Design and Evaluation of New Power Management Methods to Reduce Direct and Induced Energy Use of the Internet

Chamara Gunaratne
Department of Computer Science and Engineering
University of South Florida
Tampa, FL 33620
pgunarat@cse.usf.edu

This material is based upon work funded by the National Science Foundation under grant CNS-0520081.

Thank you!

I would like to thank Dr. Scott Campbell
for chairing this defense.

I would like to thank Dr. Ken Christensen, Dr. Rafael Perez,
Dr. Miguel Labrador, Dr. Stephen Suen, and Dr. Hüseyin Arslan
for serving on my committee.

I would like to thank Bruce Nordman from
Lawrence Berkeley National Laboratory.

I would like to thank Himanshu Anand, Mukind Kanyana,
and Pradeep Purushothaman from the
University of Florida.

Agenda

- Introduction
- Reducing direct power consumption
- Reducing induced power consumption
- Contributions

Agenda

- Introduction
- Reducing direct power consumption
- Reducing induced power consumption
- Contributions

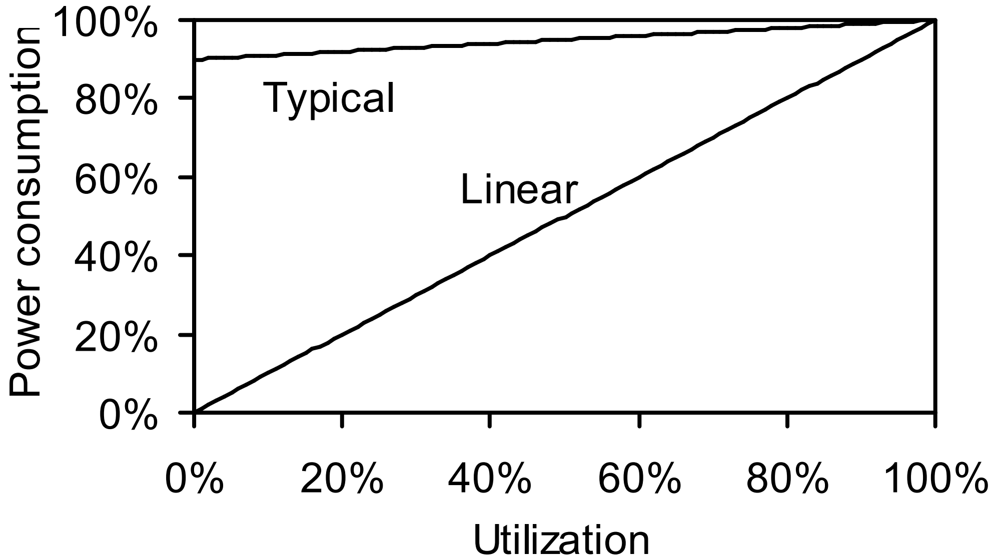
Introduction

- **This is the first investigation into reducing the power consumption of Ethernet links and network hosts**

- **Power consumption can be classified as:**
  - ***Direct* power consumption**
    - Power consumed by the network infrastructure of the Internet (e.g., links, switches, and routers)
  - ***Induced* power consumption**
    - Power consumed by network hosts that are inactive, but remain fully powered-on in order to maintain network connectivity

Direct power consumption

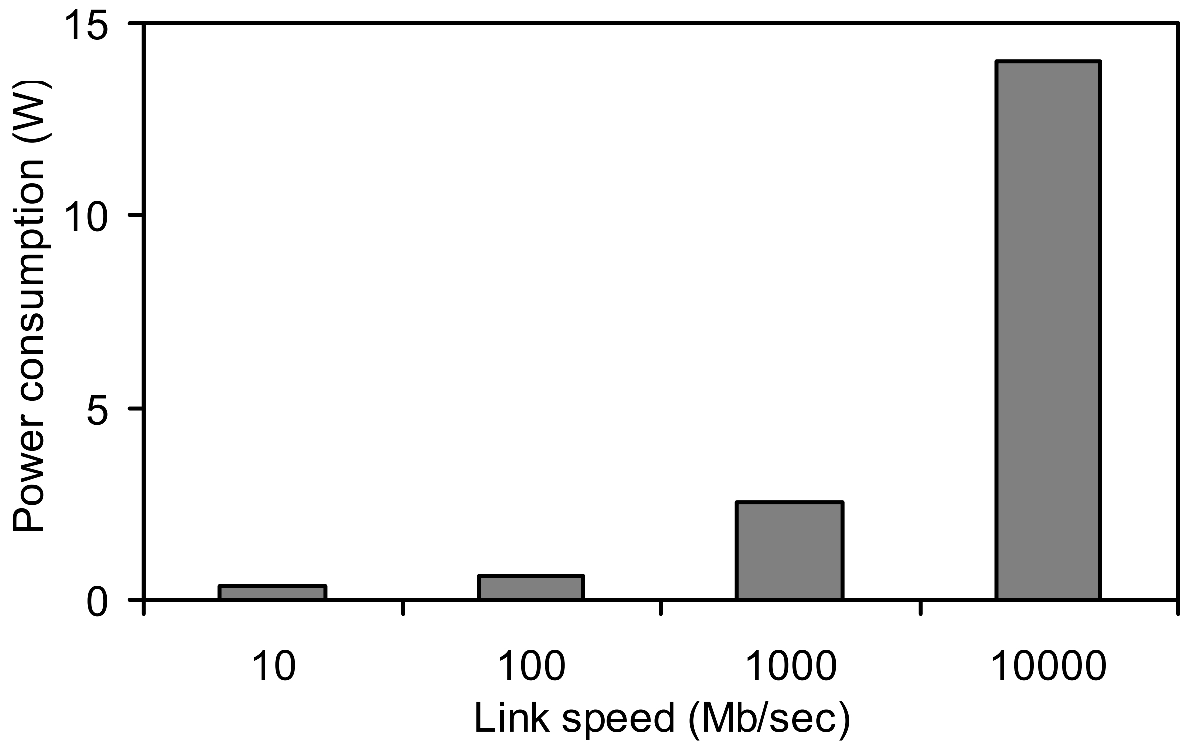
- **The Internet is a major consumer of electricity**
  - Estimated to be 7.5 TWh/yr in 2005 [108]
- **Average network utilization is low**
  - Typically less than 5% for edge Ethernet links
- **Power consumption is not linear with utilization**



Not in dissertation

Direct power consumption continued

- **Power consumption is increasing super-linearly**
  - Measurements of Ethernet NICs (average power consumption)



Induced power consumption

- **Network hosts are a major consumer of electricity**
  - Estimated to be 74 TWh/yr in 2001 [81]
- **Most desktop PCs in offices are powered-on 24/7**
  - Office desktop PCs are idle most of the time
  - 5% or less had power management enabled [105, 126]
- **Use of power management would save 17 TWh/yr**
  - Majority of savings would come from desktop PCs

Agenda

- Introduction
- **Reducing direct power consumption**
- Reducing induced power consumption
- Contributions

Adaptive Link Rate (ALR)

New contribution

- **Match Ethernet link rate to link utilization**
  – Energy is saved by operating at low link data rate
  – Consists of a *mechanism* and *control policy*

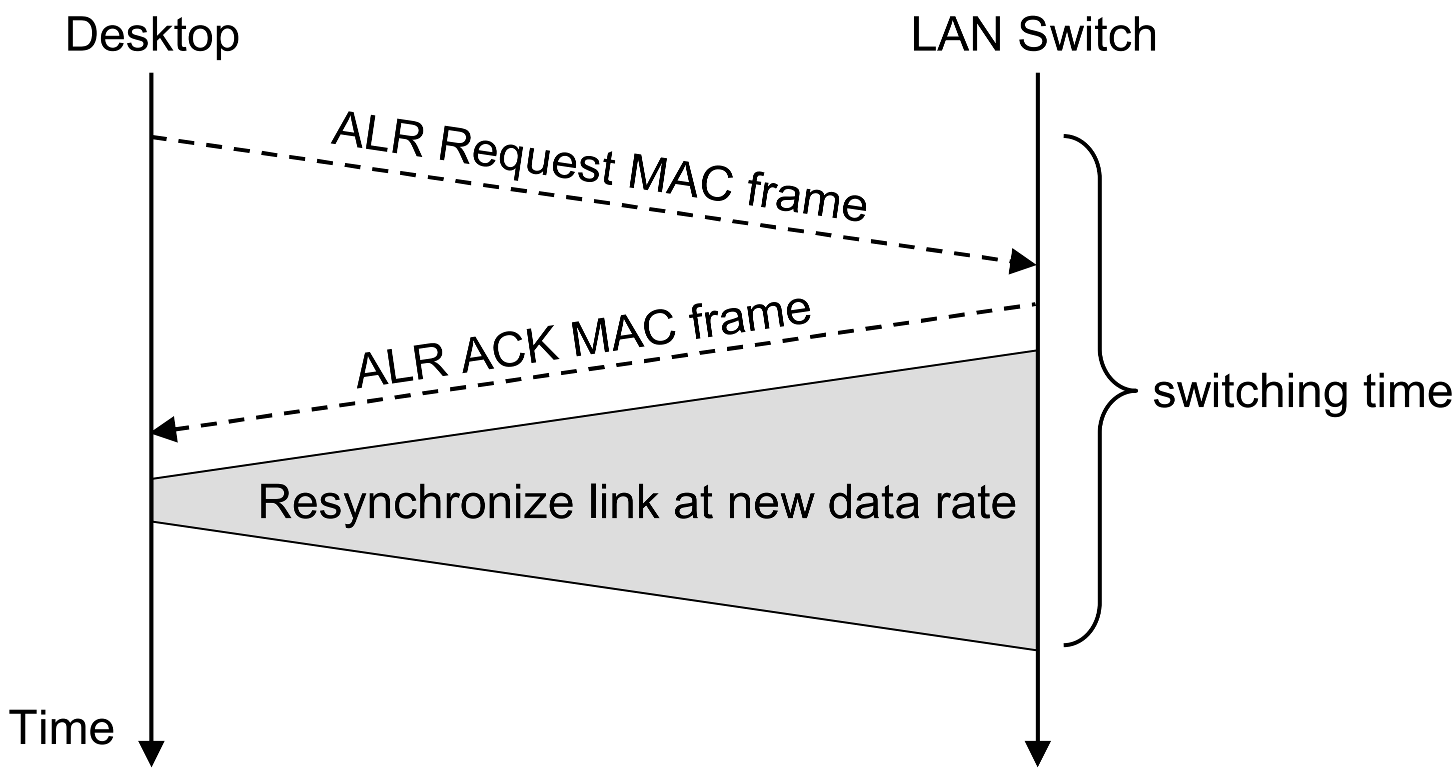
- **Mechanism**
  – How to switch link rate

- **Control policy**
  – When to switch link rate

- **ALR has a fundamental trade-off**
  – Packet delay vs. energy saved

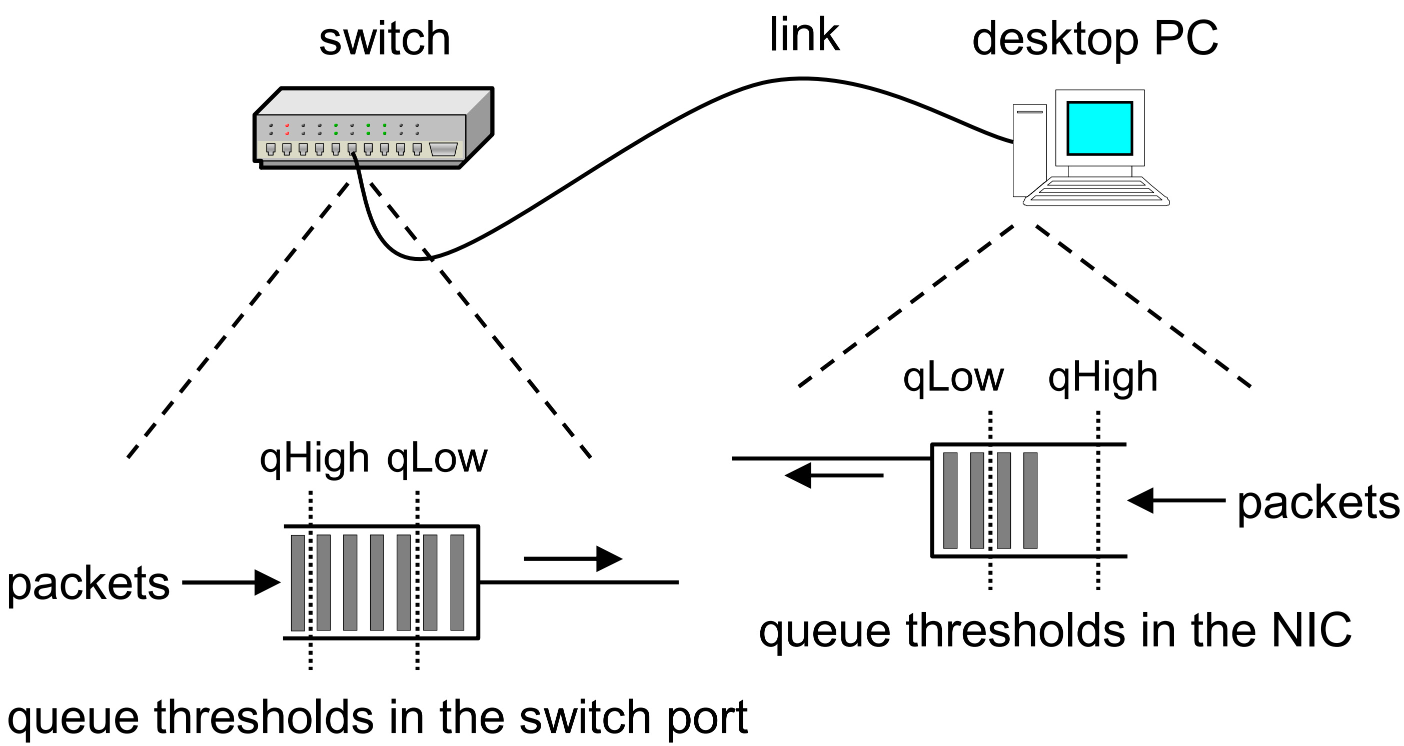
ALR mechanism

- **Mechanism implements link rate switching**
  - ALR must be implemented in both ends of an Ethernet link



ALR policy

- **Dual-threshold policy**
  - If queue is above qHigh then switch to high rate
  - If queue is below qLow then switch to low rate



Evaluation of dual-threshold policy

- **Control variables**
  - Threshold values
  - Utilization
  - Time to switch between link rates
- **Response variables**
  - Mean response time (mean packet delay)
  - Percentage of time in low data rate (energy saving)
  - Frequency of rate switches
- **First, Markov models were created**
  - Poisson arrivals and exponential service times assumed
- **Next, simulation models were created**
  - Actual traced Ethernet traffic used

Markov models

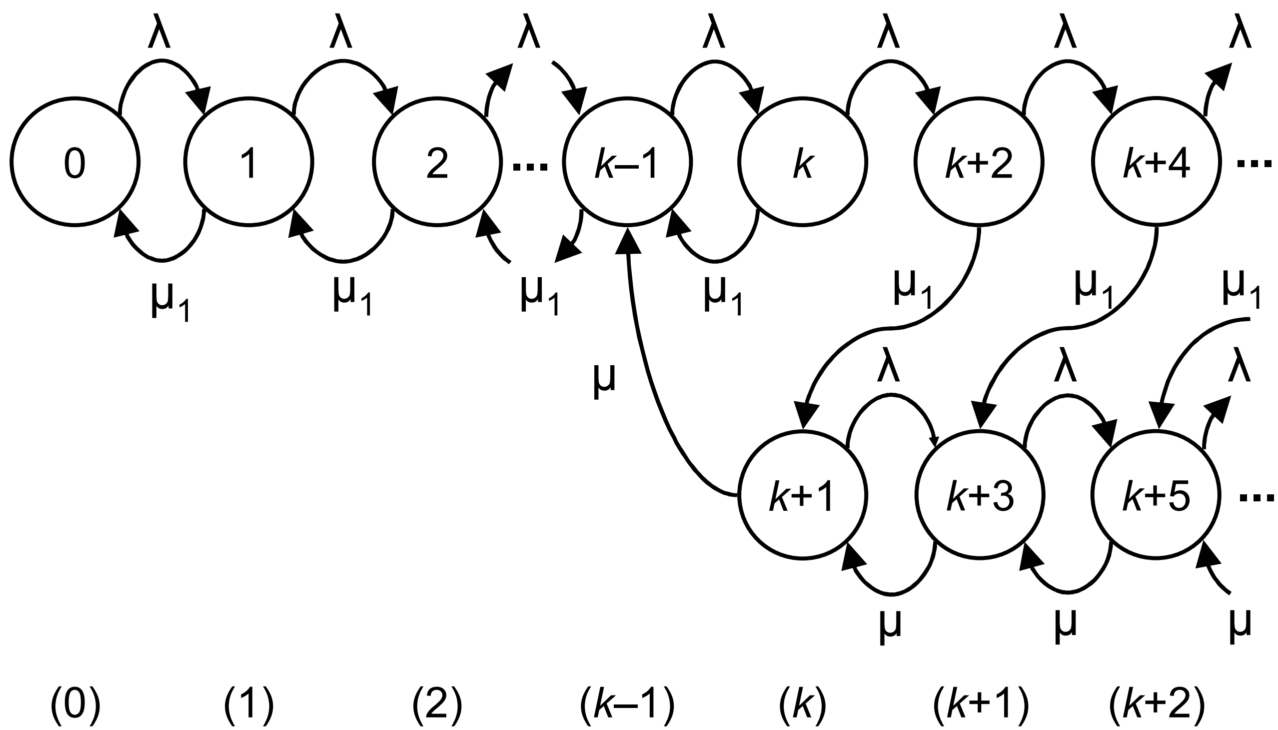
- **The following was modeled**
  - One and two thresholds
  - Rate switch *during service* and at *completion of service*
    - Rate switching time is zero

- **Switch at *completion of service* models packets**
  - Rate cannot switch until current packet is fully transmitted

- **Four Markov models**
  - #1: Single-threshold, rate switch during service
  - #2: Dual-threshold, rate switch during service
  - #3: Single-threshold, rate switch at completion of service
  - #4: Dual-threshold, rate switch at completion of service

Markov model #3

- **Single threshold, rate switch at completion of service**
  - Discovered later that same model was developed by Chong and Zhao [22] for CPU control



Markov model #3 continued

- **Model can be solved directly using global balance**
  - Requires two recursive equations and many steps
- **Dr. Suen identified a key relationship**

$$\pi_k = \left(\frac{\lambda}{\lambda + \mu_1} \right) \pi_{k-1}$$

- **This relationship enabled a solution with less steps**
  - Requires one recursive equation and fewer steps
- **This solution method "scales" to the next model**
  - Dual thresholds (Model #4)

Markov model #3 continued

- **Direct solution is novel**
  - Mean number in system (*L*) is solved numerically

$$P_n = \begin{cases} P_0 \rho_1^n & (0 \le n < k) \\ \dfrac{P_0 \rho_1^{k-1}}{\rho + \rho/\rho_1 - 1} \left(\rho^{n-k+2} - \left(1 - \dfrac{\rho}{\rho_1}\right) \left(\dfrac{\rho_1}{1+\rho_1} \right)^{n-k+1} \right) & (n \ge k) \end{cases}$$

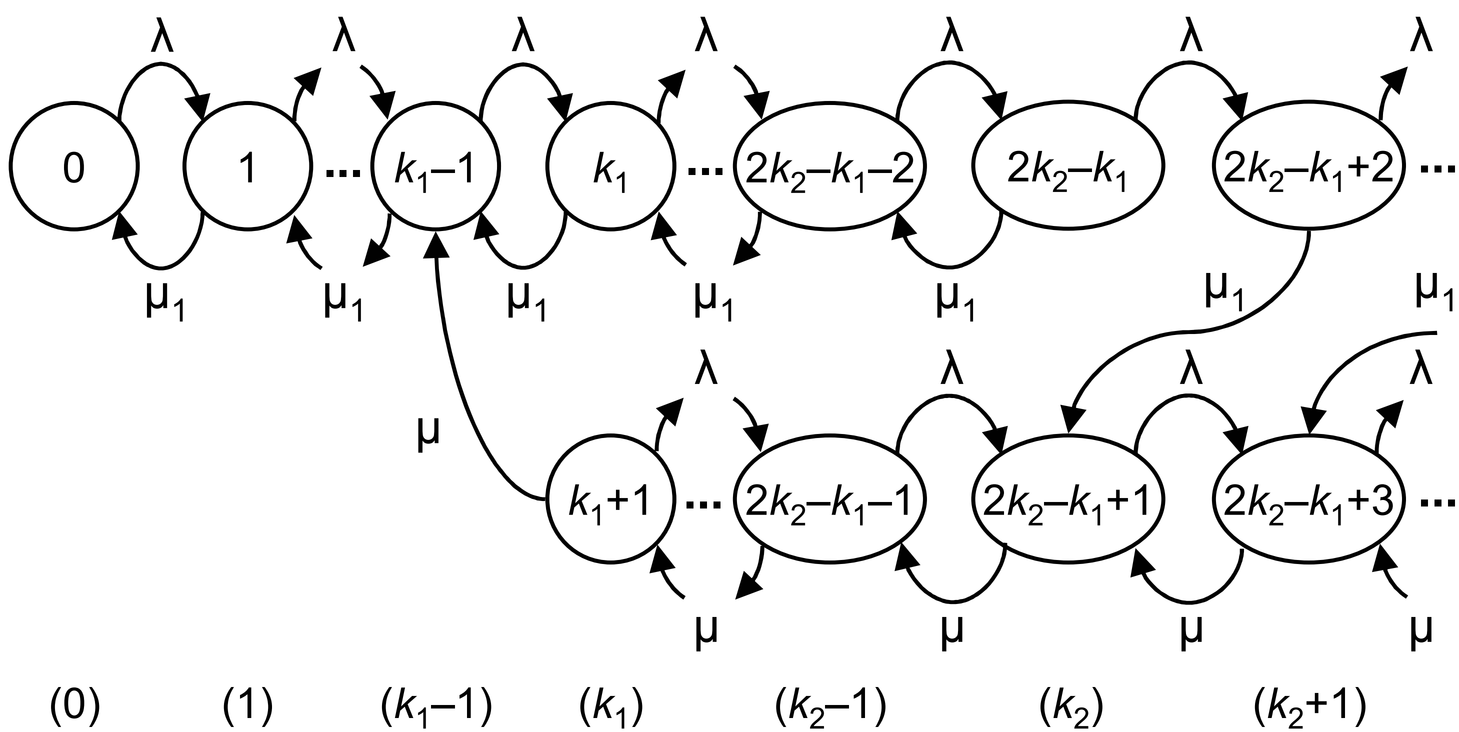
$$P_0 = \left(\frac{1-\rho_1^k}{1-\rho_1} + \frac{\rho_1^k}{1-\rho} \right)^{-1}$$

Derivation is in Chapter 5

Markov model #4

New contribution

- **Dual-threshold, rate switch at completion of service**



Markov model #4 continued

Derivation is in Appendix A

- **Direct solution was achieved**
  - Solved similarly to previous model
  - Mean number in system (*L*) is solved numerically

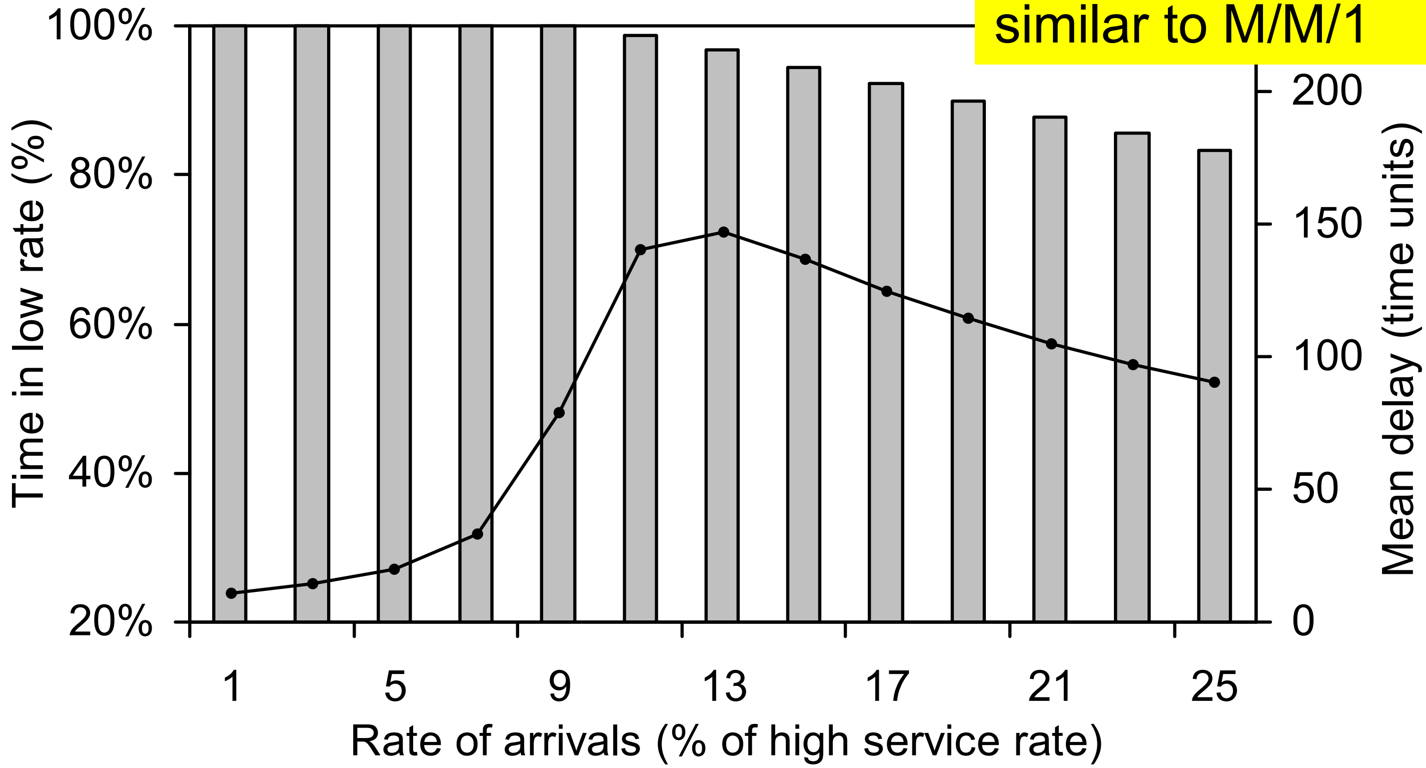
$$
P_n = \begin{cases}
P_0 \rho_1^n & (0 \le n < k_1) \\[2ex]
\dfrac{P_0 \rho_1^{k_1-1}\left(1 - 1/\rho_1\right)}{1 - 1/\rho_1^{k_2-k_1+2}} \left(\dfrac{1 - 1/\rho_1^{k_2-n+1}}{1 - 1/\rho_1} + \dfrac{\rho\left(1-\rho^{n-k_1+1}\right)}{1-\rho} \right) & (k_1 \le n < k_2) \\[2ex]
\dfrac{P_0 \rho_1^{k_1-1}}{1 - 1/\rho_1^{k_2-k_1+2}} \left(\rho^{n-k_2+1} \left(\dfrac{\left(1-1/\rho_1\right)\left(1-\rho^{k_2-k_1}\right)}{1-\rho} - \dfrac{1-1/\rho_1^2}{1-\rho-\rho/\rho_1} \right) + \dfrac{\left(1-1/\rho_1^2\right)\left(1-\rho/\rho_1\right)}{1-\rho-\rho/\rho_1} \left(\dfrac{\rho_1}{1+\rho_1} \right)^{n-k_2+1} \right) & (n \ge k_2)
\end{cases}
$$

$$
P_0 = \left(\frac{1-\rho_1^{k_1}}{1-\rho_1} + \frac{1}{1-\rho_1^{k_2-k_1+2}} \left(\frac{\rho_1^{k_2} - \rho_1^{k_1}}{\rho_1 - 1} + \frac{\rho_1^{k_2}\left(\left(k_2 - k_1\right)\left(\rho_1 - \rho\right) + \rho_1^2 - 1\right)}{\rho - 1} \right) \right)^{-1}
$$

Results from Markov model #4

- **Numerical results for switching time of zero**
  - "time unit" = mean service time at high rate

At low utilization, behavior similar to M/M/1

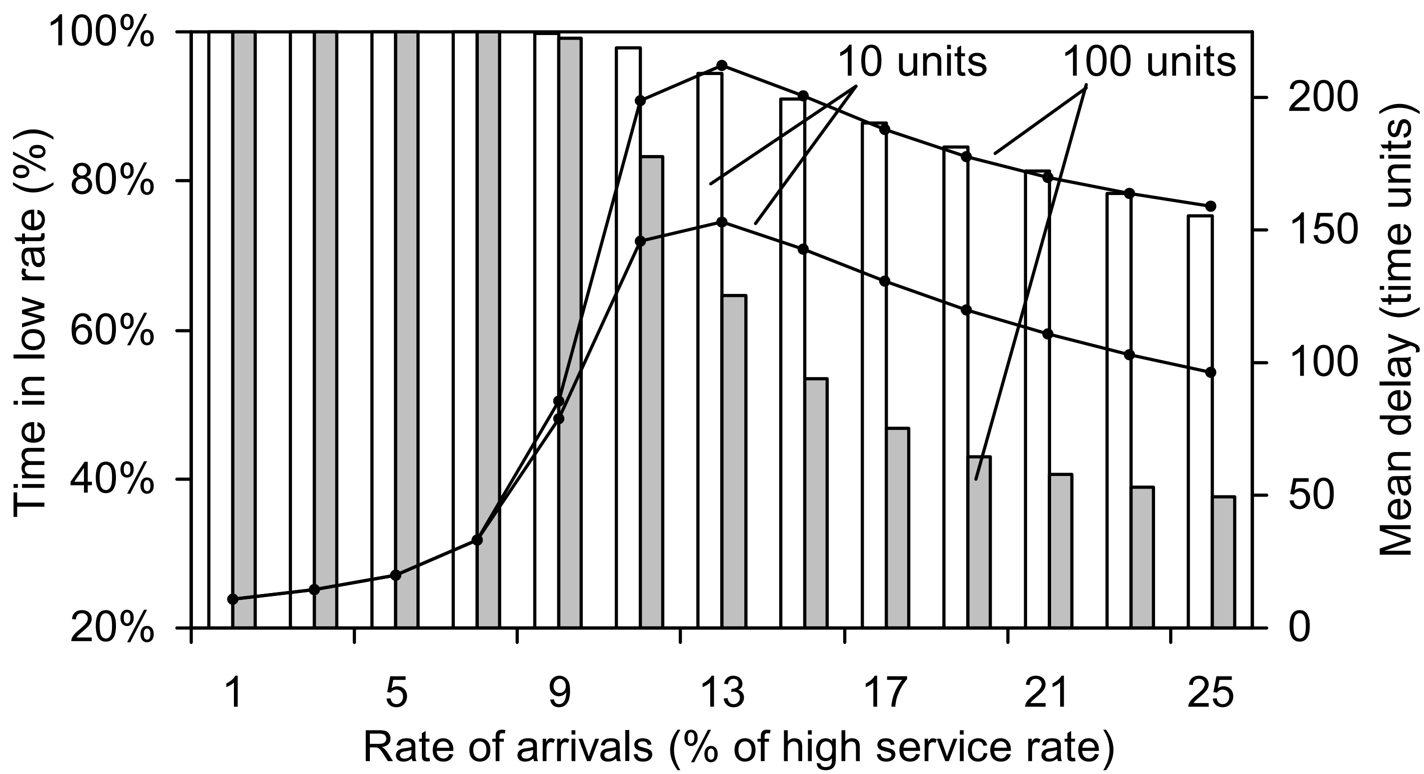


Simulation model

- **Evaluate ALR with a simulation model**
  - Built a queueing model using CSIM19
    - Validated against numerical results from Markov model
  - Can use Poisson and trace traffic inputs
  - Can model non-zero rate switching time

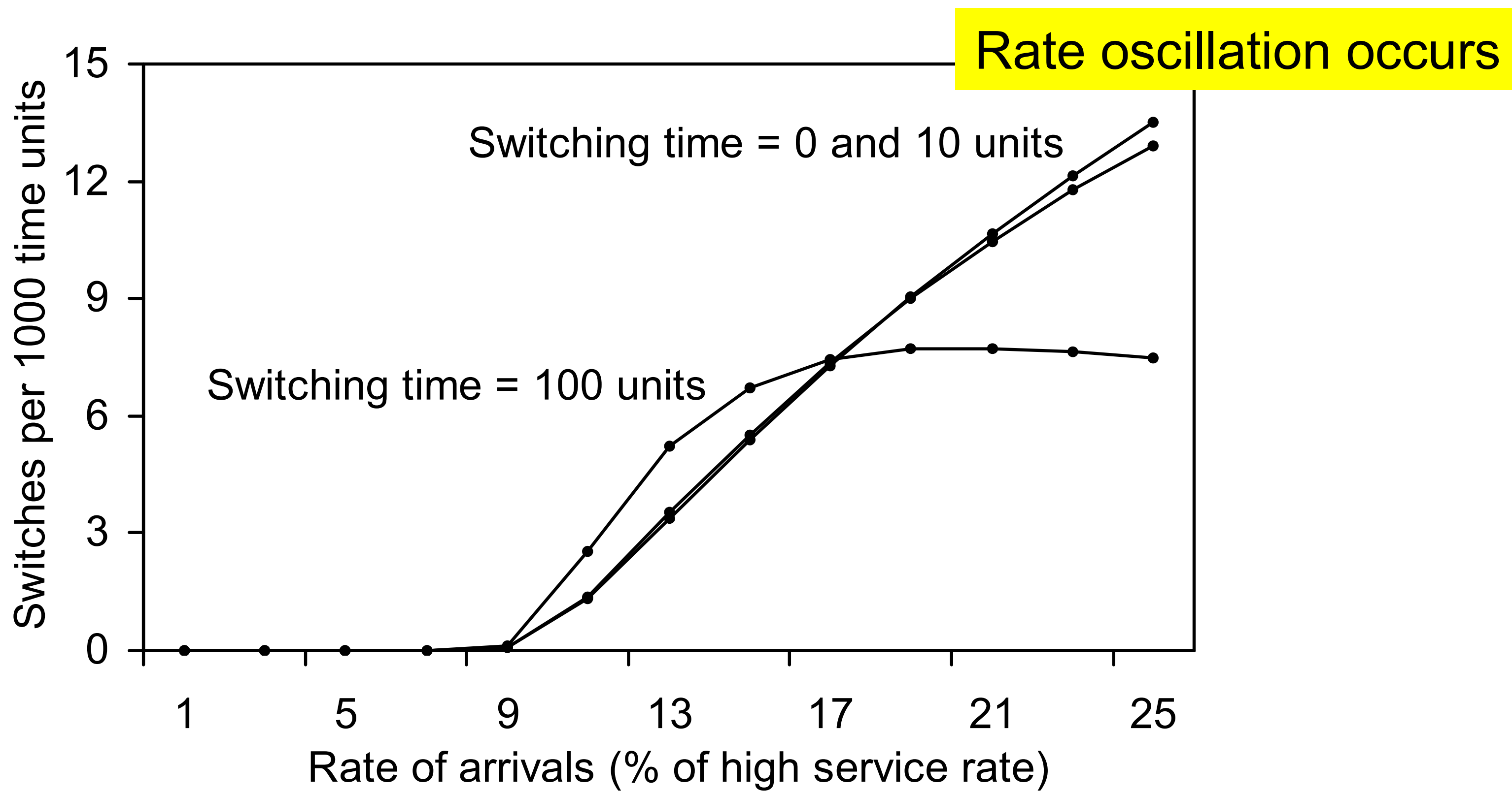
Results using Poisson traffic

- **Simulation results for non-zero switching time**
  - "time unit" = mean service time at high rate



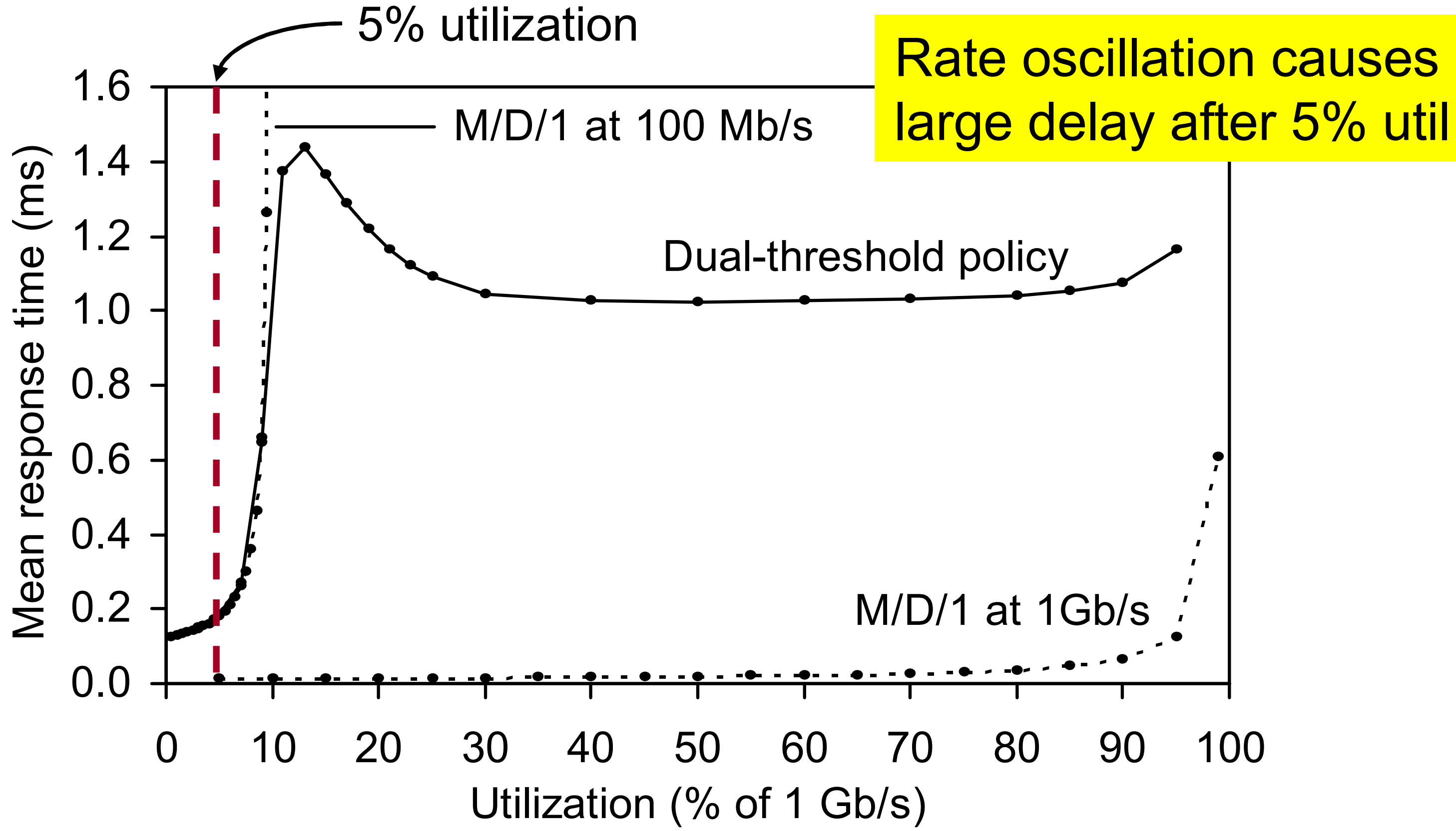
Results using Poisson traffic <u>continued</u>

- **Simulation results for non-zero switching time**



Results using Poisson traffic <u>continued</u>

- **More results for non-zero switching time**
  - Poisson arrivals, fixed length packets (M/D/1), 1 ms switch time



Results using trace traffic

- **Characteristics of trace traffic**
  - Note very low average utilization

Trace	Duration	Description	Avg util.
USF #1	0.5 hours	Link to "busiest" user in USF	4.11 %
USF #2	0.5	Link to 10th busiest user	2.63
USF #3	0.5	Link to an average user	0.03
PSU #1	2.0	Link to a desktop PC	0.13
PSU #2	2.0	Link connecting two switches	1.01
PSU #3	2.0	Link connecting switch to router	1.03

Results using trace traffic <u>continued</u>

- **Results for all traces without and with ALR**
  - Showing mean packet delay

Trace	10 Mb/s	100 Mb/s	ALR	
USF #1	7.60 ms	0.09 ms	2.79 ms	99.42 %
USF #2	3.95	0.08	1.81	99.81
USF #3	196.29	0.05	1.48	99.99
PSU #1	33.51	0.18	5.63	99.98
PSU #2	2321.31	0.12	9.55	99.12
PSU #3	1147.83	0.51	4.07	99.83

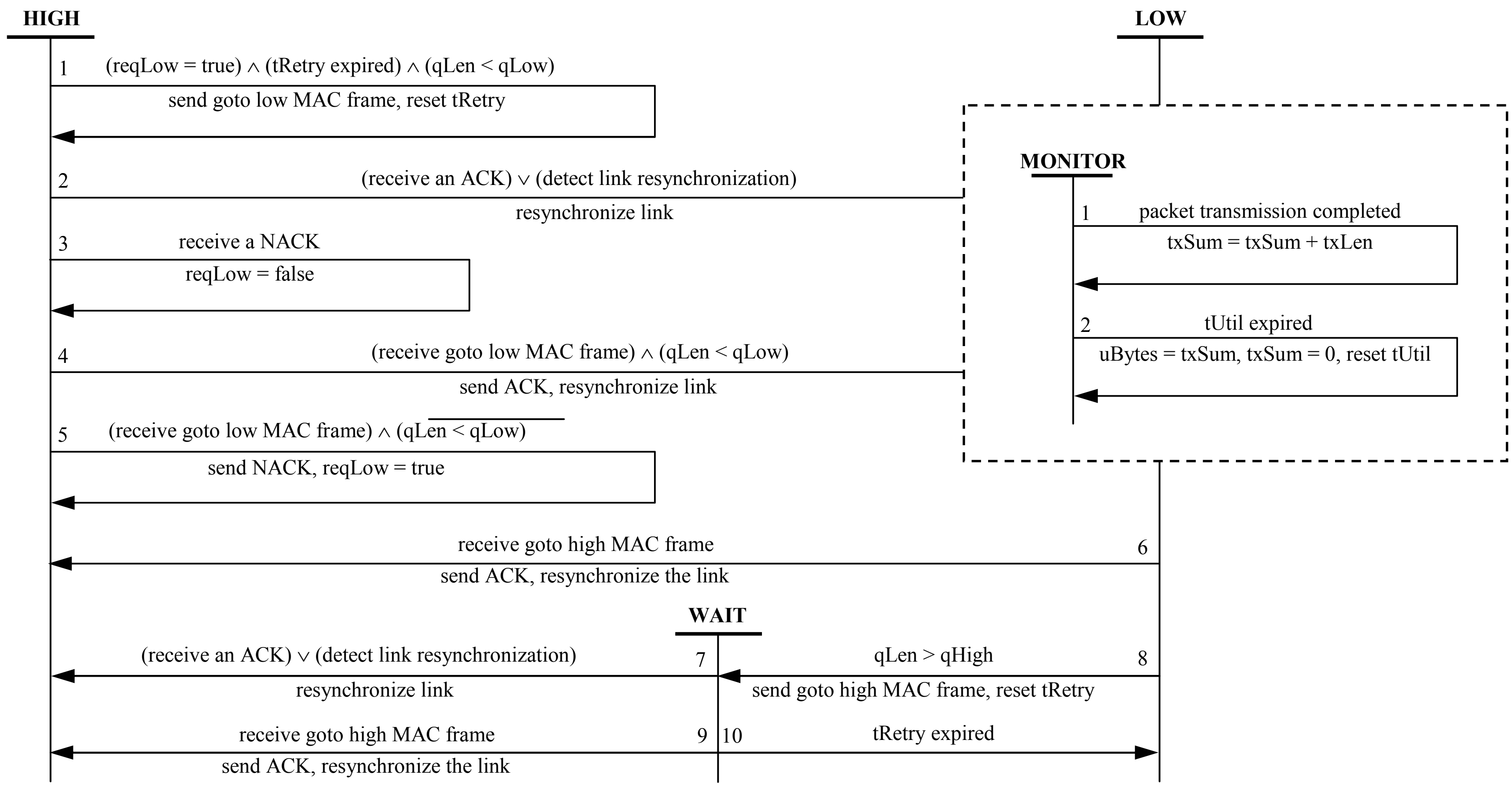
Time in 10 Mb/s

Improved ALR policy

- **ALR policy must prevent oscillation between rates**
  - Oscillation causes excessive packet delay
- **Utilization-threshold policy**
  - Use high threshold to increase link rate
  - Use explicit utilization measurement to reduce link rate

ALR utilization-threshold policy

- **Detailed FSM (Figures 6.1 and 6.2 in dissertation)**



Not in dissertation

ALR utilization-threshold policy continued

- **Simplified description**

Executes on receiving a frame

```
if (link rate is low)
  if (buffer size exceeds qHigh threshold)
    handshake for high link rate
```

Executes periodically at end of dampening time period (tUtil)

```
if (link rate is high)
  if (buffer size less than qLow threshold)
    if (bytes sent less than byte count threshold)
      handshake for low link rate
```

Comparison of ALR policies

- **Link data rate oscillations are compared**
- **Number of data rate oscillations can be calculated**
  - For dual-threshold policy – given by (5.27)

$$N_{dual} = \mu\pi_{k_1+1} + \sum_{n=1}^{\infty} \mu_1 \pi_{2k_2-k_1+2n}$$

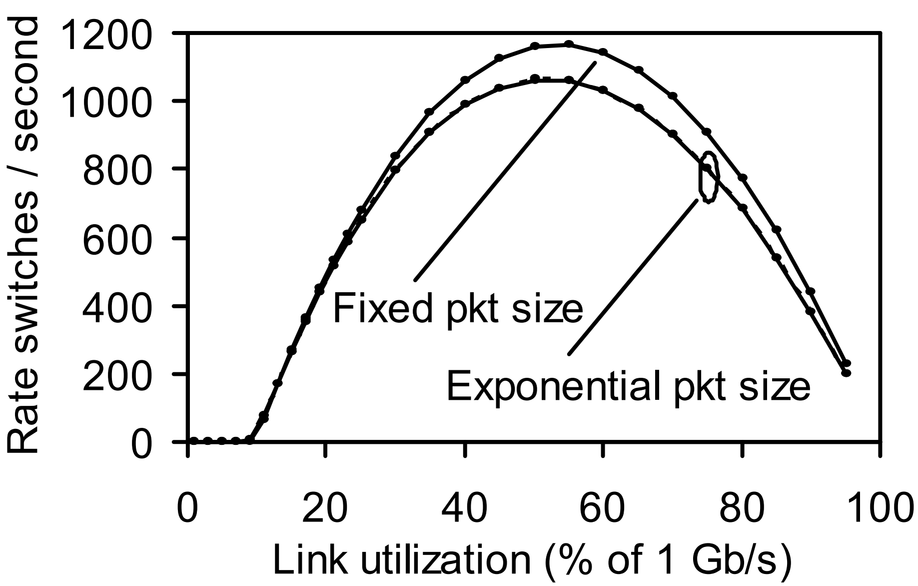
  - For utilization-threshold policy – given by (6.1) and (6.2)

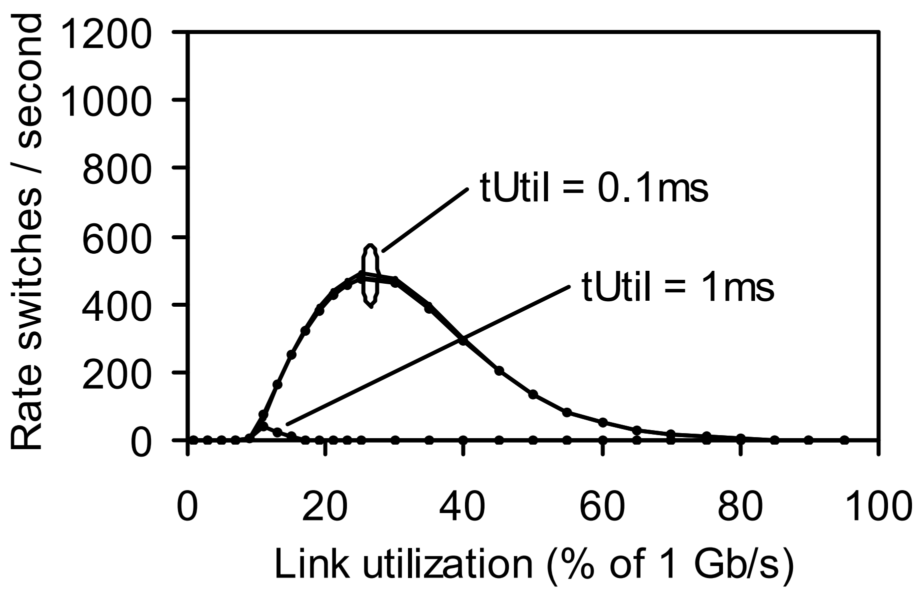
$$N_{oscilUtil} = 2\left(T_{passLow} + T_{passHigh} + \frac{tUtil}{q\sum_{n=0}^{k_1-1}\pi_n} \right)^{-1}$$

Comparison of ALR policies <u>continued</u>

- **Link date rate oscillations for both policies**
  - From calculations and simulations

- **Utilization-threshold policy has lower oscillations**
  - Poisson arrivals, service rates correspond to 1Gb/s and 100Mb/s





Evaluation of utilization-threshold policy

- **Control variables**
  - Threshold values
  - Utilization
  - Utilization sample time (tUtil timer)
  - Utilization byte count threshold value (uThresh)
- **Response variables**
  - Mean response time (mean packet delay)
  - Percentage of time in low data rate (energy saving)
- **Dual-threshold model was modified**
  - Developed realistic synthetic traffic generator

Traffic generator

New contribution

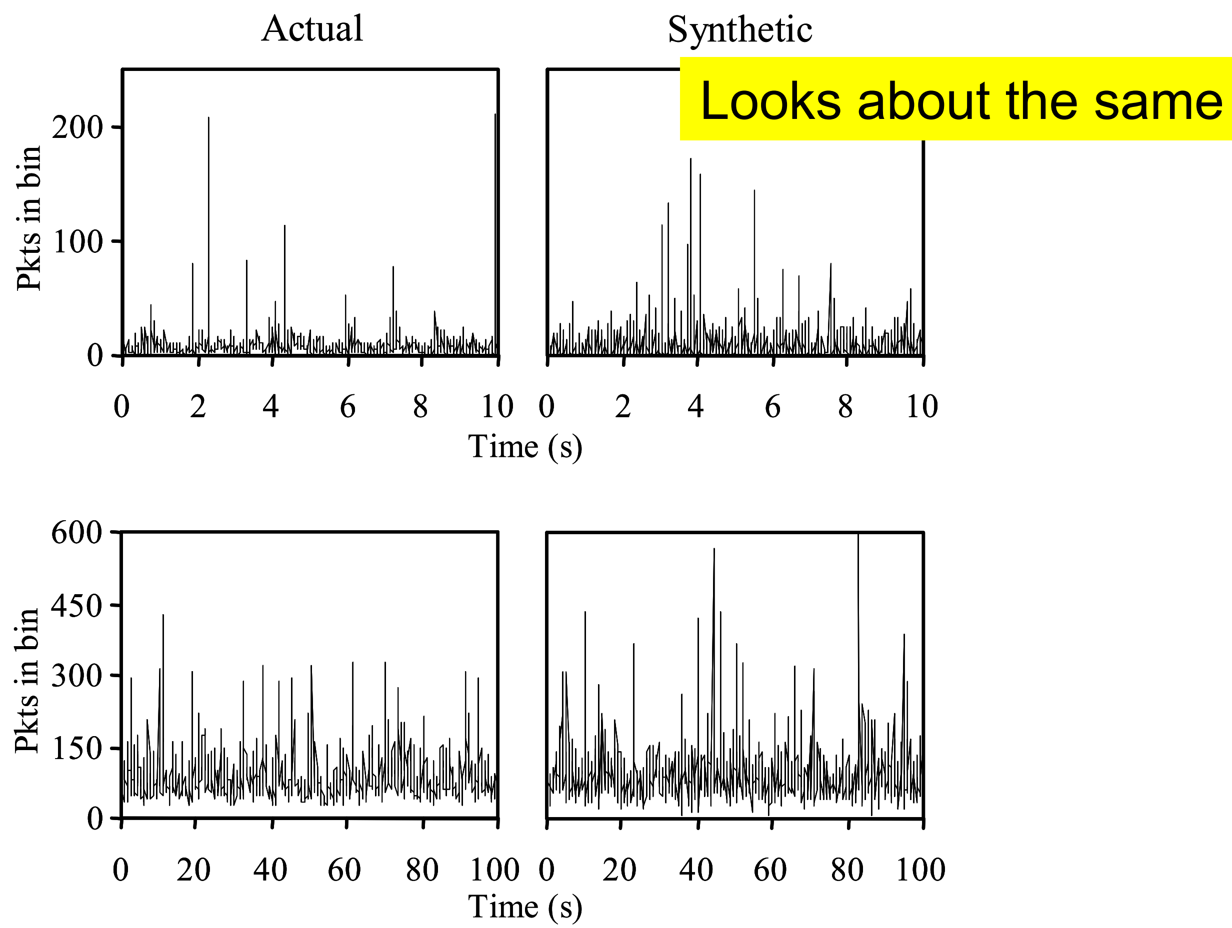
- **Developed a synthetic traffic generator**
  - Able to match most characteristics of traced traffic
    - Including utilization, burst length, CoV, and Hurst parameter

- **Generates synthetic bursty traffic**
  - Pareto burst size and exponential inter-burst time

- **Key parameters of traffic generator**
  - Data rate (100 Mb/s, 1 Gb/s, and 10 Gb/s)
  - Minimum and maximum burst size
  - Pareto index for bursts
  - Burst intensity
  - Packet length distribution
  - Mean utilization

Traffic comparison

- **Subjective visual comparison to traced traffic**



Traffic comparison <u>continued</u>

- **Objective measurement comparison to traced traffic**
  - Synthetic versus traced for 100 Mb/s

Close to the same

Characteristic	Actual	Synthetic
Mean inter-packet time (ms)	1.10	1.06
CoV of inter-packet times	1.76	3.81
Mean packet size (bytes)	577	526
CoV of packet size	1.16	1.15
Hurst parameter of packet counts	0.66	0.64
Utilization (% of 100 Mb/s)	4.11	3.95

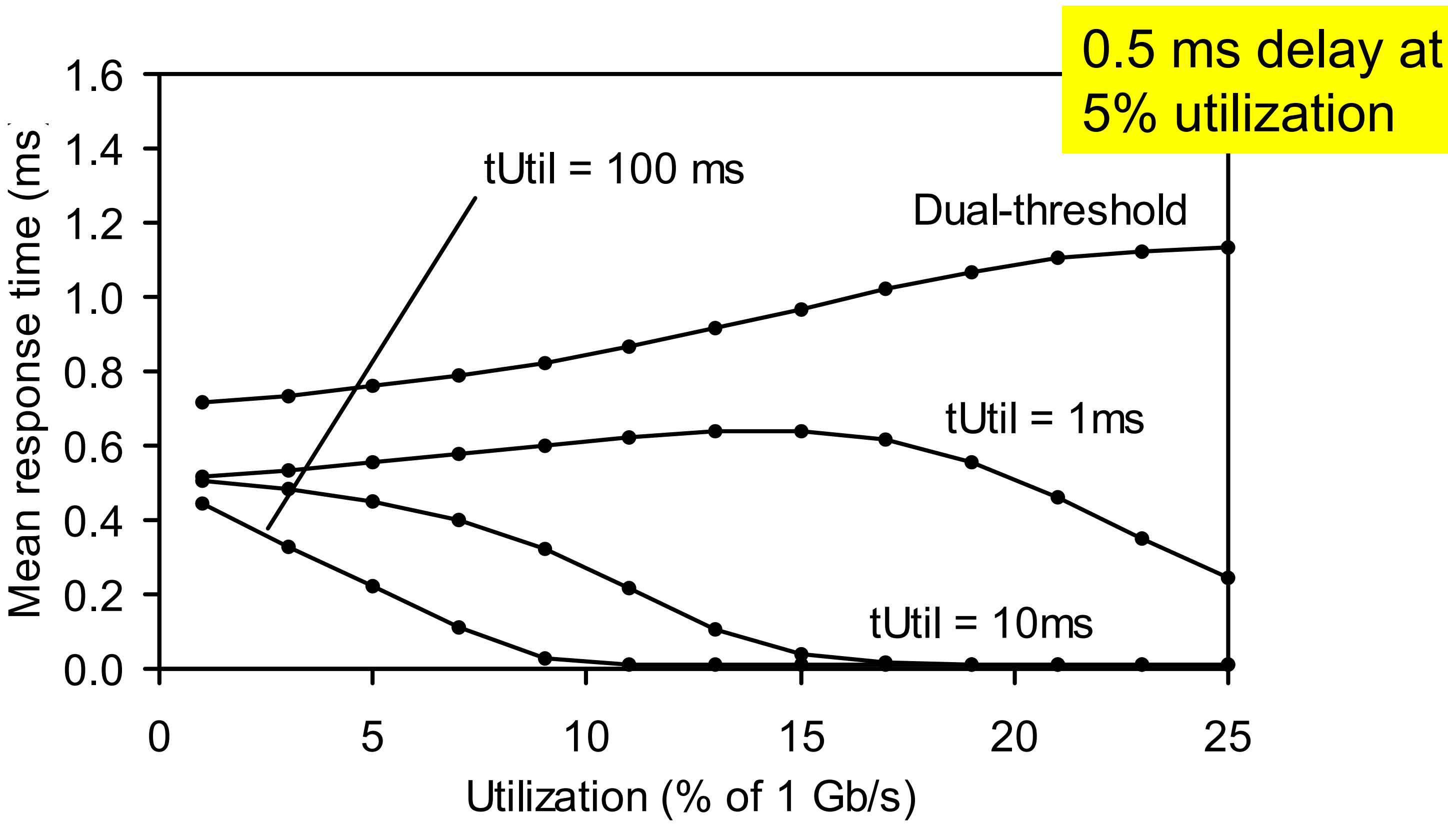
Simulation experiments

Rate switch time is 1 ms in all cases

- **Smooth traffic experiment**
  – Poisson arrivals, fixed length packets, utilization is varied
- **Single burst experiment 1**
  – One burst (Poisson) at 80% utilization and 0.4 s duration
- **Single burst experiment 2**
  – One burst (Poisson) at 80% utilization, length is varied
- **Bursty traffic experiment**
  – Bursty (mean burst = 8.4 KiB), utilization is varied
- **LAN switch experiment**
  – Use bursty traffic as input to a LAN switch model
  – Power consumption figures from Cisco Catalyst 2970
    - With no links = 46 W, each 100 Mb/s link is 0.3 W, 1 Gb/s 1.8 W
    - Measurements made at University of Florida

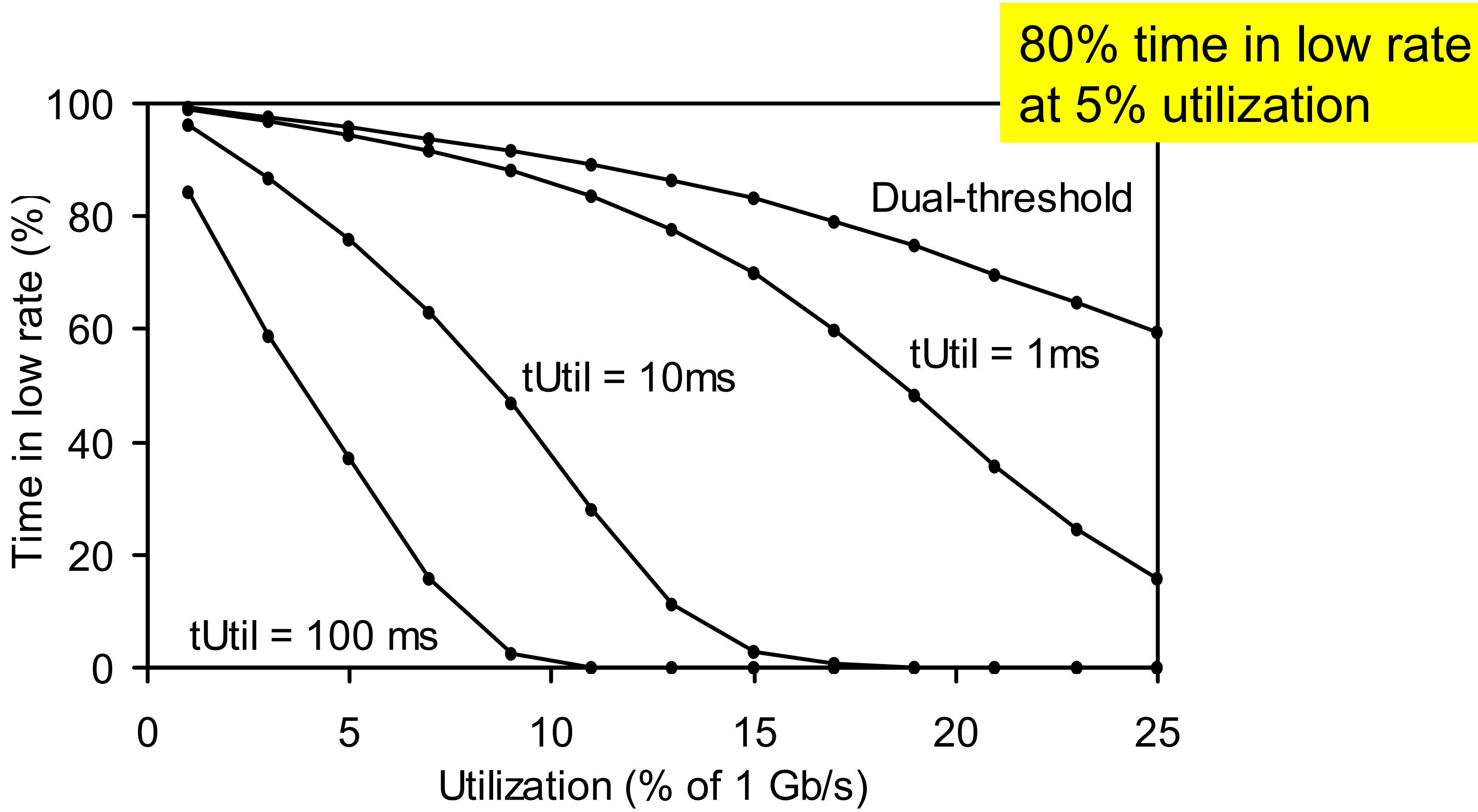
Bursty traffic experiment results

- **Mean response time (packet delay)**



Bursty traffic experiment results <u>continued</u>

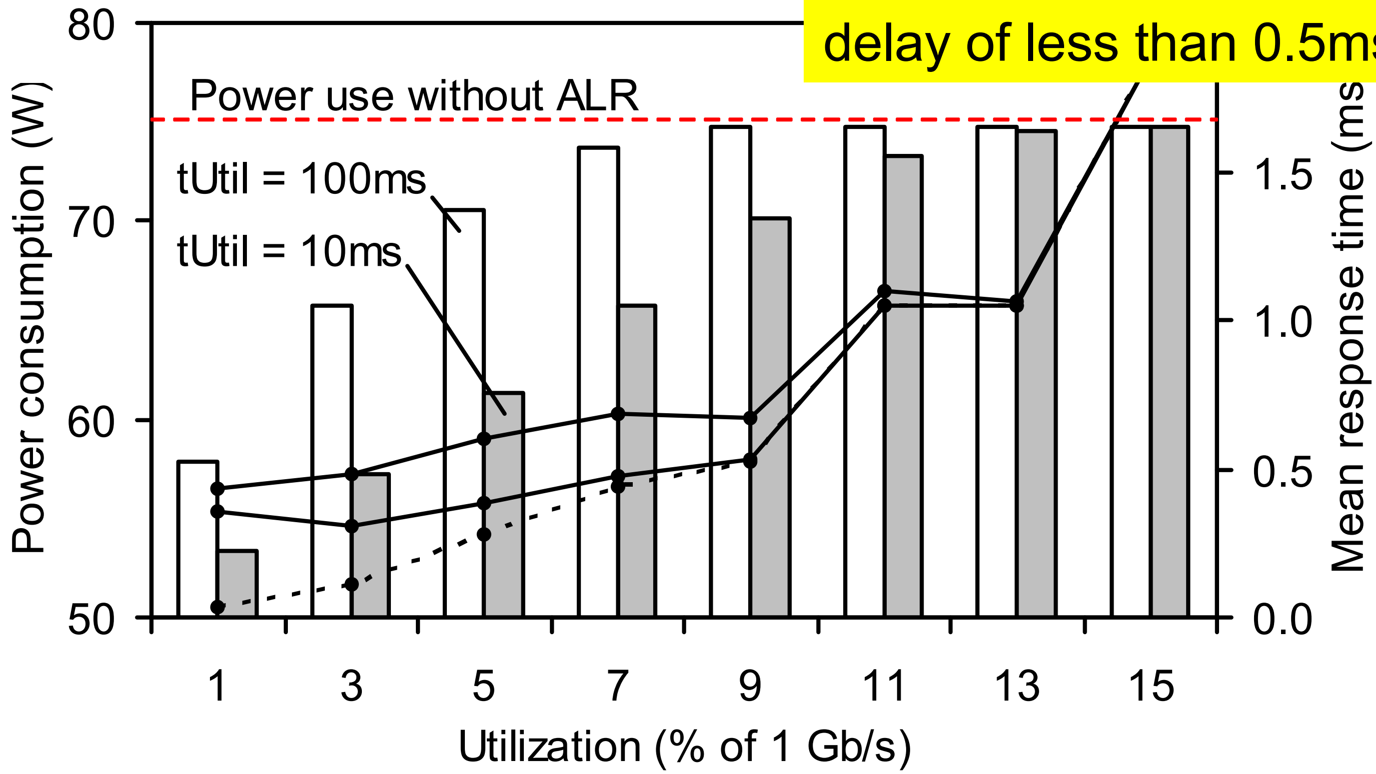
- **Time in low data rate (energy saving)**



LAN switch experiment results

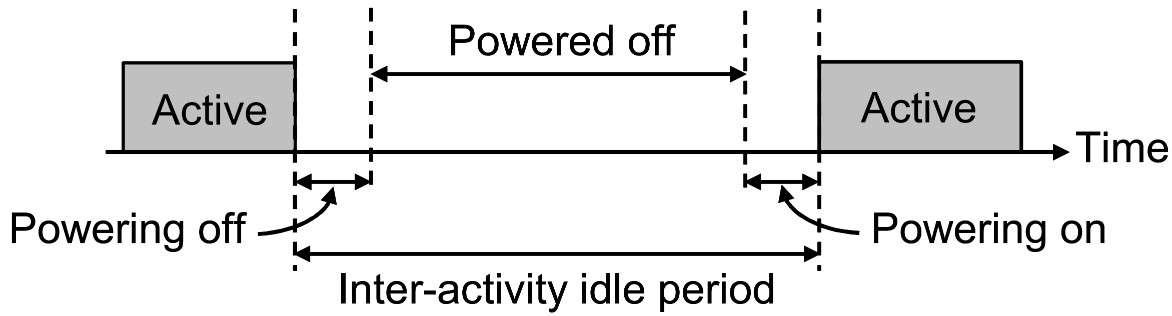
- **Power use and delay in one graph**
  - Shows trade-off

20% energy savings at 5% utilization with an added delay of less than 0.5ms



Powering off during idle periods

- **Energy can be saved by powering off when idle**
  – Key is detecting "long enough" idle periods



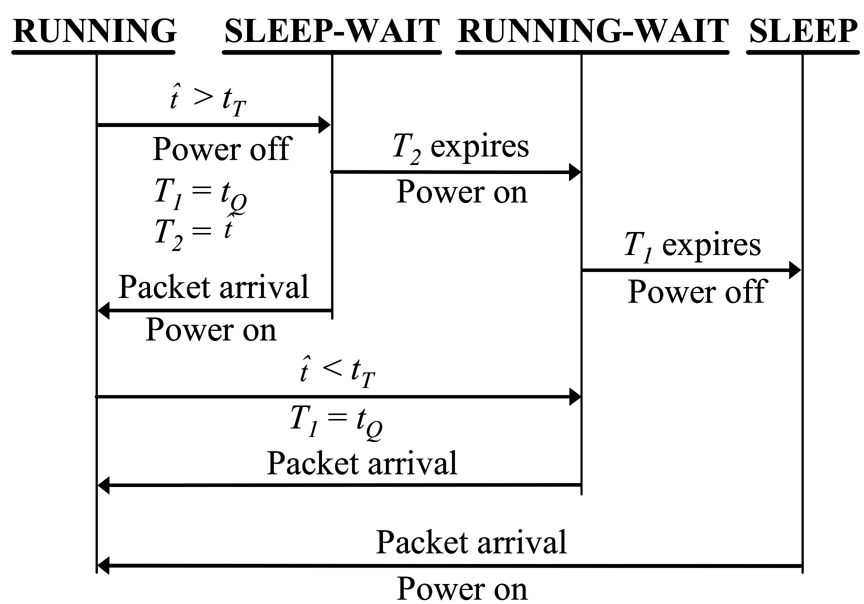
- **In networks, a few idle periods have most idle time**
  – In 3 of 4 traces, 10% of idle periods have over 50% of idle time

Powering off during idle periods

New contribution

- **Quantiles used to detect "long" idle periods**
  - If idle period extends beyond a threshold, then sleep
  - Quantile estimation is used to determine threshold t_Q
  - Powers on when request arrives

- **Performance evaluation**
  - Less powered off time than Hwang and Wu [59]
  - But, performance is better when forced wake-up is considered



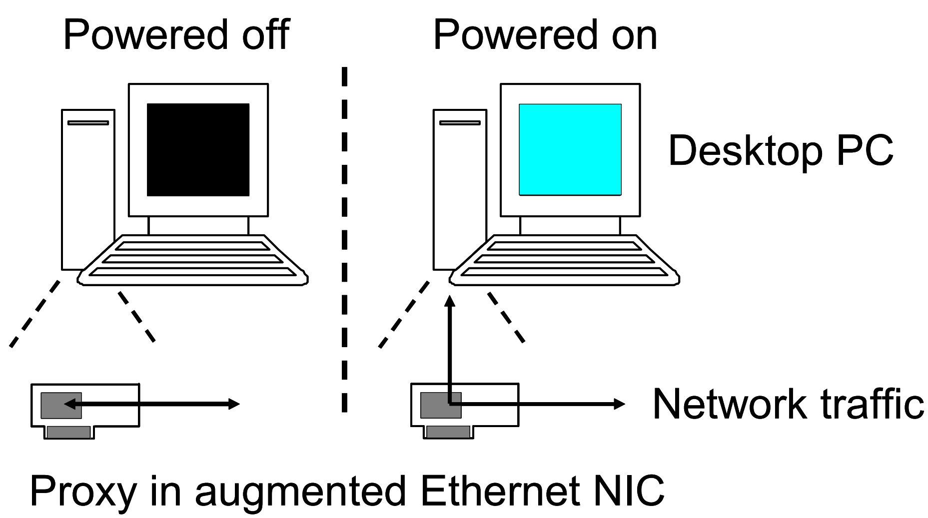
Agenda

- Introduction
- Reducing direct power consumption
- Reducing induced power consumption
- Contributions

A first look at protocol proxying

New contribution

- **Proxy capability in NIC to allow host to goto sleep**
  - Respond to ARP, ICMP messages when host is powered off
    - Maintain “network presence”
  - Wake up the PC when is services are needed



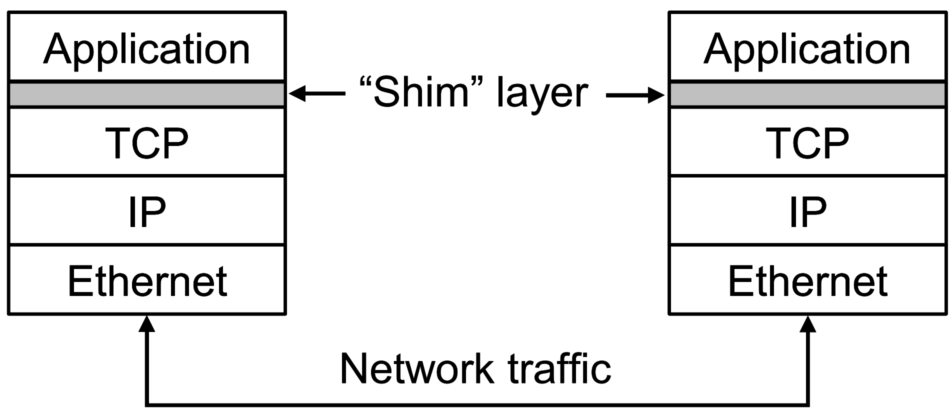
A first look at protocol proxying <u>continued</u>

- **Built a prototype proxy for a Web server**
  - Demonstrated feasibility of the idea

A first look at TCP connections

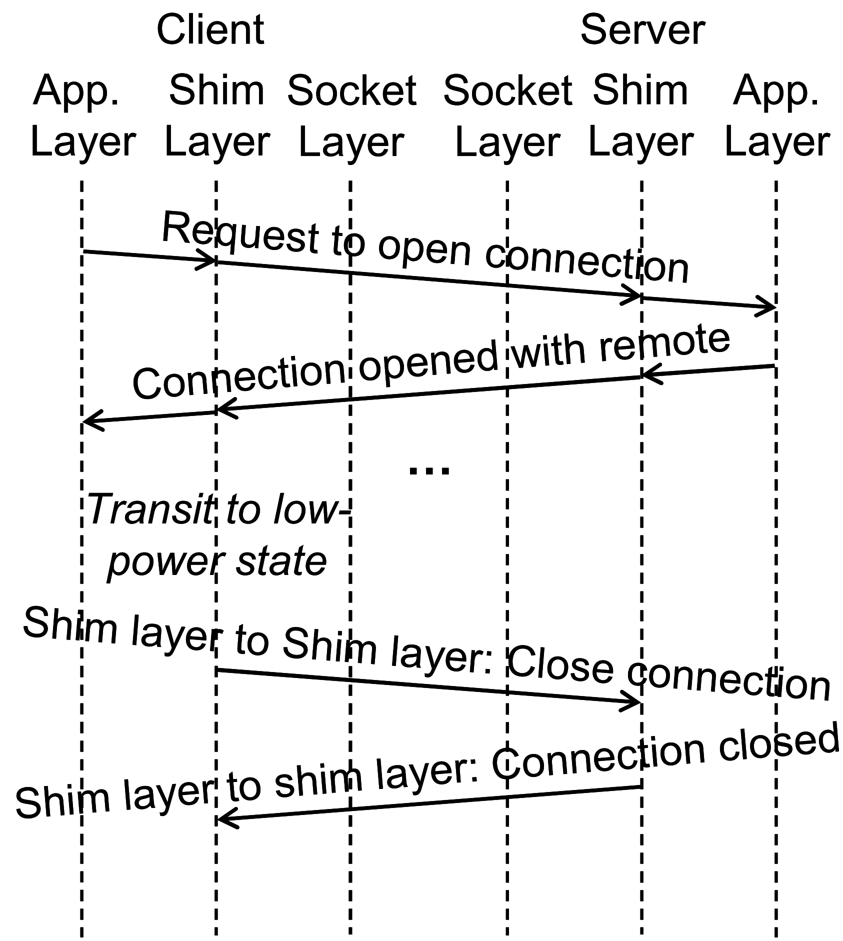
- **Long-lived TCP connections prevent power-off**
  - Need to respond to “keep-alive” messages
  - Lack of response leads to dropped connections

- **A prototype “shim” layer is developed**
  - Drops/resumes connections without application awareness
  - Permits the network host to power off



A first look at TCP connections continued

- **Built prototype "shim" using Telnet client and server**
  - Feasibility demonstrated



Agenda

- Introduction
- Reducing direct power consumption
- Reducing induced power consumption
- Contributions

Contributions

1. **The first method for reducing power consumption of Ethernet links**
   - ALR is expected to be an IEEE 802.3 standard!
2. **The first model and solution for a dual-threshold state-dependent service rate queue with rate change at service completion**
3. **A better synthetic Ethernet traffic generator**
4. **An improved idle period detection method using quantile estimation**
5. **Demonstrated feasibility of protocol proxying and splitting a TCP connection**

Publications from this work

1. **C. Gunaratne**, K. Christensen, S. Suen, and B. Nordman, "Reducing the Power Consumption of Ethernet with Adaptive Link Rate (ALR)," *IEEE Transactions on Computers*, to be submitted in December 2006.

2. **C. Gunaratne**, K. Christensen, and S. Suen, "Ethernet Adaptive Link Rate (ALR): Analysis of a Buffer Threshold Policy," *Proceedings of IEEE GLOBECOM 2006*, November 2006. (Acceptance rate = 40%)

3. **C. Gunaratne** and K. Christensen, "Ethernet Adaptive Link Rate: System Design and Performance Evaluation," *Proceedings of the 31st IEEE Conference on Local Computer Networks*, pp. 28-35, November 2006. (Acceptance rate = 35%)

4. **C. Gunaratne**, K. Christensen, and B. Nordman, "Managing Energy Consumption Costs in Desktop PCs and LAN Switches with Proxying, Split TCP Connections, and Scaling of Link Speed," *International Journal of Network Management*, Vol. 15, No. 5, pp. 297-310, September/October 2005. (Acceptance rate = 29%)

5. **C. Gunaratne** and K. Christensen, "A New Predictive Power Management Method for Network Devices," *IEE Electronics Letters*, Vol. 41, No. 13, pp. 775-777, June 2005. (Acceptance rate = low)

6. K. Christensen, **C. Gunaratne**, B. Nordman, and A. George, "The Next Frontier for Communications Networks: Power Management," *Computer Communications*, Vol. 27, No. 18, pp. 1758-1770, December 2004. (Acceptance rate = 26%)