ABS: Adaptive Buffer Sizing for Heterogeneous Networks

Presented by A. L. Narasimha Reddy

Yueping Zhang and Dmitri Loguinov

Texas A&M University
College Station, TX 77843

June 2nd, 2008

Agenda

- Introduction
  - Overview of existing buffer sizing rules
- Adaptive Buffer Sizing (ABS)
  - Motivation
  - Basic control design
  - Adaptive parameter training
- Simulations
- Conclusions

Why Does Buffer Sizing Matter?

- I/O buffer is one of the key components of Internet routers, in that it
  - Absorbs transient burstiness in packet arrivals
  - Provides certain performance guarantees, such as packet loss rate, queuing delay, and link utilization
- Improperly sized router buffers can impose an adverse impact on the system’s performance
- However, there is no consensus on how to determine the optimal buffer size given a system configuration

Existing Criteria – Rule-of-Thumb

- The rule-of-thumb (Villamizar *et al.* 1994) suggests that the buffer size b be at least the product of link capacity C and average RTT R
- This classic principle has the following limitations
  - It is derived from scenarios where only synchronized long-lived flows are present, which rarely happens in real Internet routers
  - As link speed increases, the amount of memory space required by this rule becomes progressively more unrealistic

Existing Criteria – Small Buffer Rules I

- In Internet core routers, the aggregate window size process converges to a Gaussian process
- Based on this assumption, Appenseller *et al.* prove that when router buffers are sized to $b = CR/N^{0.5}$, link utilization is lower bounded by 98.99%
- Utilizing optimization theory, Avrachenkov *et al.* derive the optimal buffer size of N unsynchronized TCP flows to be $b = (CR)^2/32N^3$
- Both principles deviate from the rule-of-thumb in that they suggest that b should scale inversely proportional to N

Existing Criteria – Small Buffer Rules II

- The small-buffer rules are further extended by Enachescu *et al.*, who suggest that buffers be 10-20 packets if TCP senders implement *Paced TCP*
- All small-buffer criteria assume Poisson arrivals
  - This may be sound for backbone routers, but is not valid for general Internet routers
- In addition, these rules are obtained with an aim to achieve high link utilization
  - They do not consider other performance metrics, such as queuing delay and packet loss rate

Existing Criteria – BSCL

- To address these issues, Dovrolis *et al.* propose a set of buffer-sizing rules called *Buffer Sizing for Congested Links* (BSCL)

- Under utilization constraint, the minimum buffer is

$$b = \frac{p(N)CR - 2SN(1 - p(N))}{2 - p(N)}$$

Fraction of flows that see at least one packet loss

- Under loss rate constraint, the buffer size is

$$b = 0.87N/\sqrt{p^*} - CR_p$$

propagation delay

target loss rate

  - In contrast to small-buffer rules, this formula indicates that buffer size should be proportional to flow population N

- If both constraints are in effect, buffer size should be the larger of the above two

Existing Criteria – ADT

- Another method *Adaptive Drop Tail* *(ADT)* proposed by Rade *et al.* formulates the relationship between buffer size and utilization as a sector-bounded nonlinearity and employs the following dynamic buffer sizing algorithm

$$b(n) = b(n-1) + K(u^* - u(n))$$

  where K is an unspecified parameter satisfying $K \in (0,\ 2/k_2)$ and k_2 is the sector nonlinearity upper bound

- It is unclear how to set K and k_2 in practice

Existing Criteria – Summary

- All buffer-sizing rules established so far are based on certain explicit modeling of the Internet traffic
  - But the Internet is such a complex system that its dynamics are difficult, if ever possible, to precisely model
- Existing work (McKeown *et al.*) concludes that it is premature to deploy any existing buffer-sizing criteria without a comprehensive theoretical tool that incorporates
  - all traffic patterns
  - network topologies
  - router architectures
  - transient and stationary system dynamics
  - proper performance metrics

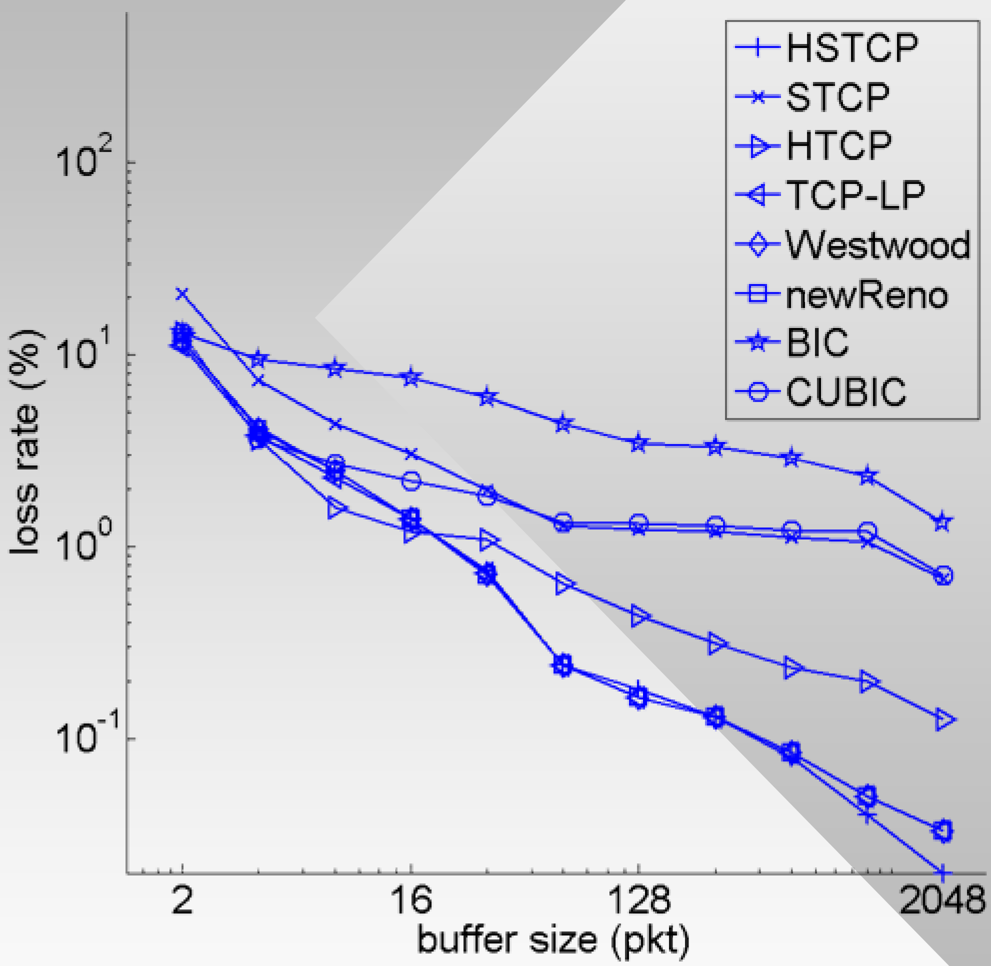
Objectives of the Presented Work

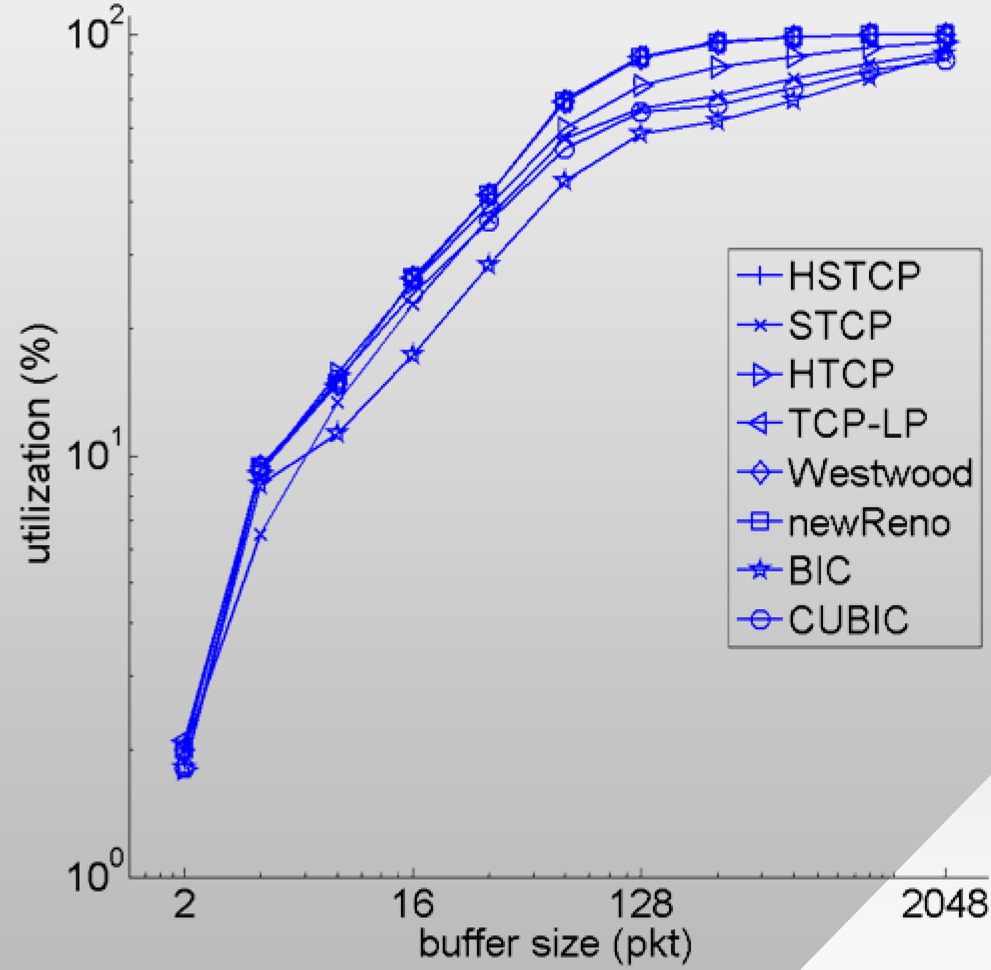
- Can we achieve the goal of buffer sizing without comprehensive knowledge of Internet dynamics?
- Can we design simple yet robust buffer-sizing methods under generic Internet traffic (i.e., mixtures of long- and short-lived, TCP and non-TCP flows)?
- Can we incorporate multiple performance metrics in one buffer-sizing mechanism?
- Can we develop a technique that adapts the buffer-sizing scheme to dynamically-changing Internet traffic?

Agenda

- Introduction
  - Overview of existing buffer sizing rules
- Adaptive Buffer Sizing (ABS)
  - Motivation
  - Basic control design
  - Adaptive parameter training
- Simulations
- Conclusions

Motivation – Simulation Illustration





- For all these protocols, their loss rate p decreases and utilization u increases as buffer size b grows

Motivation – Intuitive Explanation

- Intuitively, the relationship between buffer size and loss rate and between buffer size and utilization should be monotonic
- A large buffer can
  - absorb more bursts in packet arrivals, there by reducing the frequency of packet drops
  - allows the bottleneck link to sustain full utilization for a longer time, thereby increasing average link utilization
- The paper proves this monotonic relationship in a simple, yet generic, congestion control model
- Leveraging this result, we next design an adaptive buffer sizing scheme, called ABS

ABS Design I

- Consider only the utilization constraint u^*
- We use an Integral controller

$$b_u(n) = b_u(n-1) - I_u T\big(u(n) - u^*\big)$$

utilization constraint (u in b_u); sampling interval (T); Integral gain (I_u)

- However, this controller may have serious problems in non-bottleneck routers
  - If u^* is set above the maximally achievable utilization of the router, $u(n) - u^*$ is always negative
  - The buffer size will be driven to infinity

ABS Design II

- Our solution is to introduce a damping term to mitigate the effect of term $u(n) - u^*$
- The new controller is given as

$$b_u(n) = b_u(n-1) - I_u T\Big(u(n) - u(n-1)\Big)\Big(u(n) - u^*\Big)$$

- In the steady state of a non-bottleneck router, $u(n) - u(n-1) = 0$, forcing buffer $b_u(n)$ to converge to its equilibrium value
- The controller for buffer $b_p(n)$ under the packet loss constraint p^* can be obtained similarly

ABS Design III

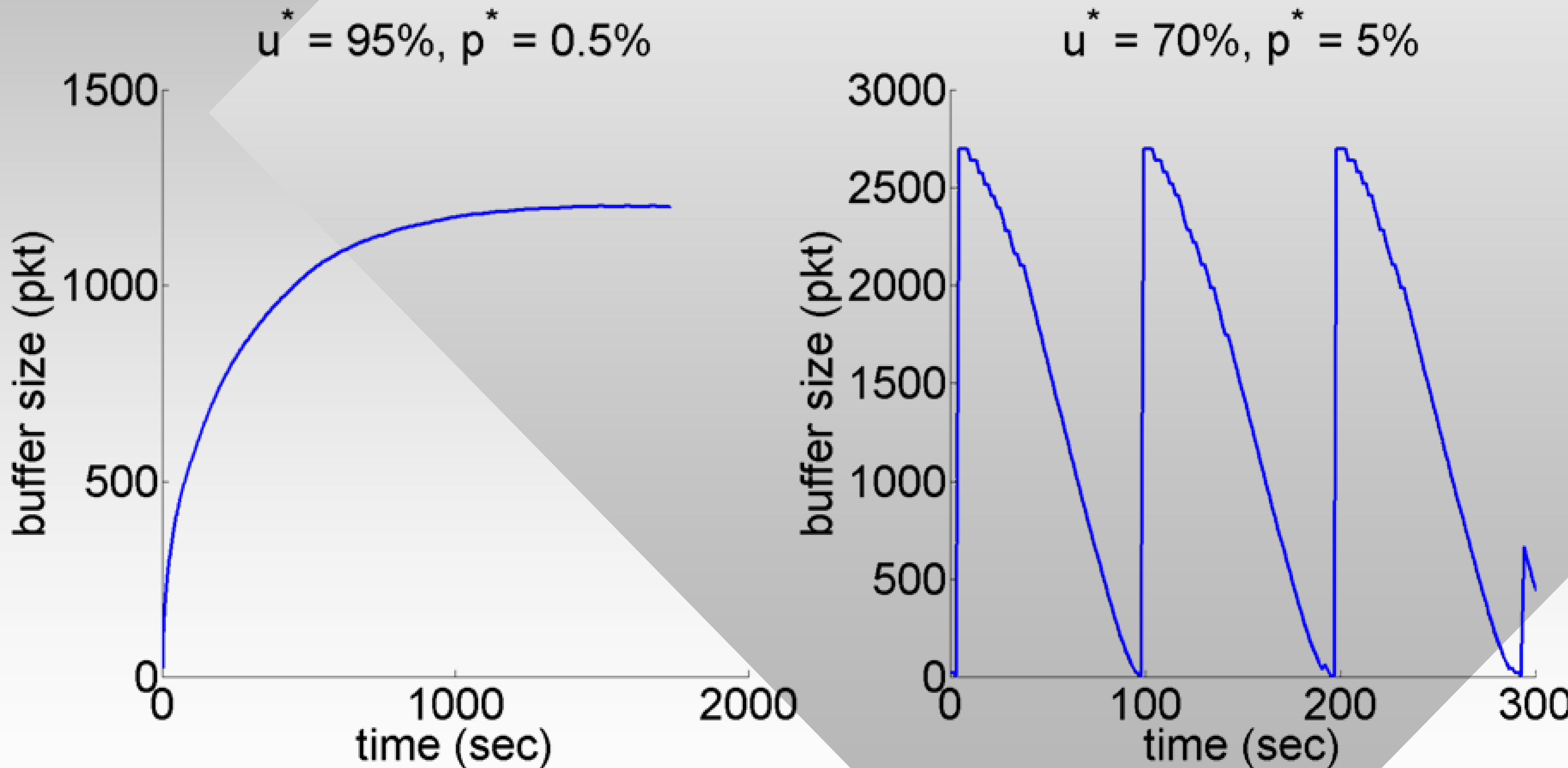
- Buffer size $b(n)$ satisfying both constraints is

$$b(n) = \max\big(b_u(n), b_p(n)\big)$$

- The resulting controller is called *Adaptive Buffer Sizing* (ABS) and its sub-controllers under utilization and loss constraints are denoted by ABS_u and ABS_p
- However, it is still unclear how to choose optimal gain parameters I_u and I_p
  - If they are chosen too small, the system may suffer from a sluggish convergence rate to the equilibrium
  - If they are set too large, the system may exhibit exceedingly aggressive adaptation behavior and persistently oscillation

ABS Design IV

- To illustrate this problem, consider ns2 simulations where $I_u = I_p = 3000$, $C = 10$ mb/s, and $N = 20$



Control constants I_u and I_p must depend on C, N, $u*$, $p*$ and the underlying ingress traffic

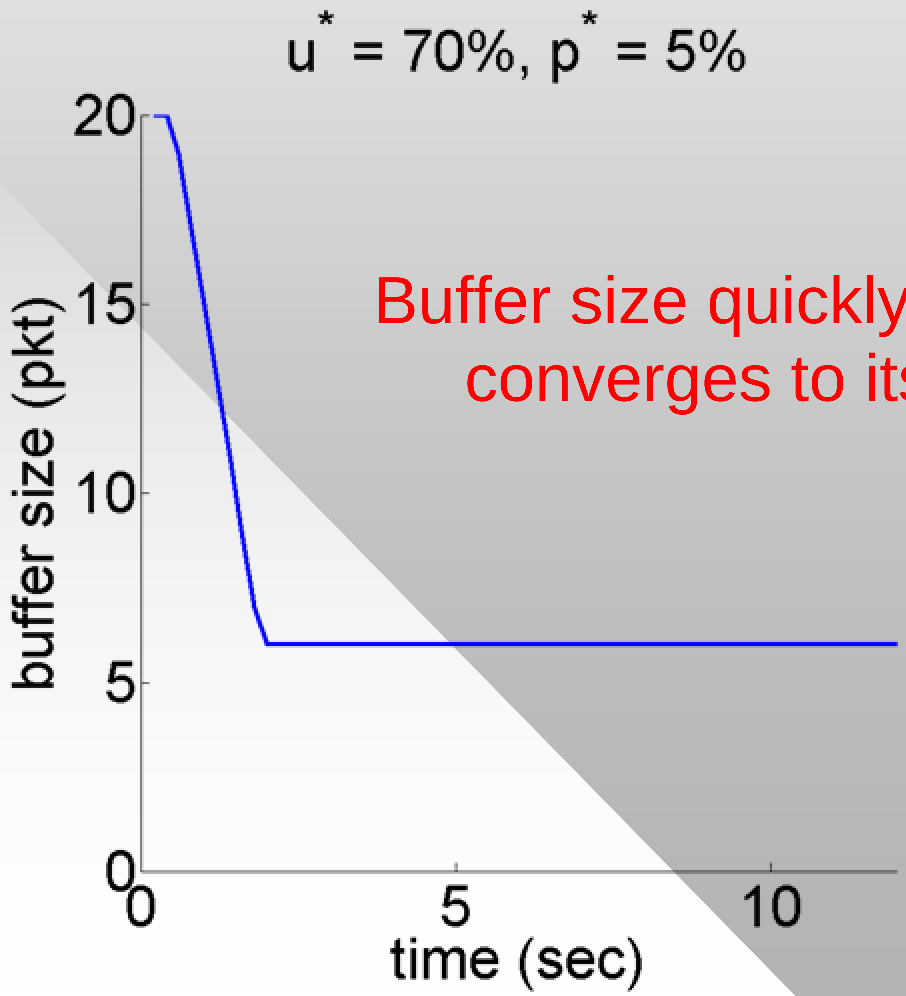
Adaptive Parameter Training I

- Since Internet traffic model is unknown, it is unlikely that any off-line parameter selection can be effective
- We solve this problem by developing a parameter training mechanism, which dynamically finds the control gains I_u and I_p that are most suitable for the underlying traffic
- This is accomplished by a combination of the output error and gradient descent methods
- Then, the final control equation of buffer size under the utilization constraint becomes

$$I_u(n+1) = I_u(n) - \gamma T\Big(u(n+1) - u(n)\Big)\Big(u(n+1) - u^*\Big)$$

Adaptive Parameter Training V

- The equation for $I_p(n)$ can be derived similarly
- Rerun the ns2 simulation with parameter training



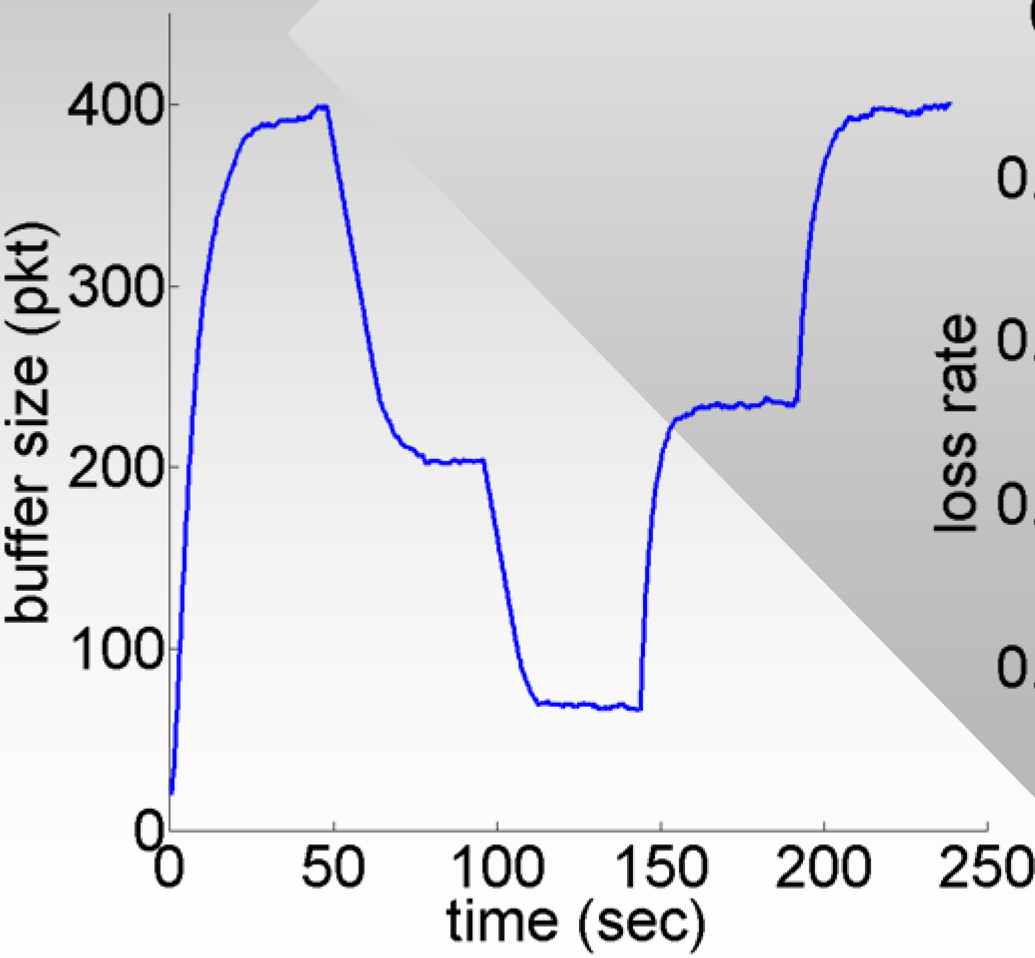
Buffer size quickly and monotonically converges to its equilibrium point

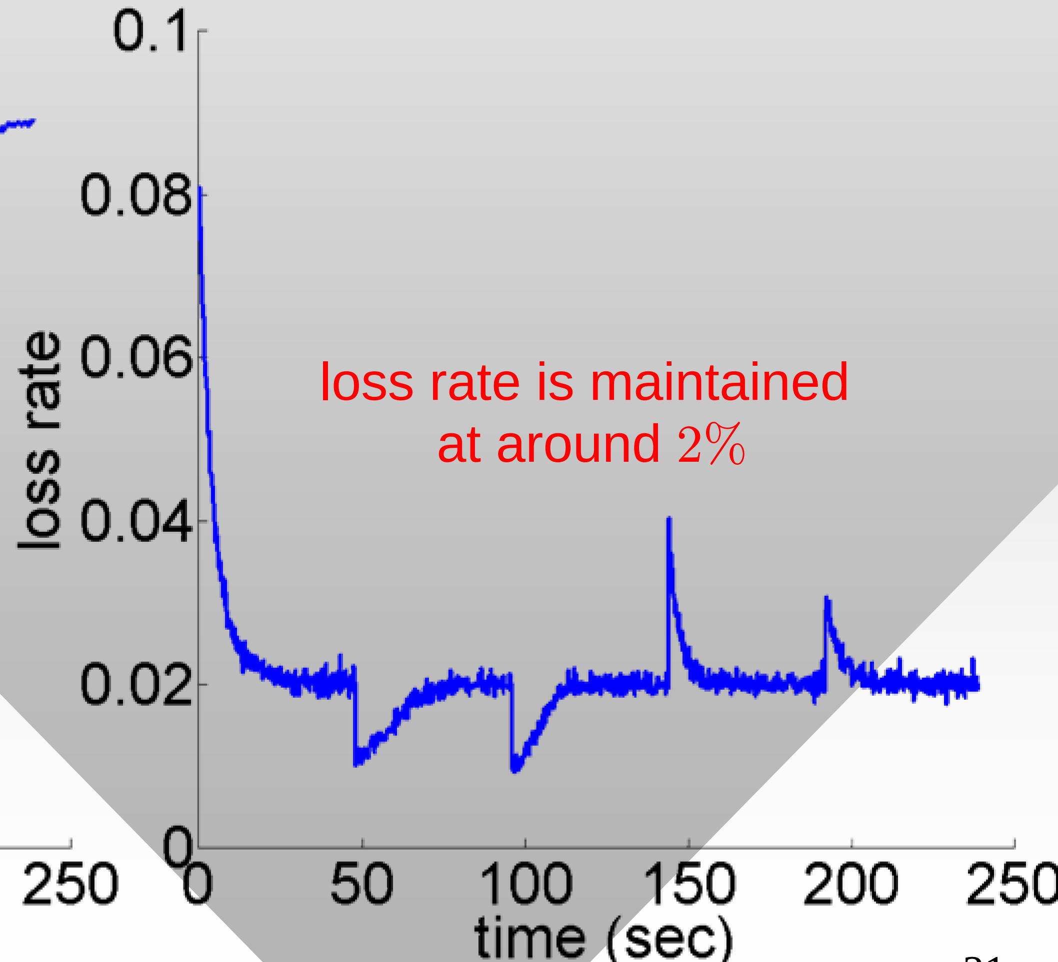
Agenda

- Introduction
  - Overview of existing buffer sizing rules
- Adaptive Buffer Sizing (ABS)
  - Motivation
  - Basic control design
  - Adaptive parameter training
- Simulations
- Conclusions

Simulations – Load Changes

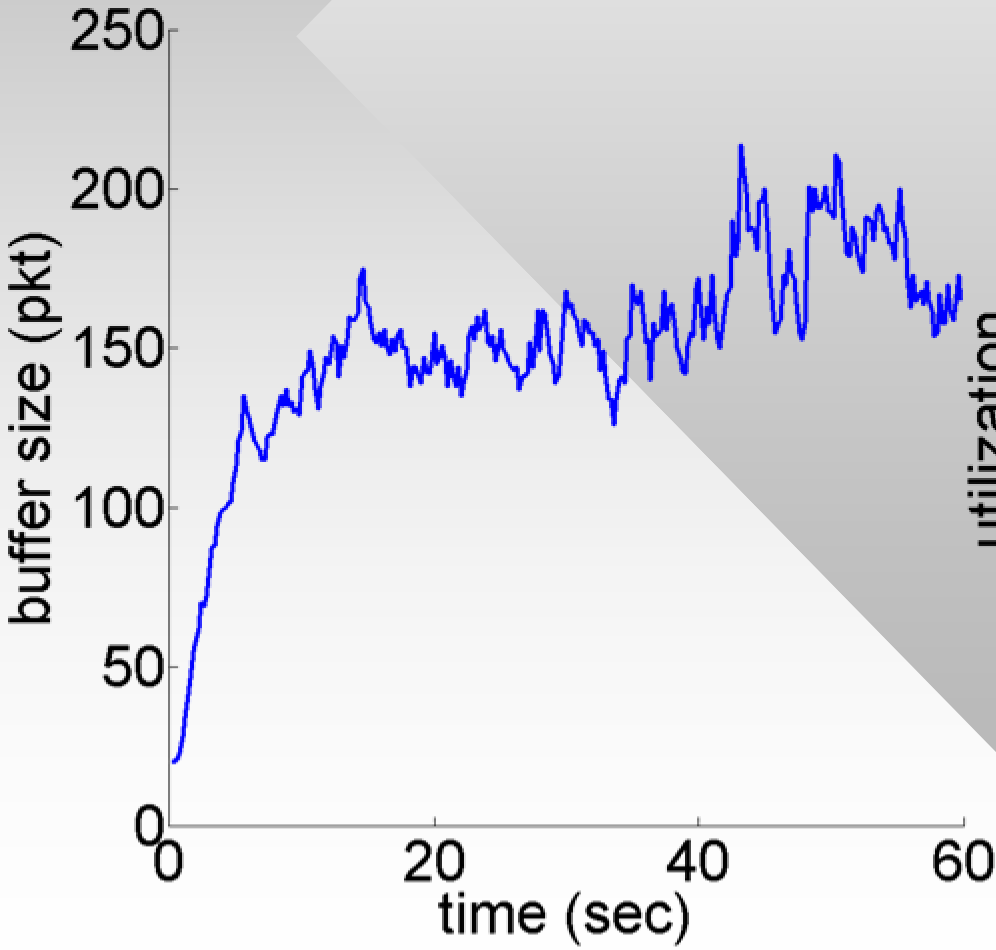
- Set $u^* = 90\%$ and $p^* = 2\%$
- Flows frequently join and leave the system

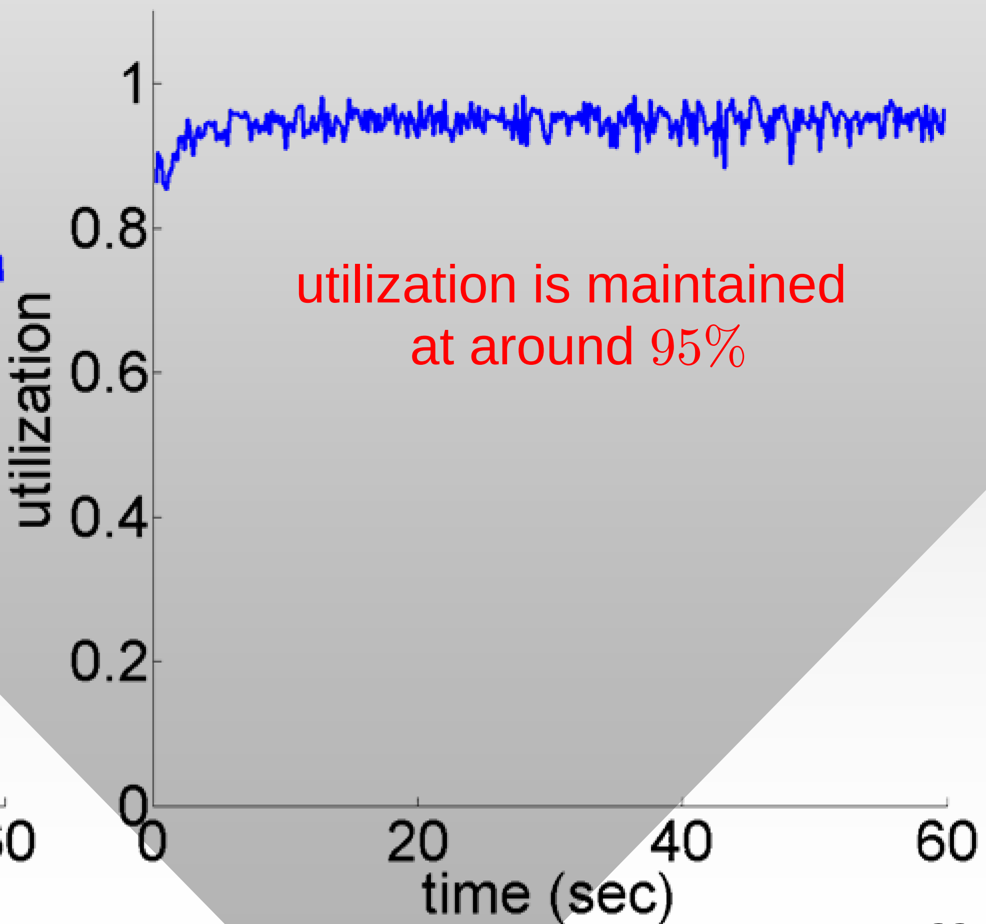




Simulations – Web Traffic

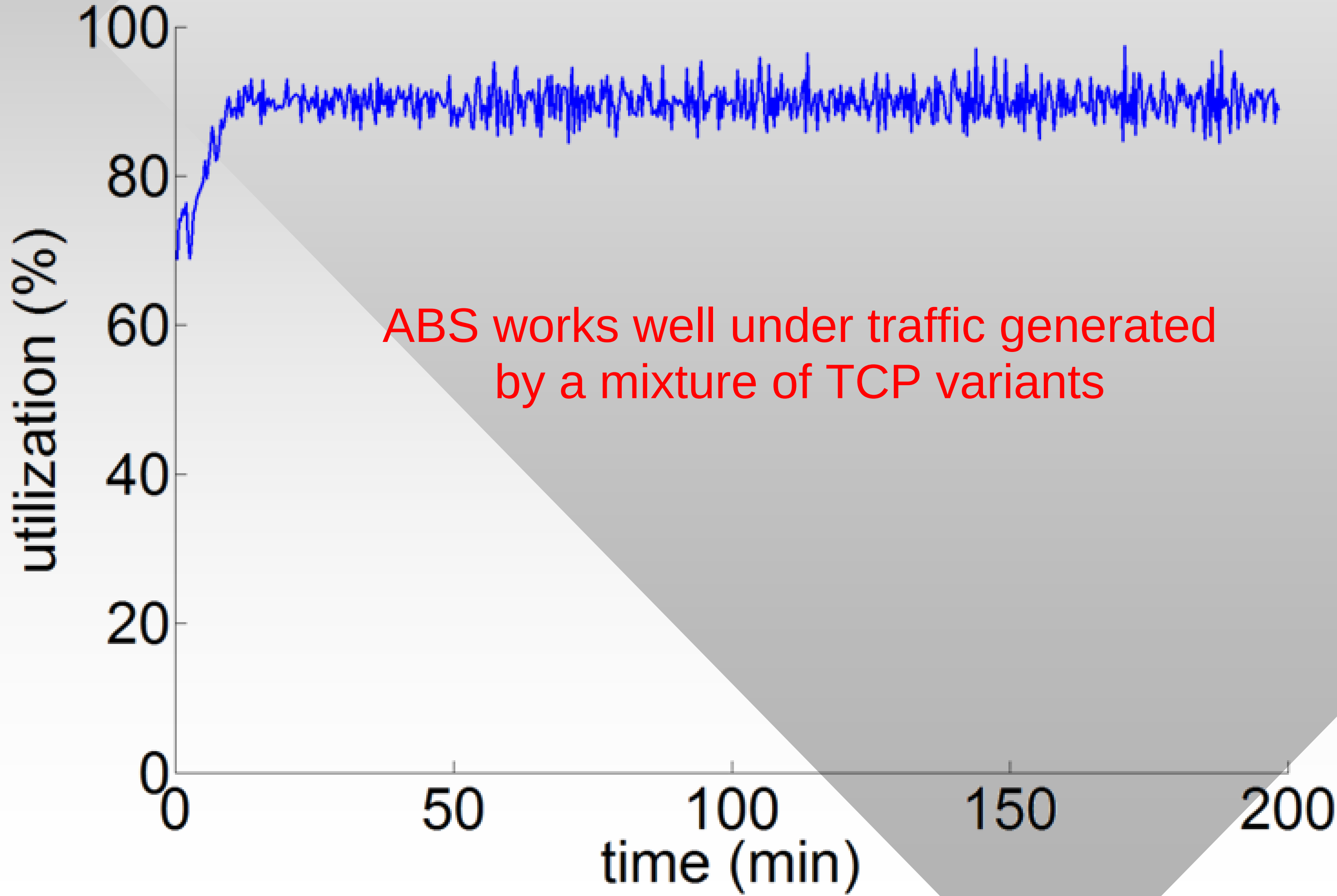
- Set $u^* = 95\%$ and $p^* = 1\%$
- Mice traffic generated by 100,000 HTTP sessions





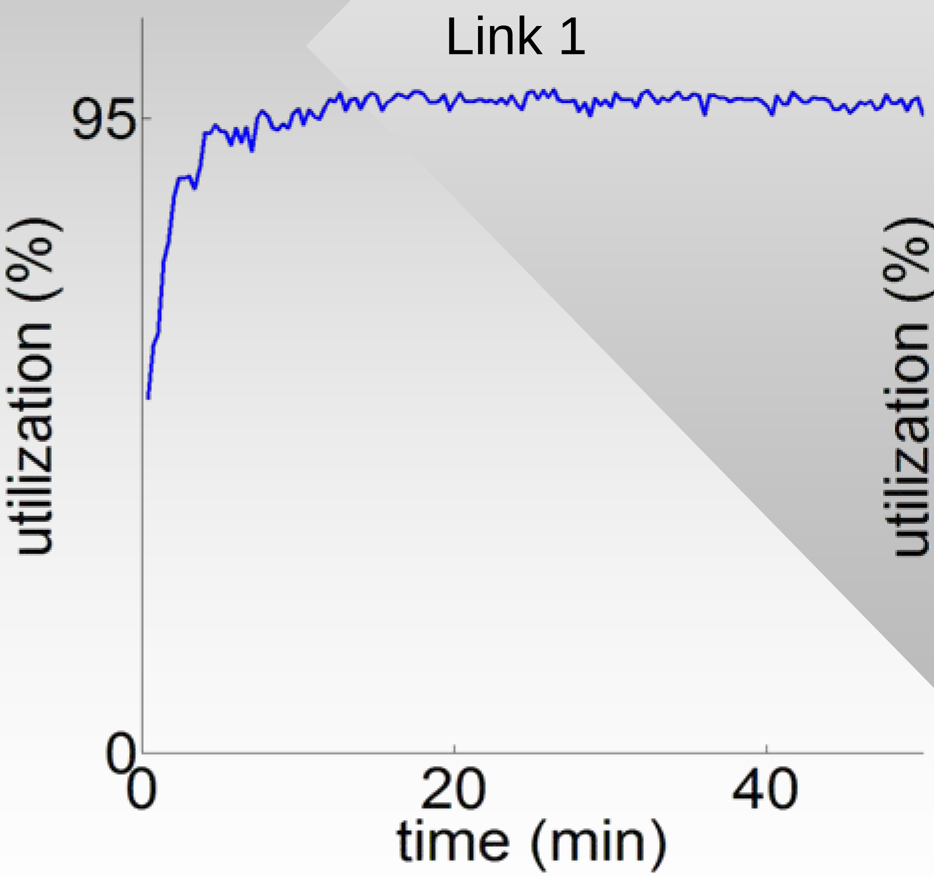
Simulations – TCP Variants

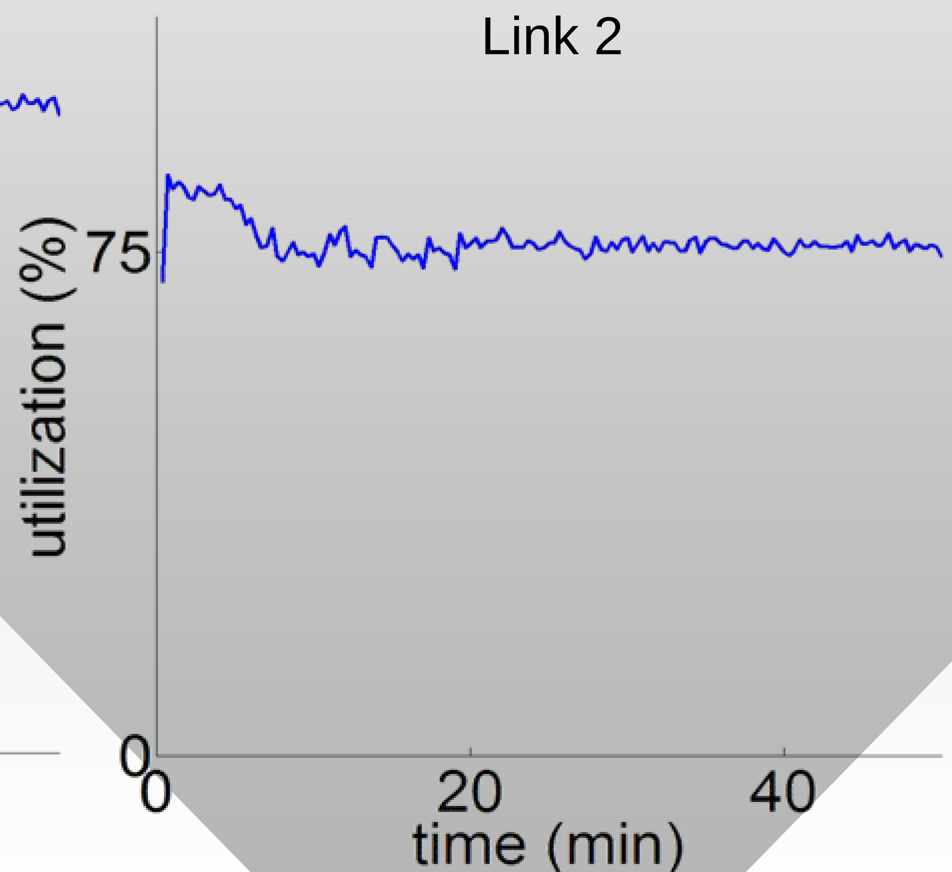
- A single link of capacity 100 mb/s shared by 10 Reno, 10 HSTCP, 10 STCP, 10 HTCP, and 10 Westwood flows ($u^* = 90\%$)



Simulations – Multi-Link Topology

- Two-link "parking lot" topology
- Target link utilization is $p_1{}^* = 95\%$ and $p_2{}^* = 75\%$





Utilization is maintained at the target level of each link

Conclusions

- In this paper, we presented a new buffer sizing scheme called ABS, which can dynamically choose the smallest buffer size satisfying the given performance constraints
- In contrast to existing approaches, ABS does not rely on any explicit formulation of Internet traffic
- ABS performs well under generic Internet traffic composed of short/long TCP and non-TCP flows
- Future work involves
  - Implementing ABS in real systems and testing it in the Internet
  - Simplifying ABS