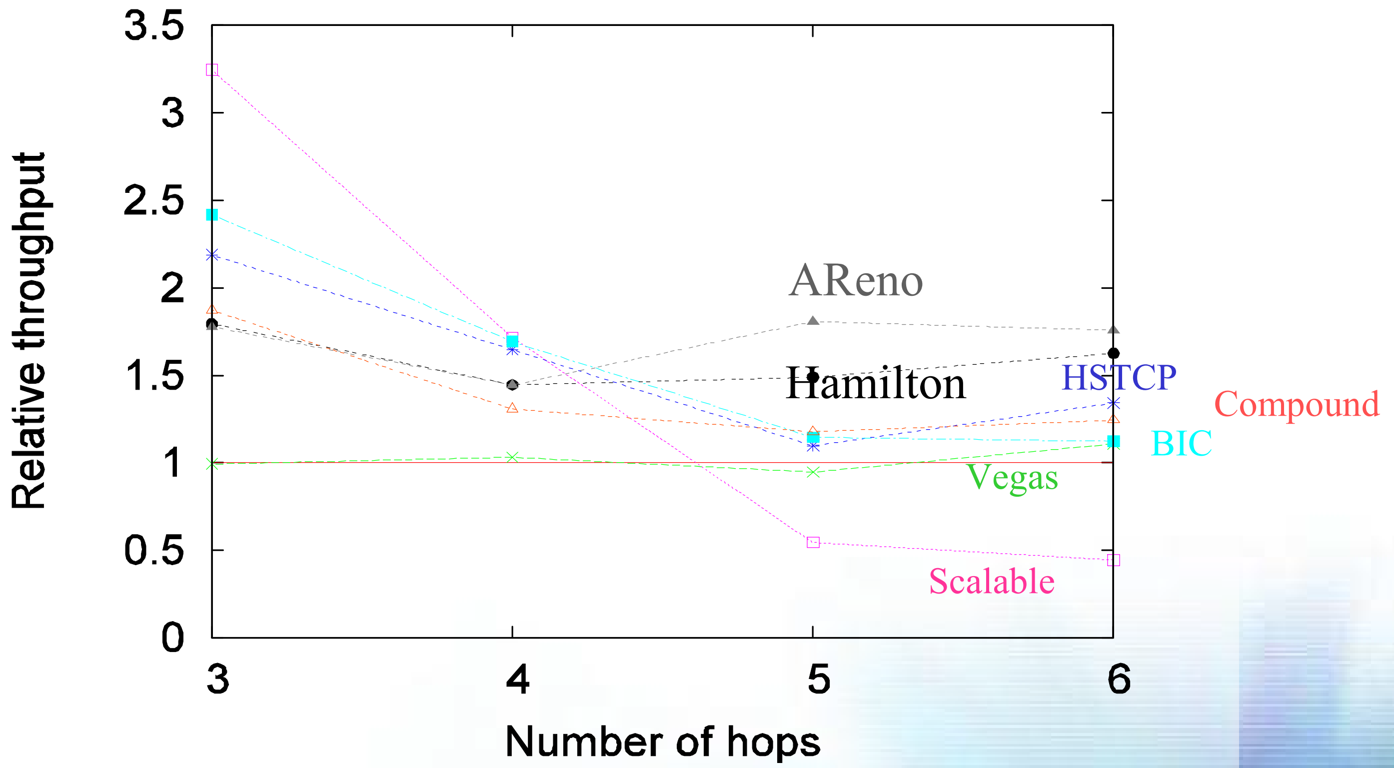


continued


Heavy load condition
(40 long-lived flows)
Relative throughput
Round trip delay [msec]
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0
50
100
150
200
250
300
350
AReno
Hamilton
BIC
HSTCP
Compound
Vegas
Scalable
S
B
HS
A
H
C
V

Efficiency improvement (3)
Throughput improvement vs. hop-count

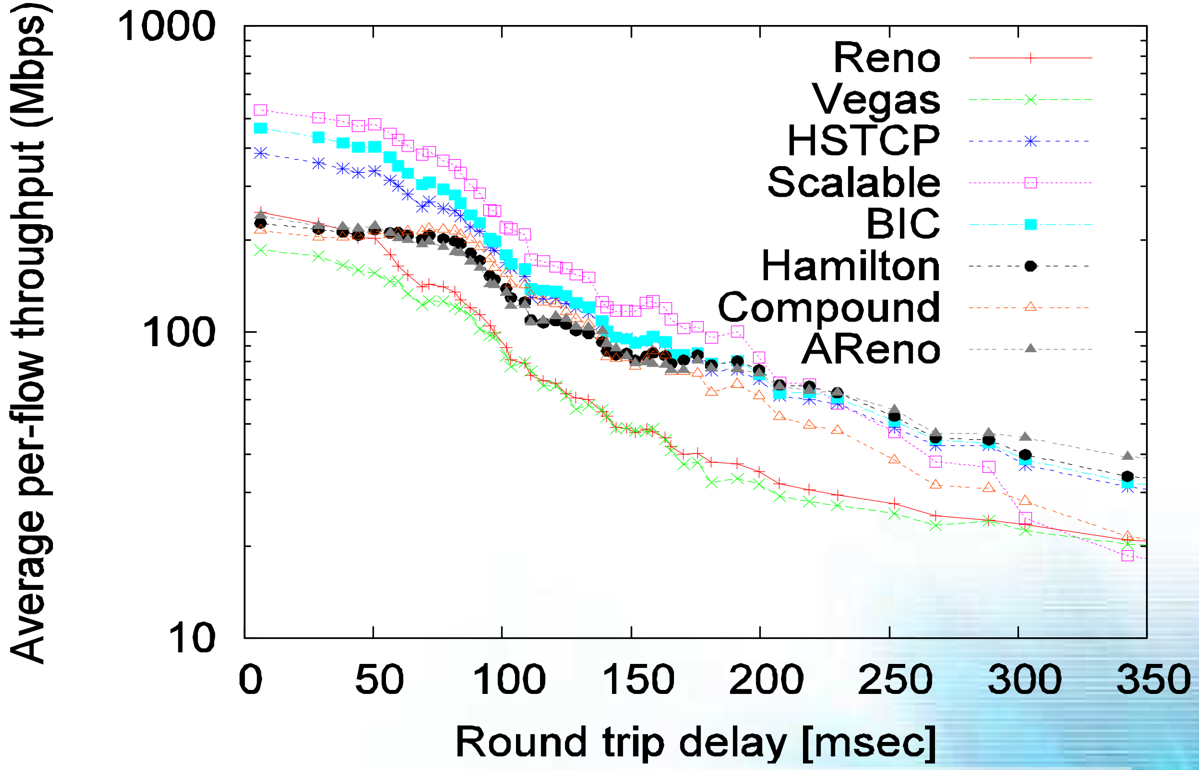
- Per-flow throughput improvement vs. hop-count



•Scalable : only short hop flows improve
•AReno and Hamilton : all flows improve regardless of hop counts

Fairness (1) Throughput vs. distance

- Per-flow throughput in heavy load condition (40 long-lived flows)

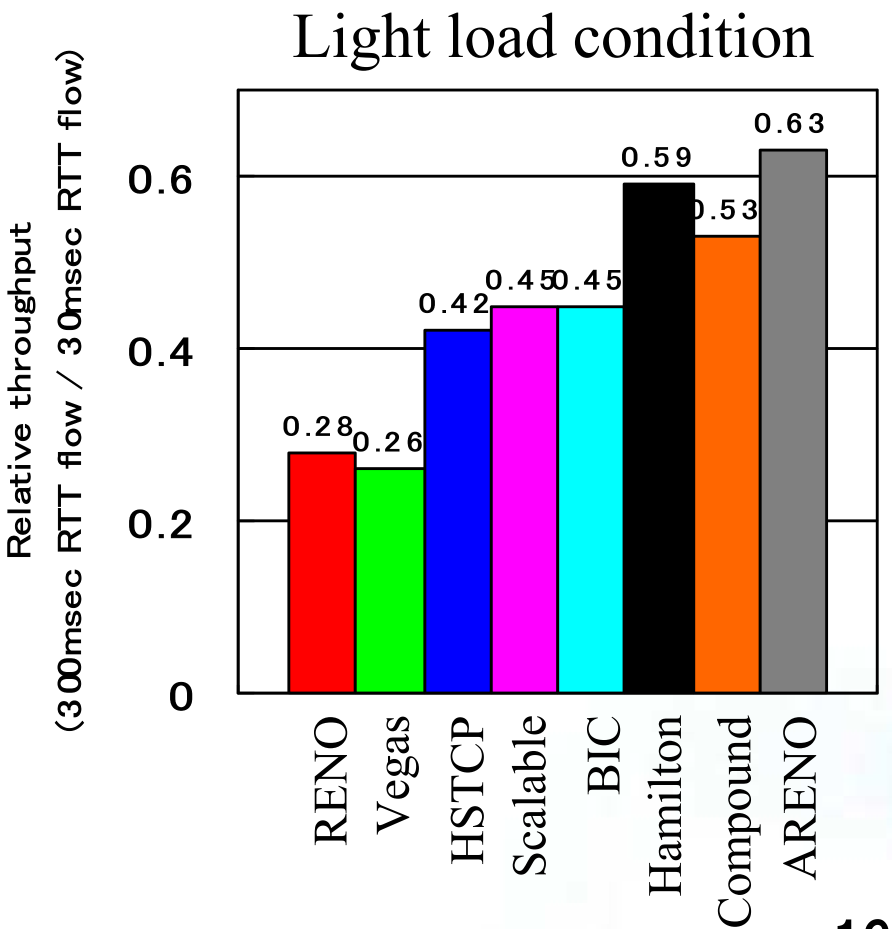


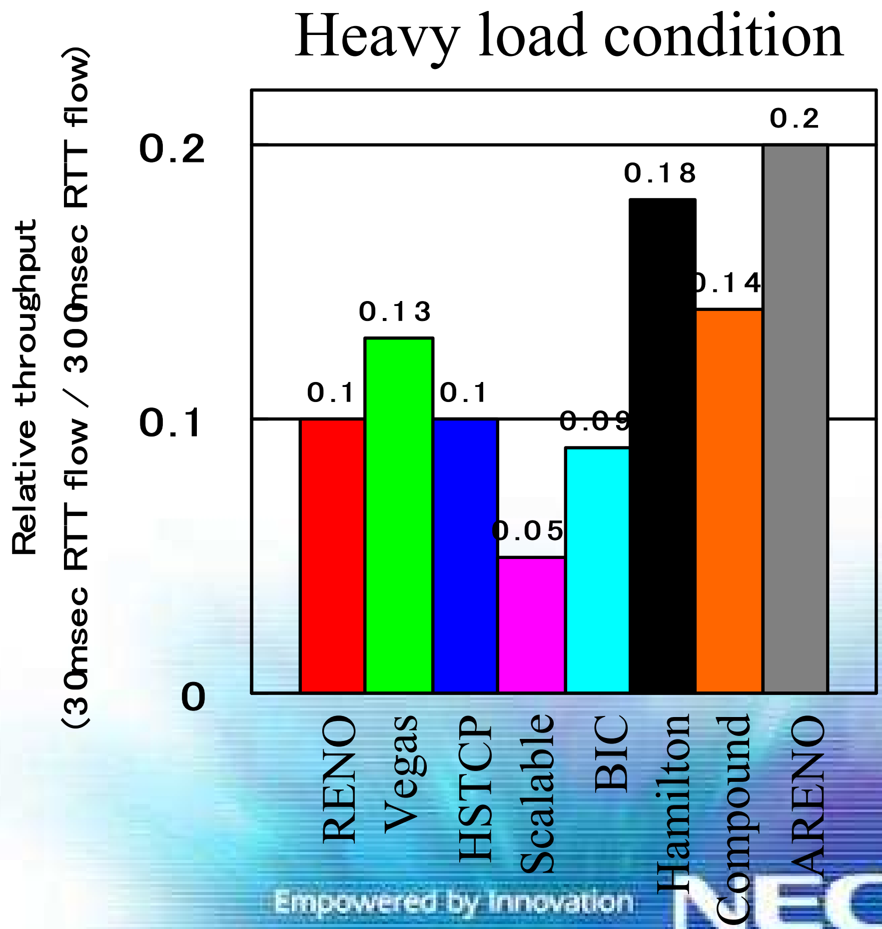
•Scalable : more steep

•AReno and Hamilton : more flat

Fairness (2) Throughput vs. distance

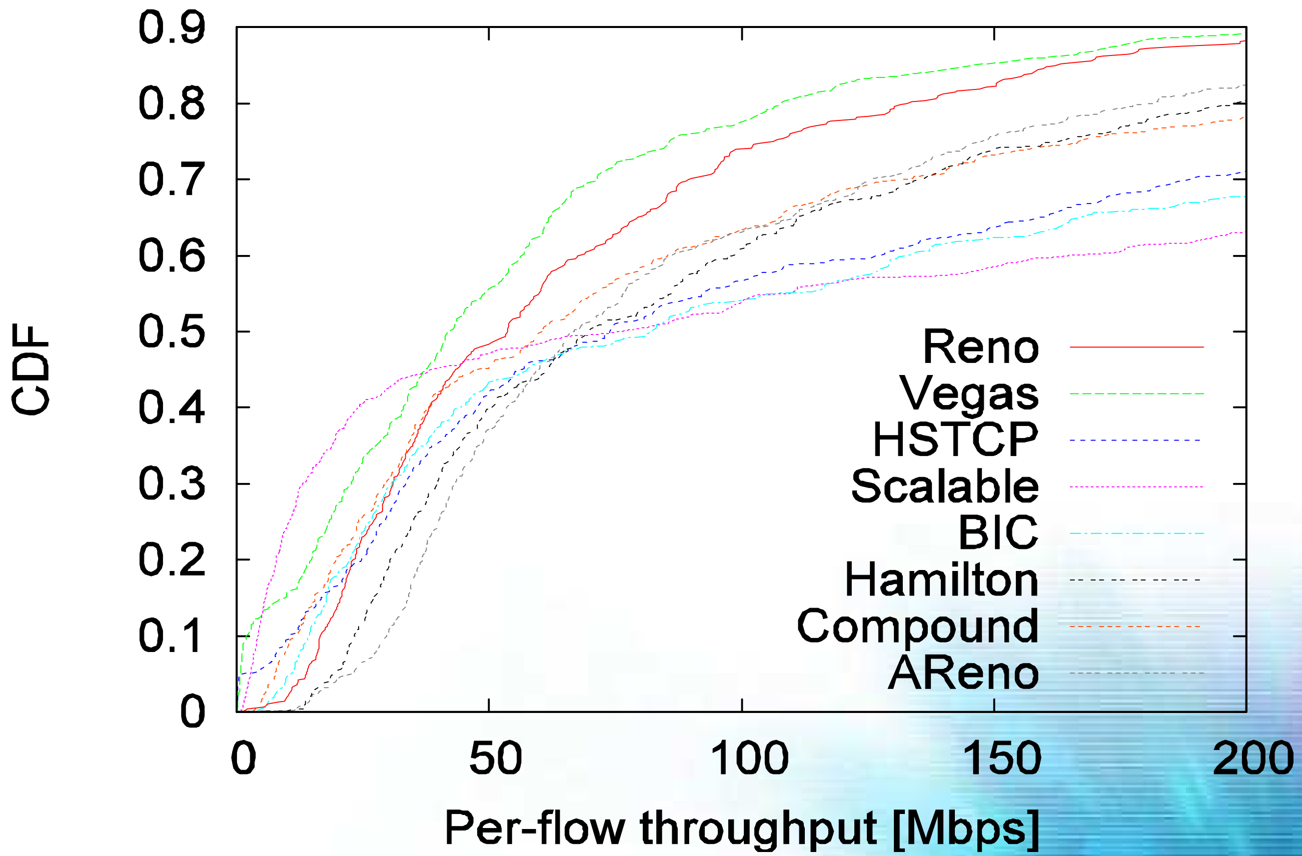
- Relative throughput of long flow (300msec RTT) and short flow (30ms RTT)



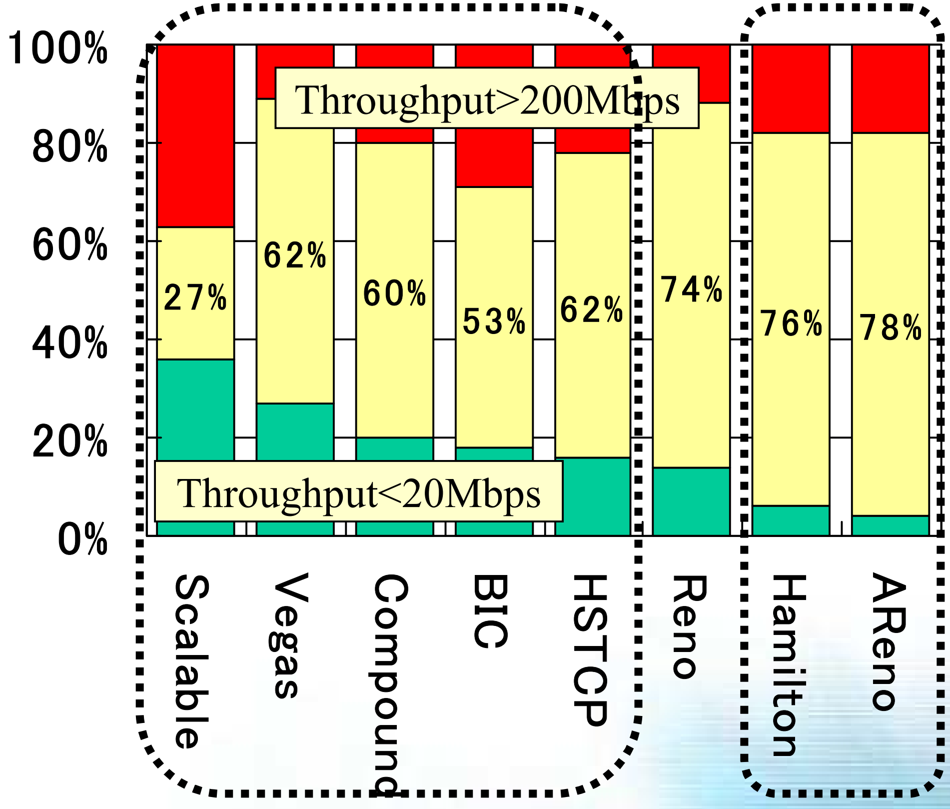


Fairness (3) CDF of per-flow throughput

- Cumulative distribution of per-flow throughput in heavy load condition (40 long-lived flows)



continued

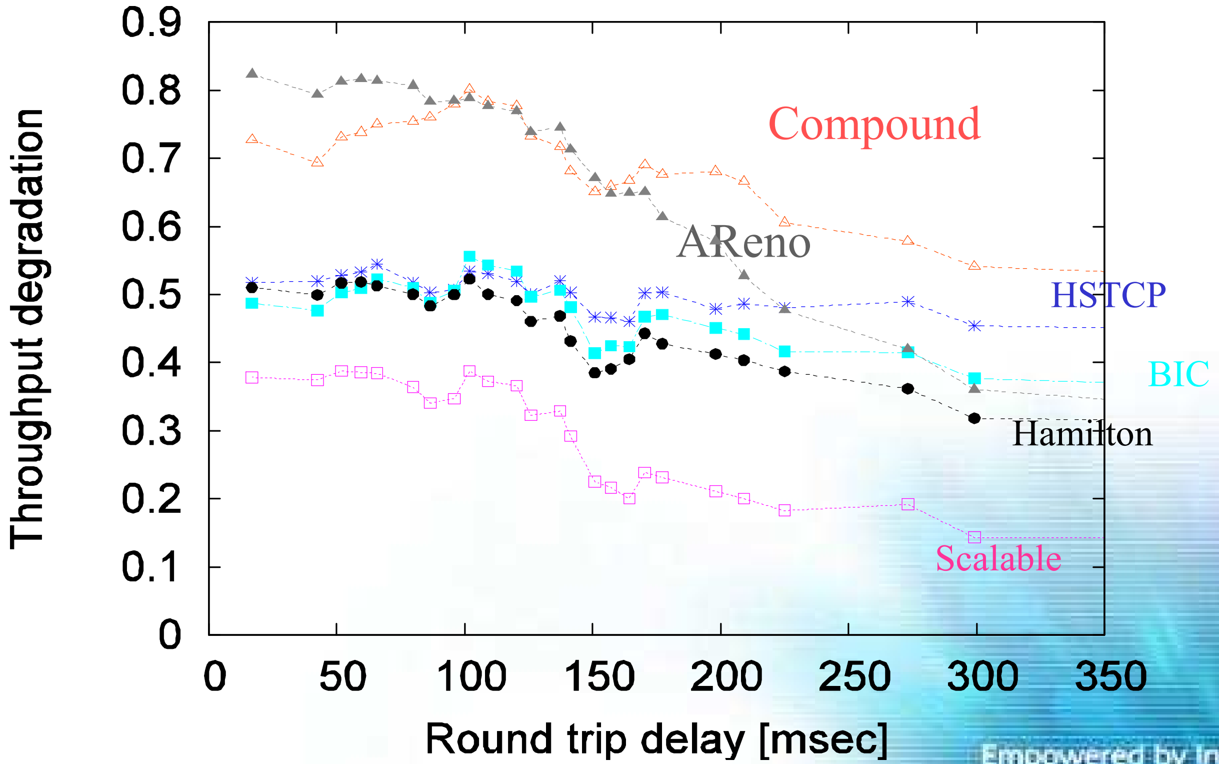

100%
80%
60%
40%
20%
0%
Throughput>200Mbps
Throughput<20Mbps
27%
62%
60%
53%
62%
74%
76%
78%
Scalable
Vegas
Compound
BIC
HSTCP
Reno
Hamilton
AReno

Friendliness to Reno (1)
Throughput degradation of Reno vs. RTT

- Throughput degradation of Reno flows

$$\text{Relative throughput} = \Sigma_{i<N} \left(\frac{\text{Throughput of flow i (coexisting with HS flows)}}{\text{Throughput of flow i (coexisting with Reno flows)}} \right) / N$$

  – Indexed by coexisting high-speed flow



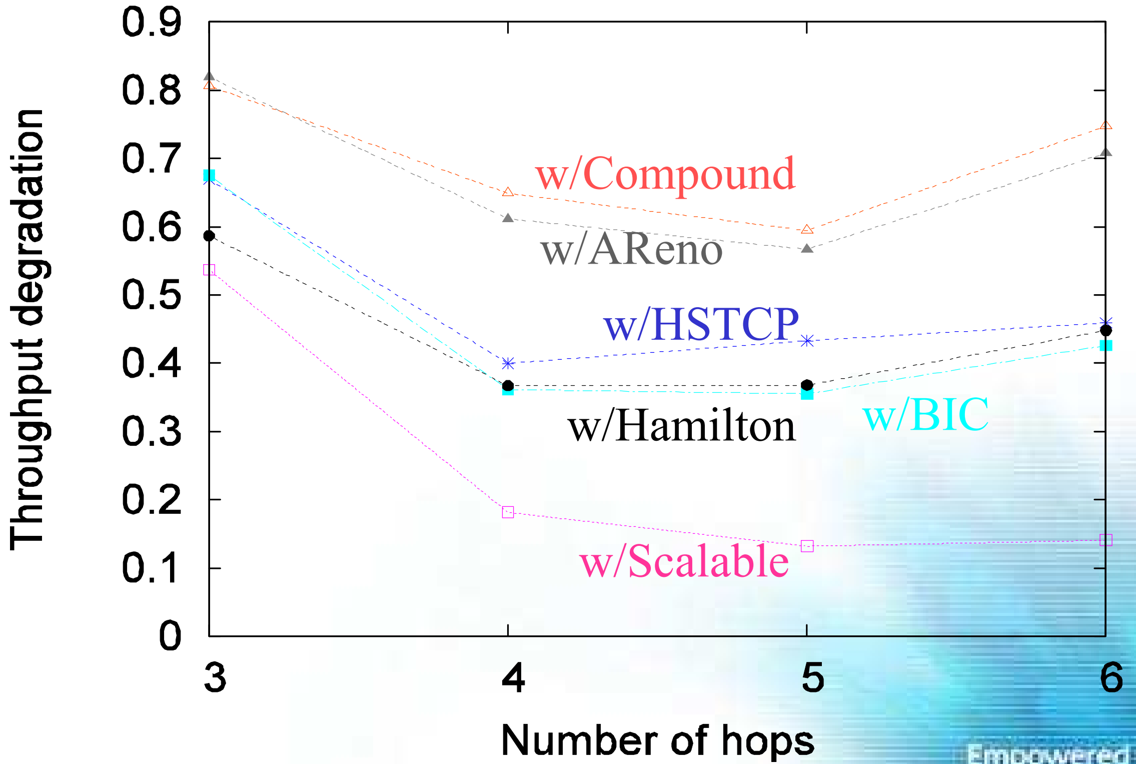
Friendliness to Reno (2)

Throughput degradation of Reno vs. hop-count

- Throughput degradation of Reno flows

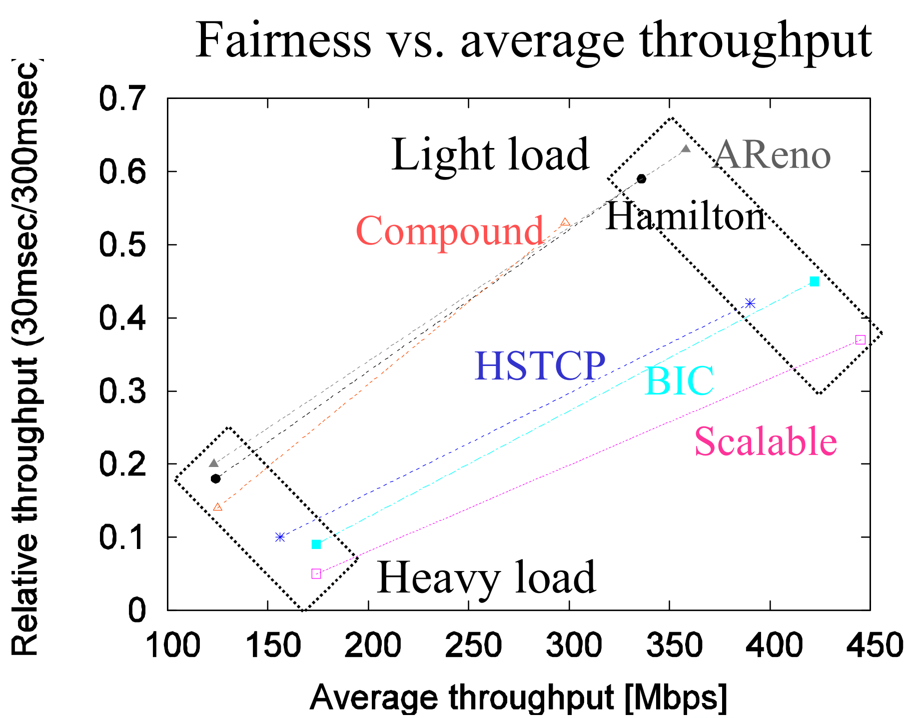
$$\text{Relative throughput} = \Sigma_{i<N} \left(\frac{\text{Throughput of flow i (coexisting with HS flows)}}{\text{Throughput of flow i (coexisting with Reno flows)}} \right) / N$$

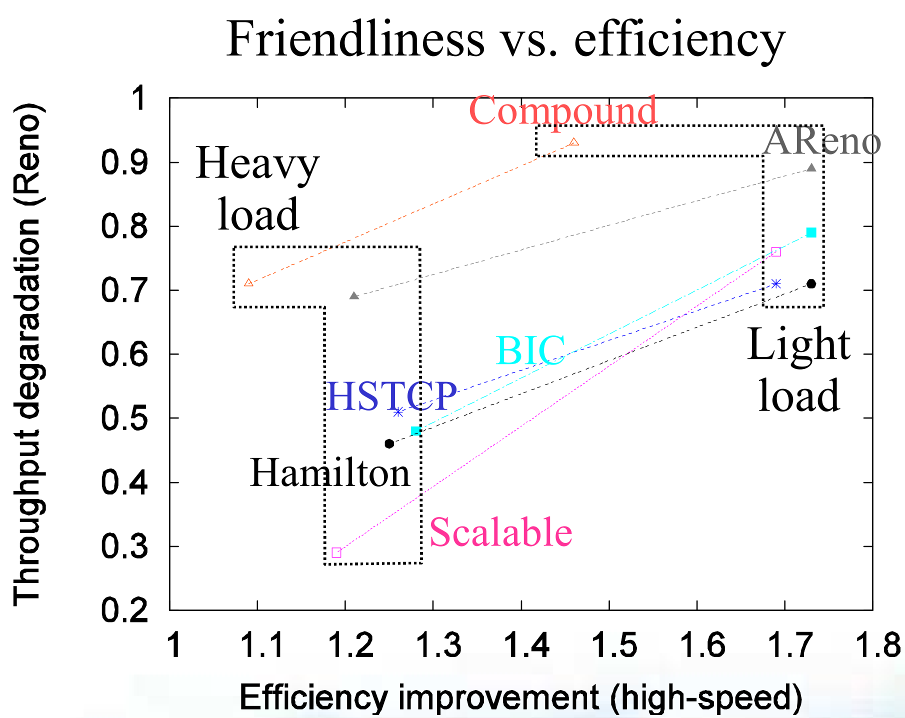
  - Indexed by coexisting high-speed flow



Empowered by Innovation NEC

Tradeoff chart





Allocate more resource on…

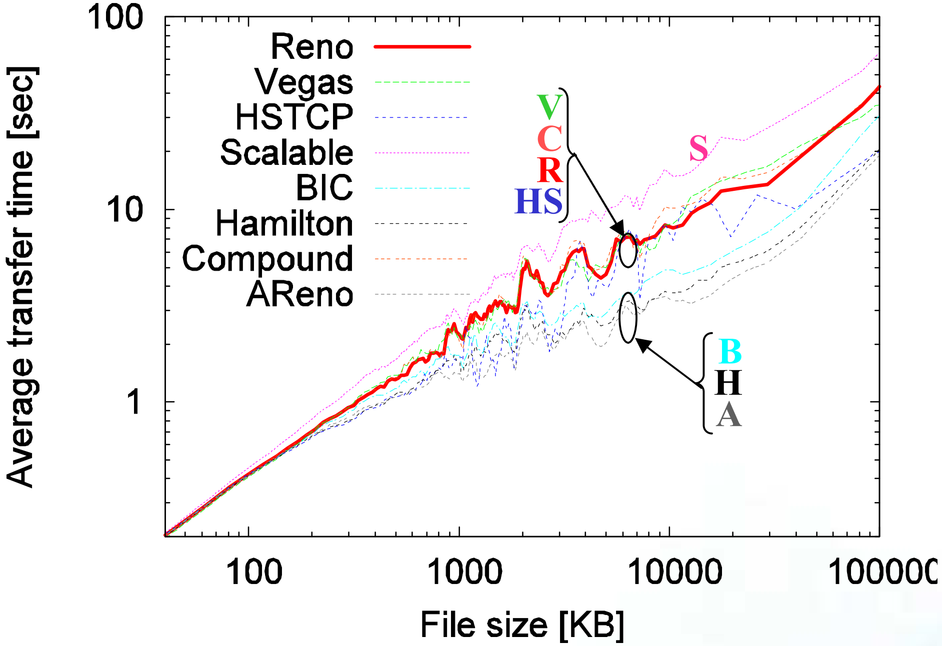
- Long-flows: AReno, Hamilton, Compound
- Short-flows: BIC, HSTCP, Scalable

- Friendliness-efficiency tradeoff: Compound vs. BIC, Scalable, HSTCP, Hamilton
- Both friendliness and efficiency: AReno

Empowered by Innovation NEC

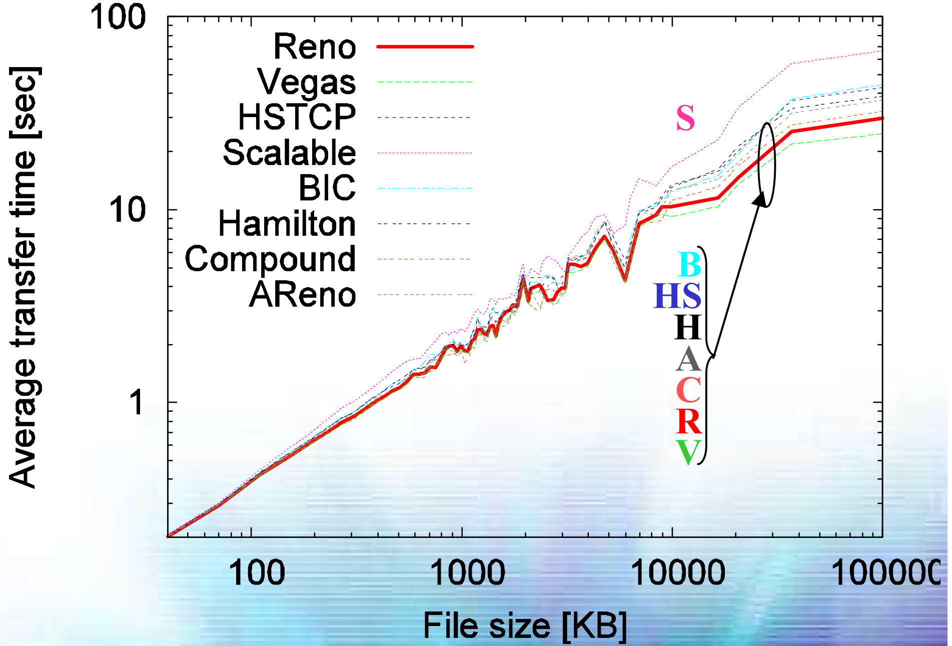
File transfer time of short-lived flows

Average transfer time of high-speed flows



Average transfer time of Reno flows

Indexed by coexisting high-speed flows



Conclusion

- It's SO time consuming
- Graphs are hard to read, sorry
- Snap-shot results, strong parameter dependency

	Per-flow throughput	Per-flow fairness	Efficient link utilization	Friendliness to Reno
High-speed TCP	X		X	
Scalable TCP	X		X	
BIC	X		X	
Hamilton-TCP		X	X	
Compound-TCP				X
TCP-AReno		X	X	X

- Future work: Linux experiment (partly done today)

With/without random packet losses

•10 long-lived flows, 100 short-lived flows