A Study between Networks and General Purpose Systems for High Bandwidth Video Streaming

John Bresnahan, Ioan Raicu & Gohar Margaryan

June 3rd, 2004

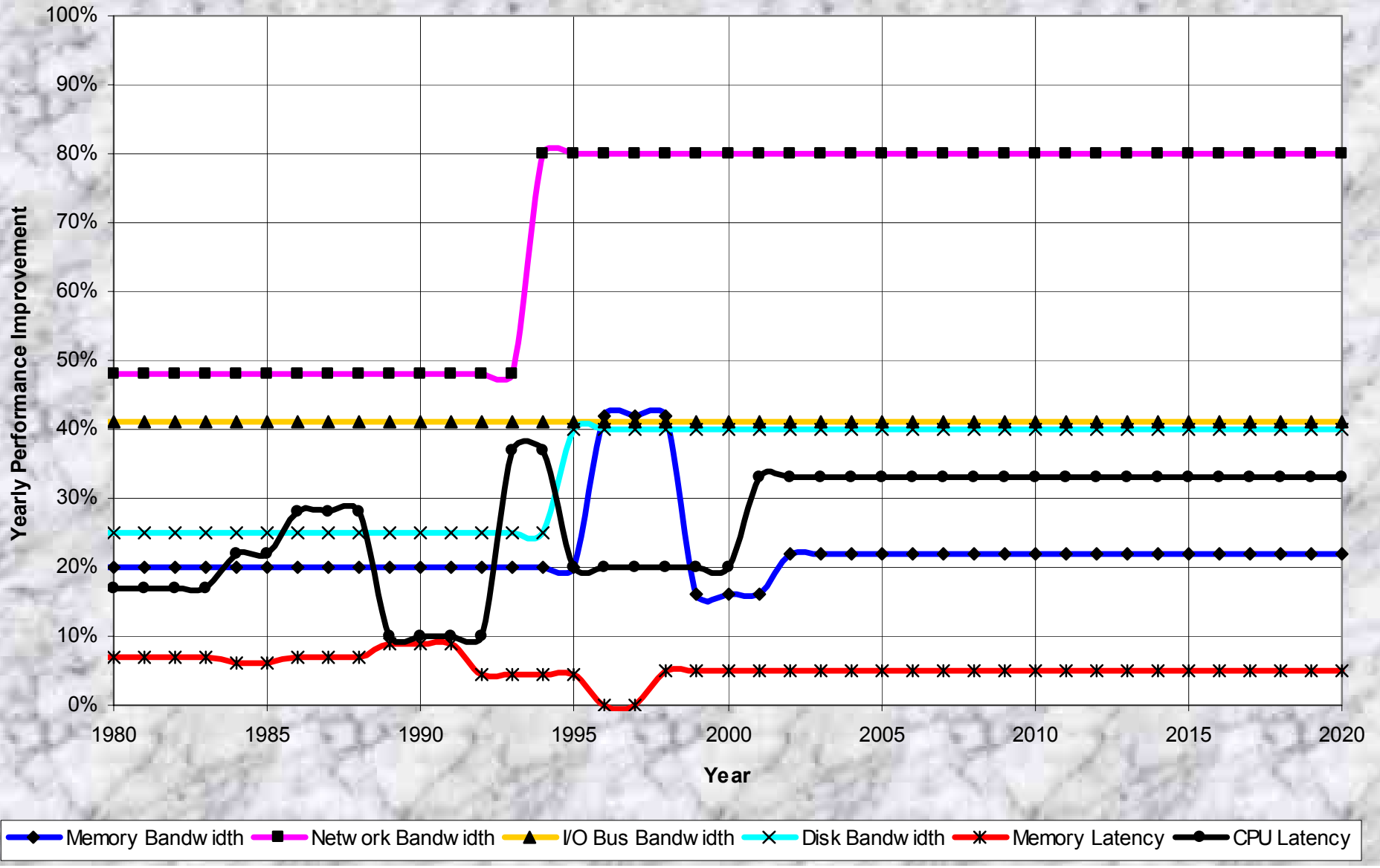
Computer Architecture – Spring Quarter 2004

Computer Science Department
University of Chicago

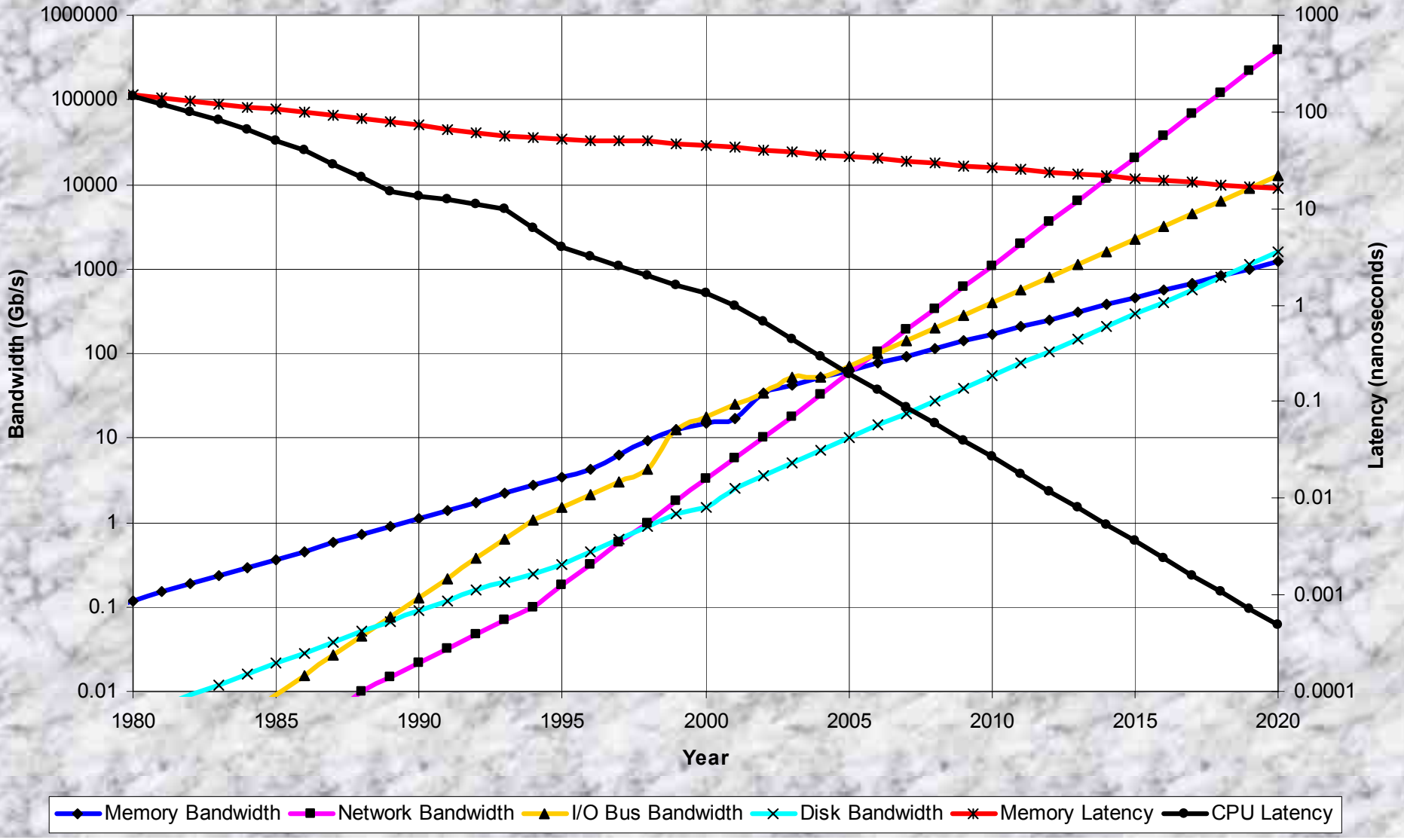
Problem Description & Motivation

- Study interaction between network, memory and CPU
  - Hardware is improving at different rates
  - Flow of bytes between components
  - CPUs involvement in rate of flow
- Predict appropriate hardware for a given high bandwidth workload
- Identify bottlenecks
- Visualize applications in pseudo-realtime

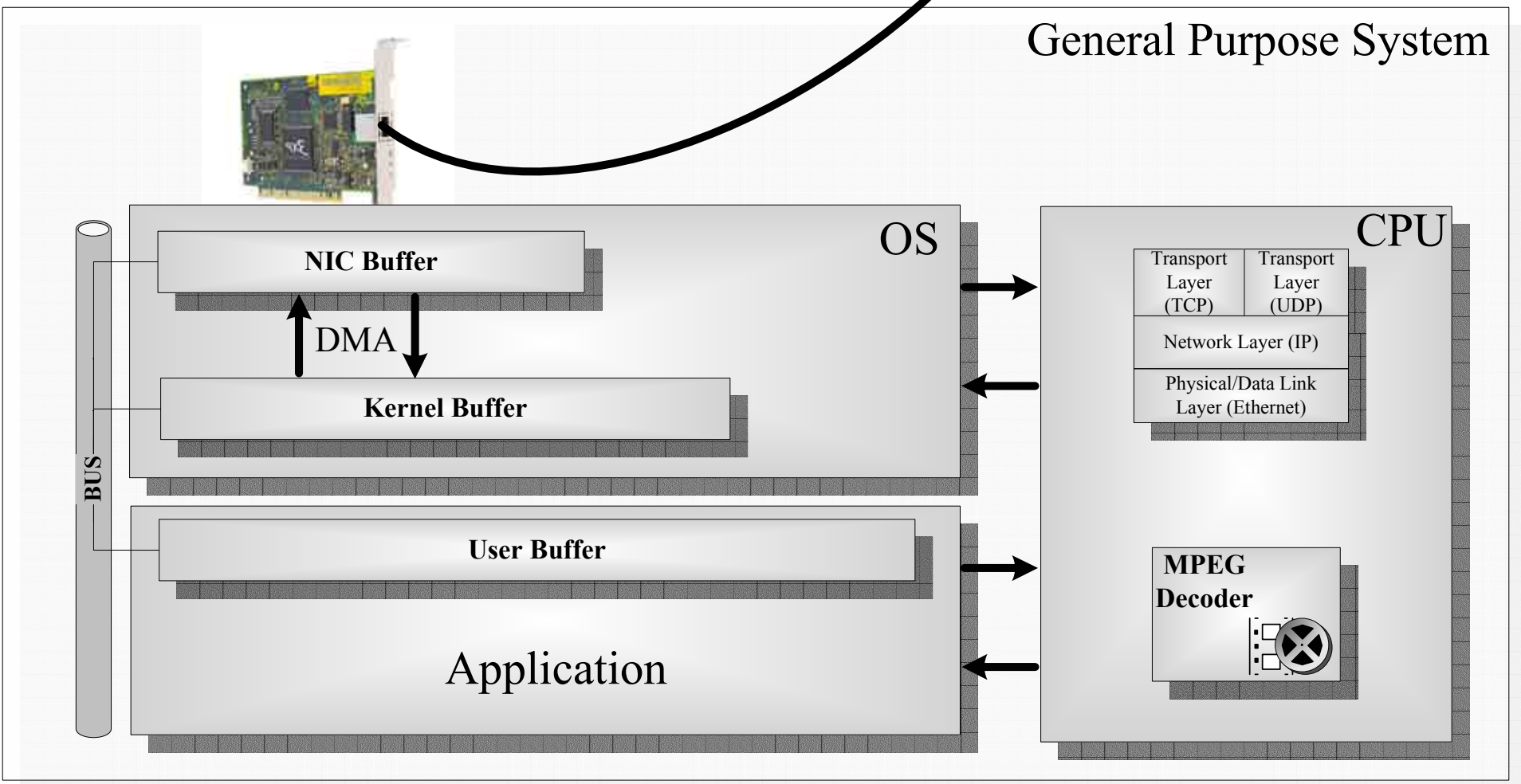
Yearly Performance Improvement



Historical Trends

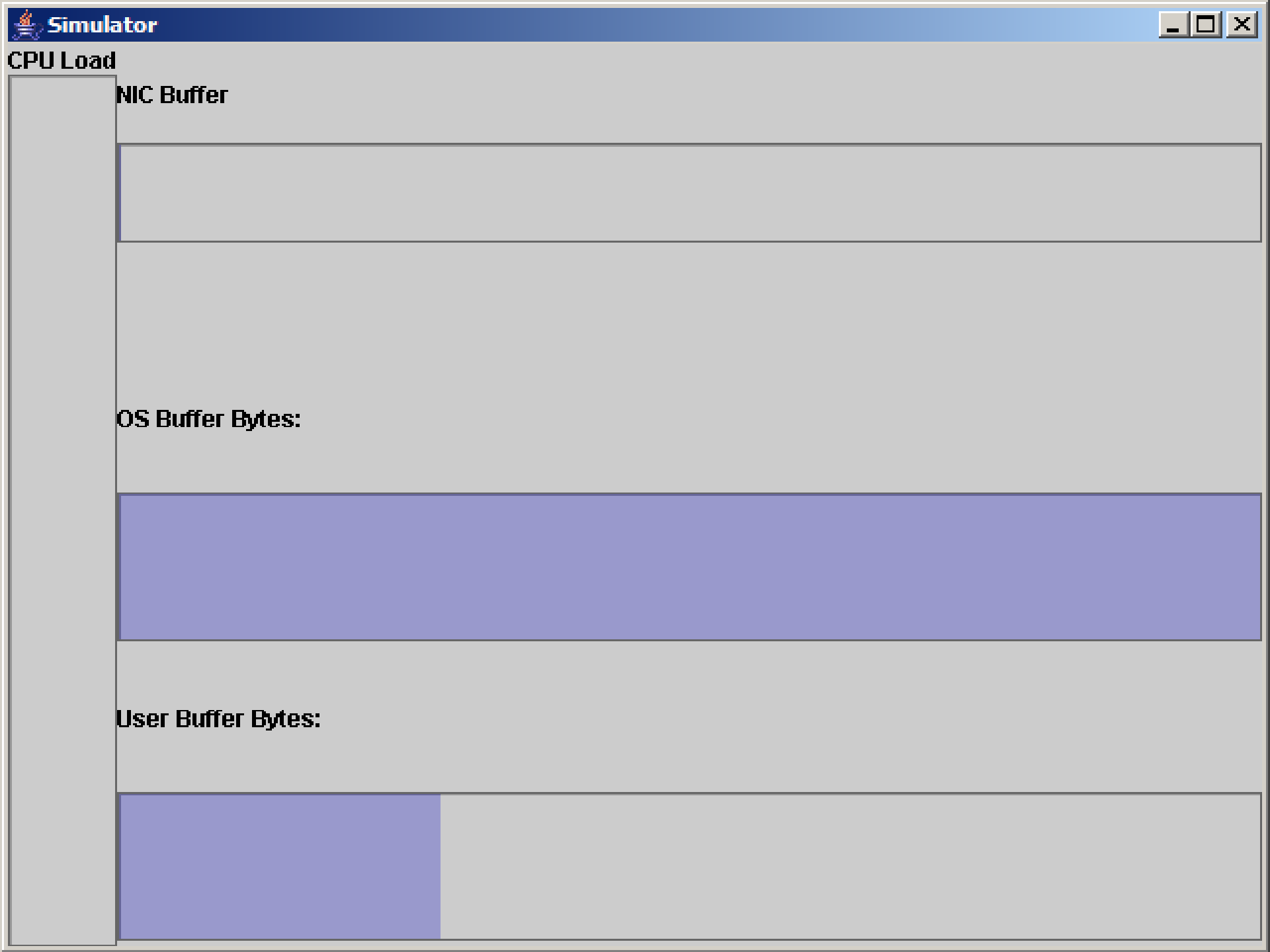


Network Application Model



Our Approach

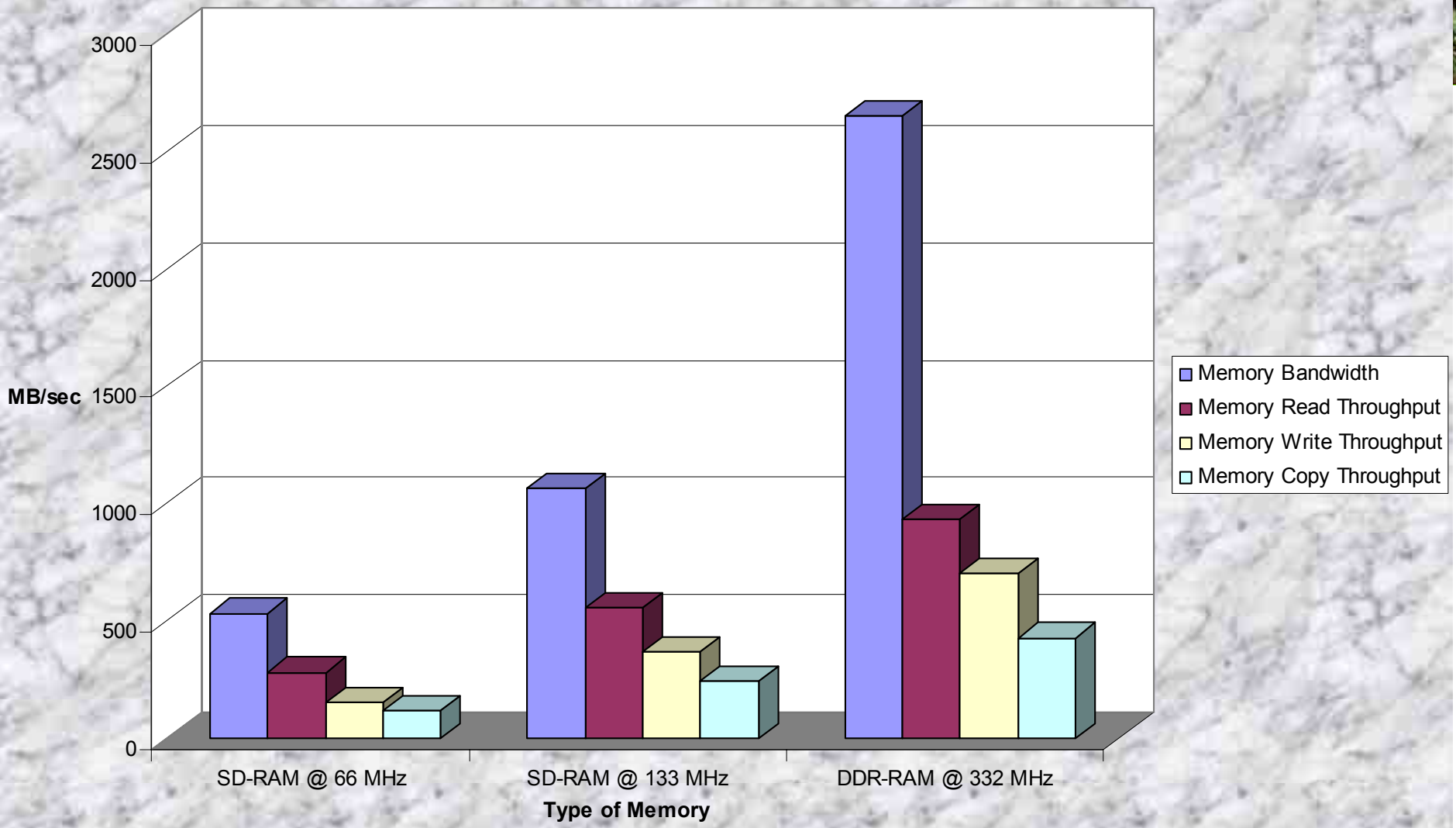
- Create a discrete event simulator
  - Model network app components
  - Flow of data between components
  - Configurable parameters…
- Empirical study to collect component performance
- Profiling jobs
  - Visualize use of components
- Achieved throughput, dropped packets


Simulator
CPU Load
NIC Buffer
OS Buffer Bytes:
User Buffer Bytes:

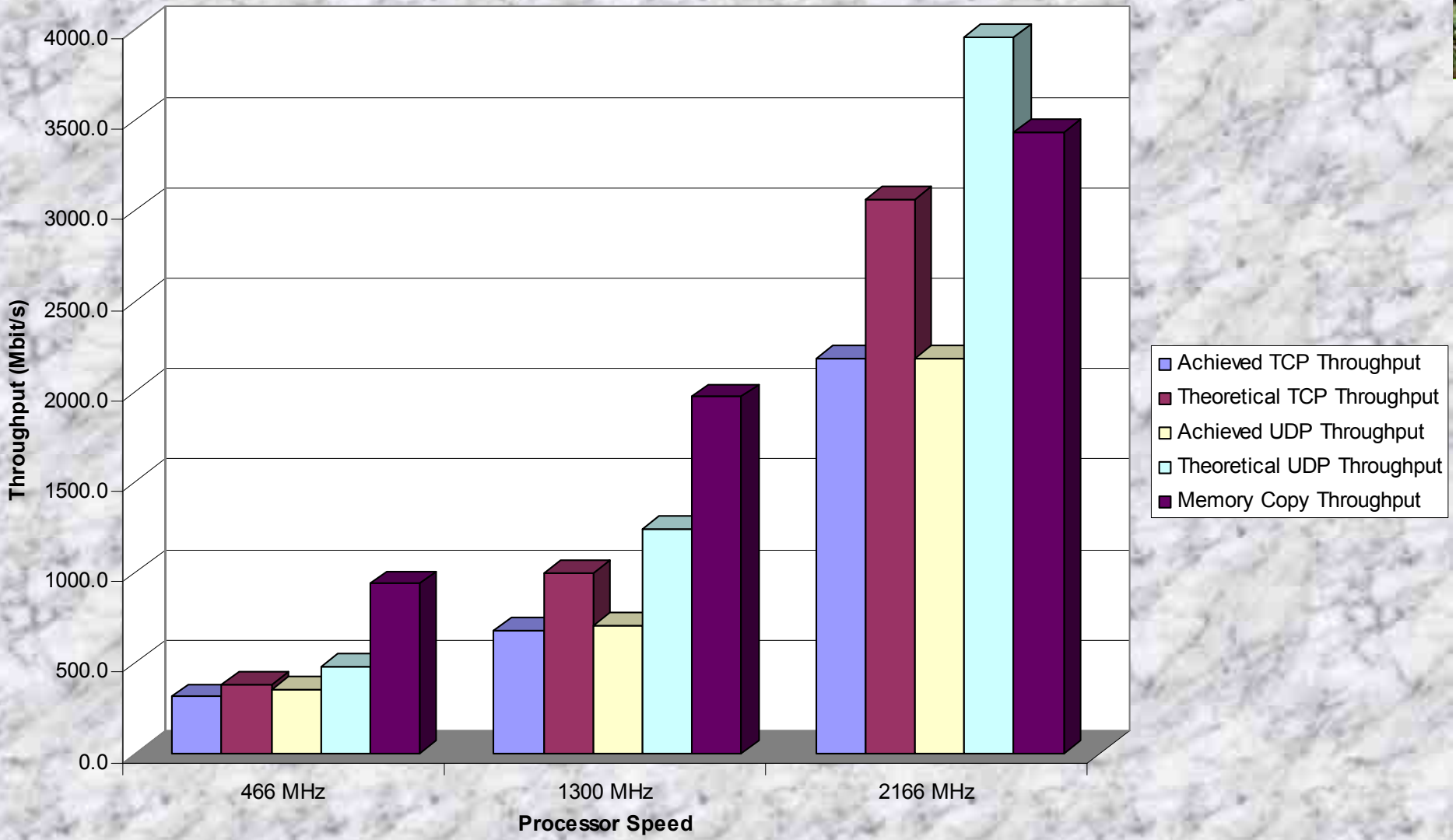
Component Modeling

- NIC
  - Tests over 10/100/1000 Mb/s running TCP and UDP
- Memory
  - Cache Burst 32
  - Measures L1 cache, L2 cache, and main memory read, write, and copy throughput & latency
- CPU
  - Network processing
    - packet/second ➔ CPU Cycles per byte of header processing
    - 2 copies: NIC buffer ➔ Kernel buffer ➔ User buffer
    - Iperf over local loopback address
  - MPEG
    - CPU Cycles per byte of processing

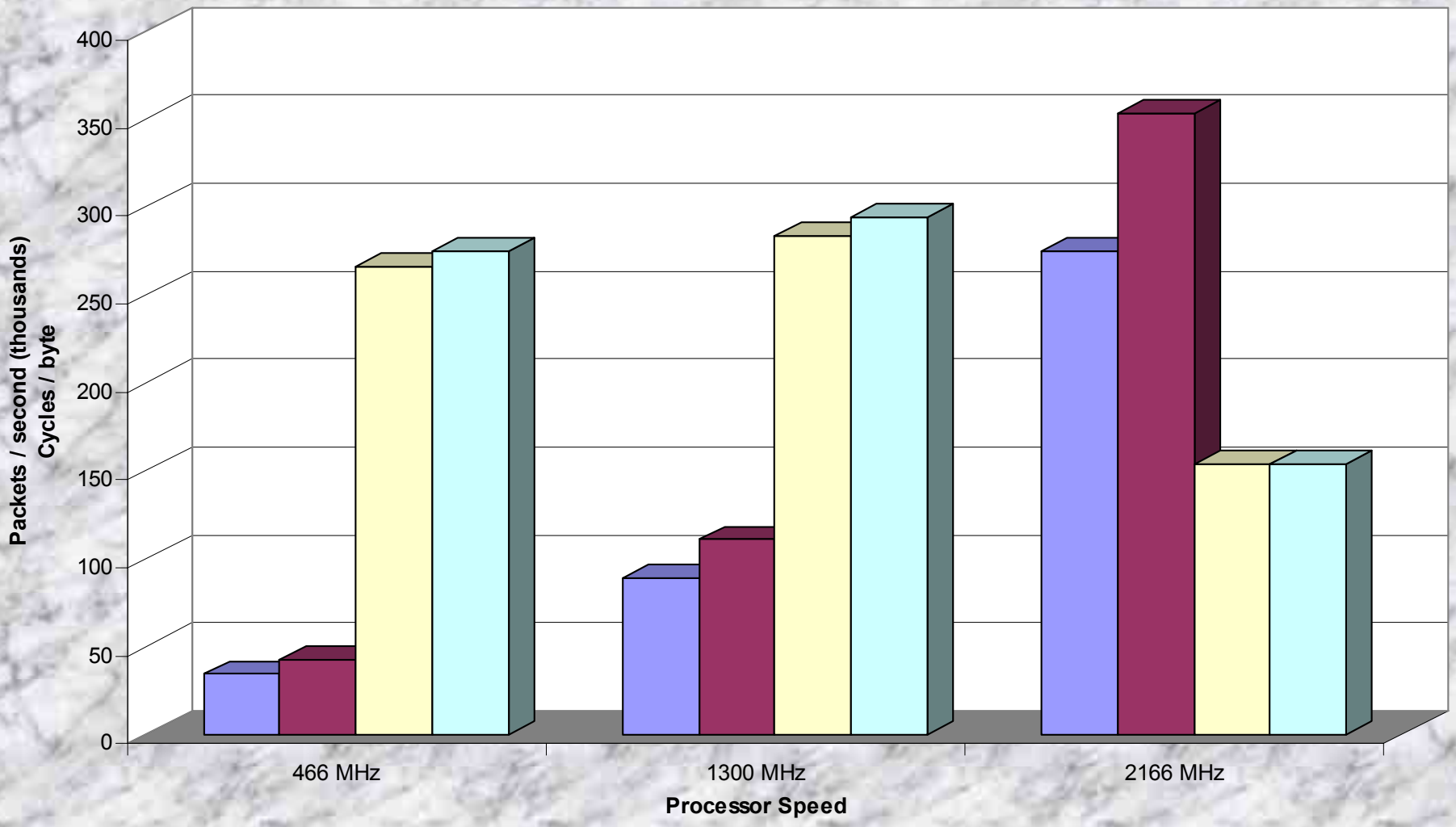
Main Memory Performance



TCP, UDP, and Memory Copy Performance



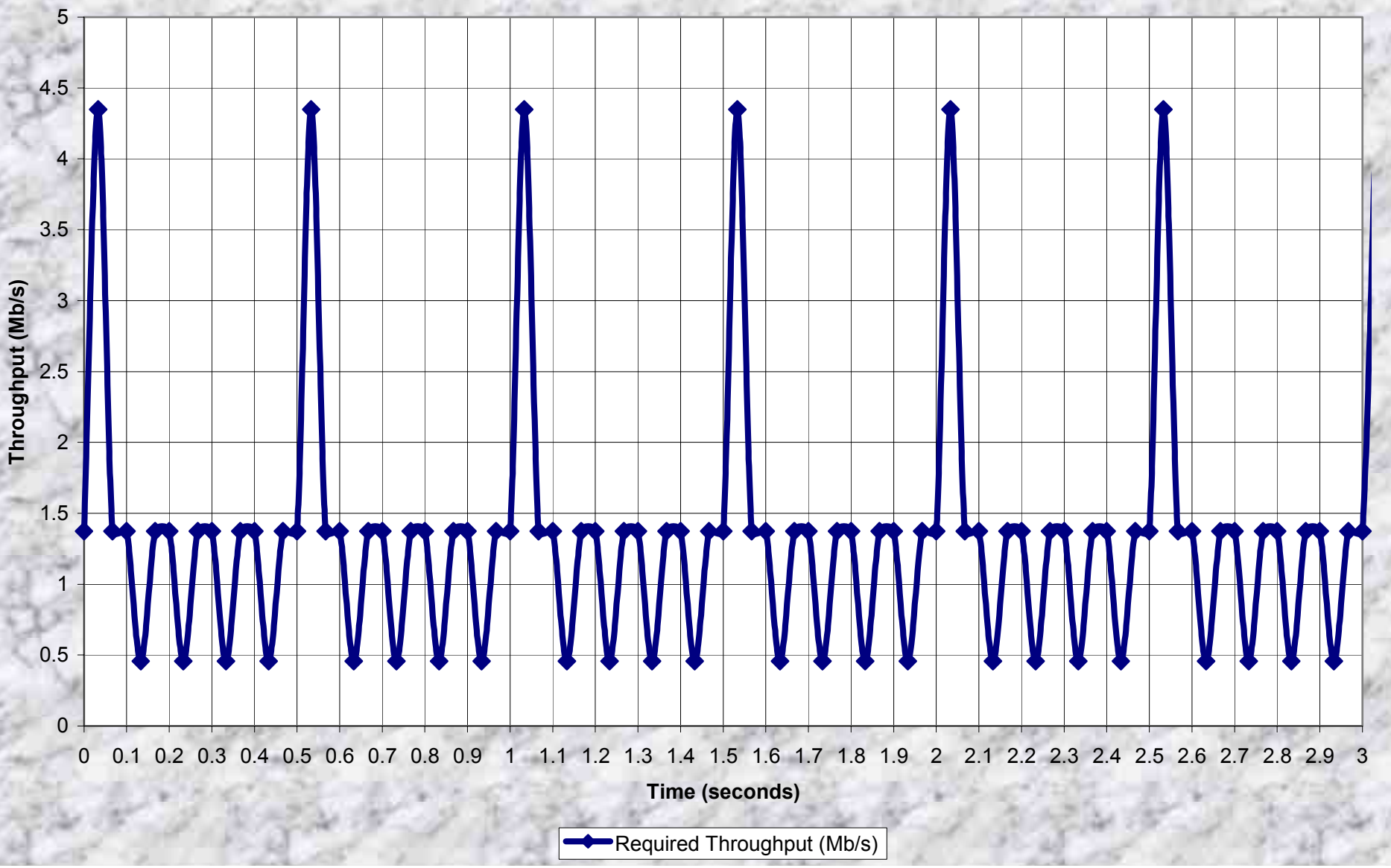
TCP and UDP Performance vs. Processing Power



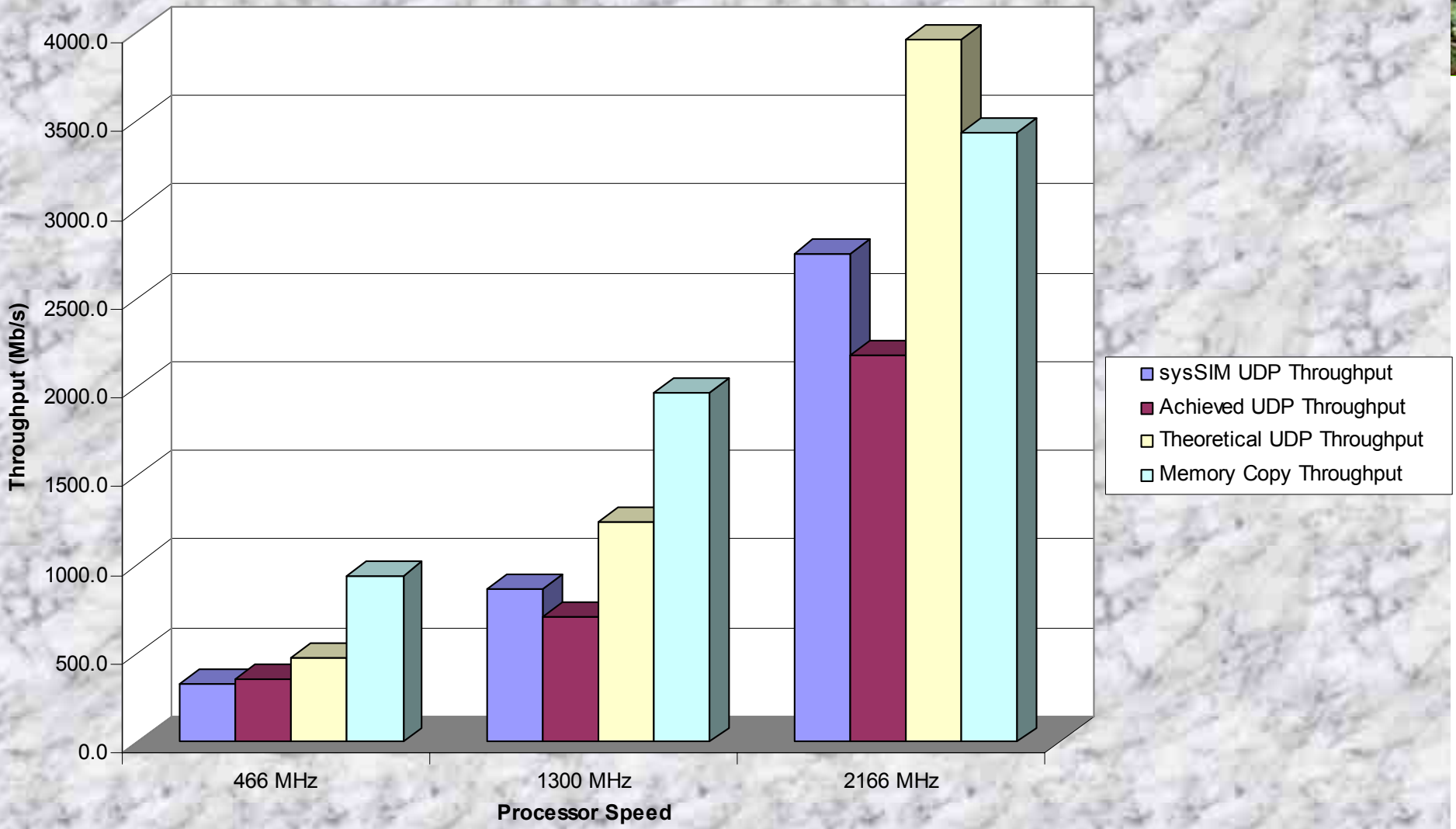
Benchmarks

- ***MPEG_sw***
  - software decoding (intensive CPU) and variable network traffic
- ***MPEG_hw***
  - hardware decoding and variable network traffic
- ***RAW***
  - Constant network traffic

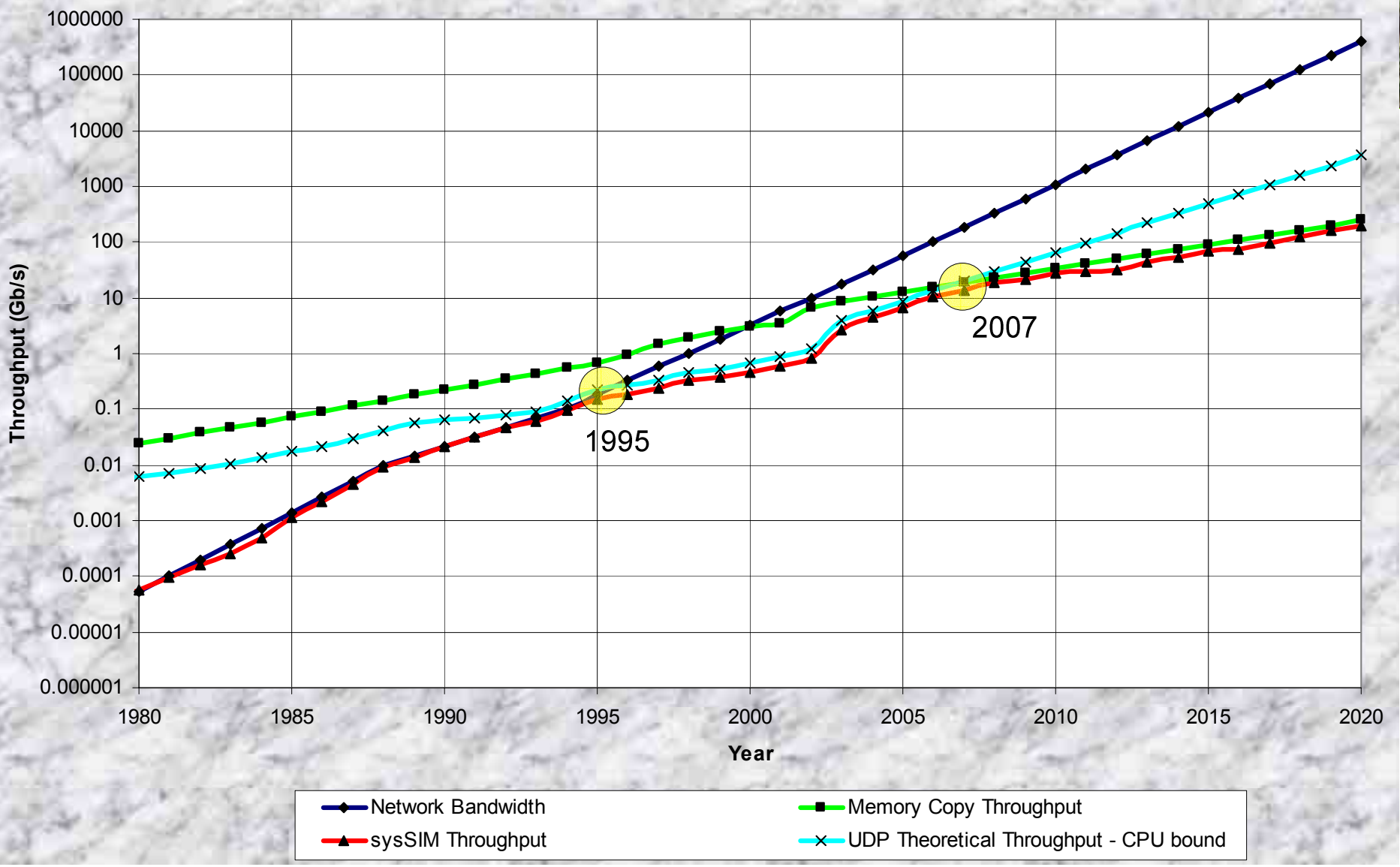
Video 1 Required Variable Throughput (Mb/s)



sysSIM Validation



Bottleneck Shifting in Time



Assumptions and Weaknesses

- LAN environment
  - NO out of order arrival of packets
  - NO “lost” packets
  - NO erroneous packets
- Unidirectional traffic
  - OK for modeling UDP, but oversimplification for TCP
- TCP/UDP/IP: 2 copies of data in protocol stack
- Future trends will follow past trends
- Empirical studies sampled only 3 machines
- Many details about network protocol stack and OS left untouched

Related work

- Simulators
  - SimOS: complete machine simulation environment that runs commercial OS
  - M5: simulation system targeting network intensive workloads that runs unmodified commercial OS
  - CSIM: discrete event simulator for describing parallel processor architectures and software mappings
- Visualizations
  - Visualization Tool (VT)
  - FlowScan: A Network Traffic Flow Reporting and Visualization Tool
- Empirical Studies
  - The Architectural Costs of Streaming I/O: A Comparison of Workstations, Clusters, and SMPs
  - Server Network I/O Acceleration: Fundamental to the Data Center of the Future
  - Imbench: Portable Tools for Performance Analysis

Conclusions

- Memory is not a bottleneck yet, but the gap is closing
- CPU is the bottleneck, but at the rate of increase in CPU speeds, it will not be a bottleneck for long
- At the current rate of network speed increases, we don’t foresee the network to be a bottleneck

Solutions and Open Problems

- Multiple memory banks
- TCP offloading / Network processors
- Hardware threads
- Multiple processors (SMP)
- Use high speed cache memory for buffers
- 0-copy scheme

References

[1] S. McCanne and S. Floyd. “NS-2 Network Simulator”. http://www.isi.edu/nsnam/ns/.

[2] MENDEL ROSENBLUM, EDOUARD BUGNION, SCOTT DEVINE, and STEPHEN A. HERROD. “Using the SimOS Machine Simulator to Study Complex Computer Systems.” ACM Transactions on Modeling and Computer Simulation, Vol. 7, No. 1, January 1997, Pages 78–103.

[3] Carl Hein. “CSIM - Parallel Process and Diagrams Simulator”, Lockheed-Martin ATL, 2004, http://www.atl.lmco.com/proj/csim/simulator/csim_doc.html.

[4] Nathan L. Binkert, Erik G. Hallnor, and Steven K. Reinhardt. “Network-Oriented Full-System Simulation using M5”. Sixth Workshop on Computer Architecture Evaluation using Commercial Workloads (CAECW), Feb 2003.

[5] R. H. Arpaci-Dusseau, A. C. Arpaci-Dusseau, D. E. Culler, J. M. Hellerstein, and D. A. Patterson. "The architectural costs of streaming I/O: a comparison of workstations, clusters, and SMPs," Proc. 4th Symposium on High-Performance Computer Architecture (HPCA-4), pages 90 - 101, February 1998.

[6] Mellanox Technologies. “Comparative I/O Analysis: InfiniBand Compared with PCI-X, Fiber Channel, Gigabit Ethernet, Storage over IP, HyperTransport, and RapidIO.” Mellanox Technologies White Paper, 2001. http://www.mellanox.com/technology/shared/IOcompare_WP_140.pdf.

[7] Andrea Emilio Rizzoli. “A Collection of Modelling and Simulation Resources on the Internet”, April 2004, http://www.idsia.ch/~andrea/simtools.html.

[8] Min Xu, Milo Martin+, Doug Burger*, and Mark Hill. “WWW Computer Architecture Page“, 2004, http://www.cs.wisc.edu/~arch/www/tools.html.

[9] John R. Mashey. Big Data and the Next Wave of InfraStress Problems, Solutions, Opportunities. Invited Talk, USENIX 1999. http://www.usenix.org/events/usenix99/invited_talks/mashey.pdf.

[10] Ajay Tirumala, Feng Qin, Jon Dugan, Jim Ferguson, Kevin Gibbs. “Iperf”, March 2003, http://dast.nlanr.net/Projects/Iperf/#whatis.

[11] Lawrence A. Rowe, Steve Smoot, Ketan Patel, and Brian Smith. “MPEG Video Software Statistics Gatherer.” Computer Science Division-EECS, Univ. of Calif. at Berkeley, February 1st, 1995.

[12] Vladimir Afanasiev. “Cache Burst 32”. October 17, 2002. http://user.rol.ru/~dxover/cburst/.

[13] A Beginner’s Guide for MPEG-2 Standard. http://www.fh-friedberg.de/fachbereiche/e2/telekom-labor/zinke/mk/mpeg2beg/beginnzi.htm.

[14] J. Postel. “User Datagram Protocol”, Request for Comments 768, Internet Engineering Task Force, August 1980.

[15] DARPA Internet Program. “Transmission Control Protocol”, Request for Comments 793, Internet Engineering Task Force, September 1981.

[16] Alessandro Rubini & Jonathan Corbet. “Linux Device Drivers, 2nd Edition, Chapter 14, Network Drivers”. June 2001.

[17] Glenn Herrin. “Linux IP Networking: A Guide to the Implementation and Modification of the Linux Protocol Stack”. May 31, 2000.