Combining TCP/UDP/IP with Active Messages over ATM

Kimberly Keeton
Cedric Krumbein
Richard Martin
{kkeeton, ced, rmartin}@cs.berkeley.edu

Computer Science 294-5
University of California, Berkeley
December 2, 1994

Objective:

- To support traditional LAN communication protocols (TCP/UDP/IP) and high-speed protocols (AM) over the same ATM network.
- Implement protocols as peers, without encapsulating one within the other.

  1. Maintain low latency of AM.
  2. Maintain wide-area TCP/UDP/IP compatibility.

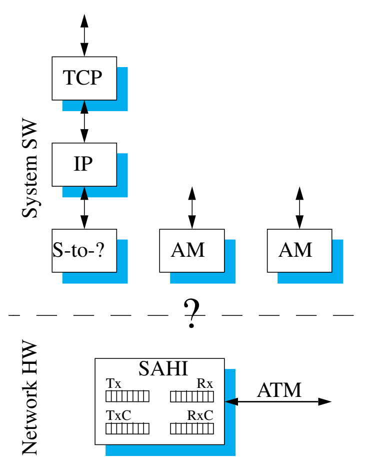
Goals:

- Develop a combined protocol strategy that will work on various workstation platforms (Sun, HP, ...) and network media (ATM, Myrinet, ...).
- Minimize impact to performance of each protocol.
- Robust support of each protocol.

Issues in Sharing the Network:

- Multiplexing/demultiplexing.
- Protection.
- Address translation.
- Efficiency (no copying).
- Interface to existing software and hardware.

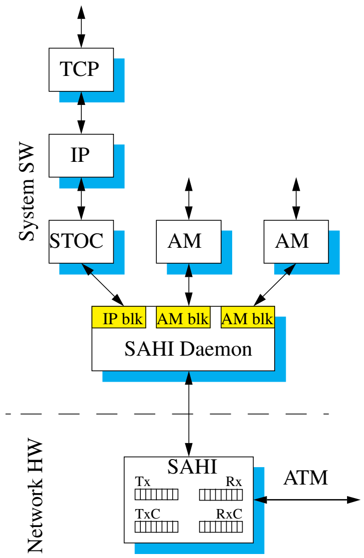
How to implement?



Options:

- Use multiple SAHIs. Give each protocol its own network interface.
- Use usual kernel method. All modules are kernel-resident, packets are copied between modules.
- Multiplex in hardware. Use SAHI's multiple Transmit Rings: allocate one ring to each protocol.
- Multiplex in software, within SAHI daemon.

Multiplex IP and ATM within SAHI Daemon



SAHI daemon resides on its own dedicated processor.

Software must handle all networking issues:

- Multiplexing (round-robin polling on each process).
- Demultiplexing.
- Protection.
- Some copying needed.

How to Evaluate:

- Microbenchmarks to compare performance of combined *vs.* individual protocol implementations.

  1. Round trip.
  2. Bulk transfer.

- NOW-like applications to quantify effects of network sharing.

IP	AM
Laddis (NFS benchmark)	Ray tracer (split-C)
MPEG (video player, X11)	

Status:

- Not working yet! We hope to have the software debugged very shortly.

Conclusions:

- Hard! SAHI is a complex network interface designed to provide high constant bandwidth, not low latency.
- This scheme requires a dedicated processor to achieve adequate performance.
- A custom network interface would be very useful. Ideally it should provide:

  1. Multiplexing/demultiplexing.
  2. Protection mechanism to isolate different processes.
  3. Address translation.