System Software Design for Multimedia Networking

Jonathan C.L. Liu, Ph.D.

CISE Department

University of Florida

Multimedia Communication Applications

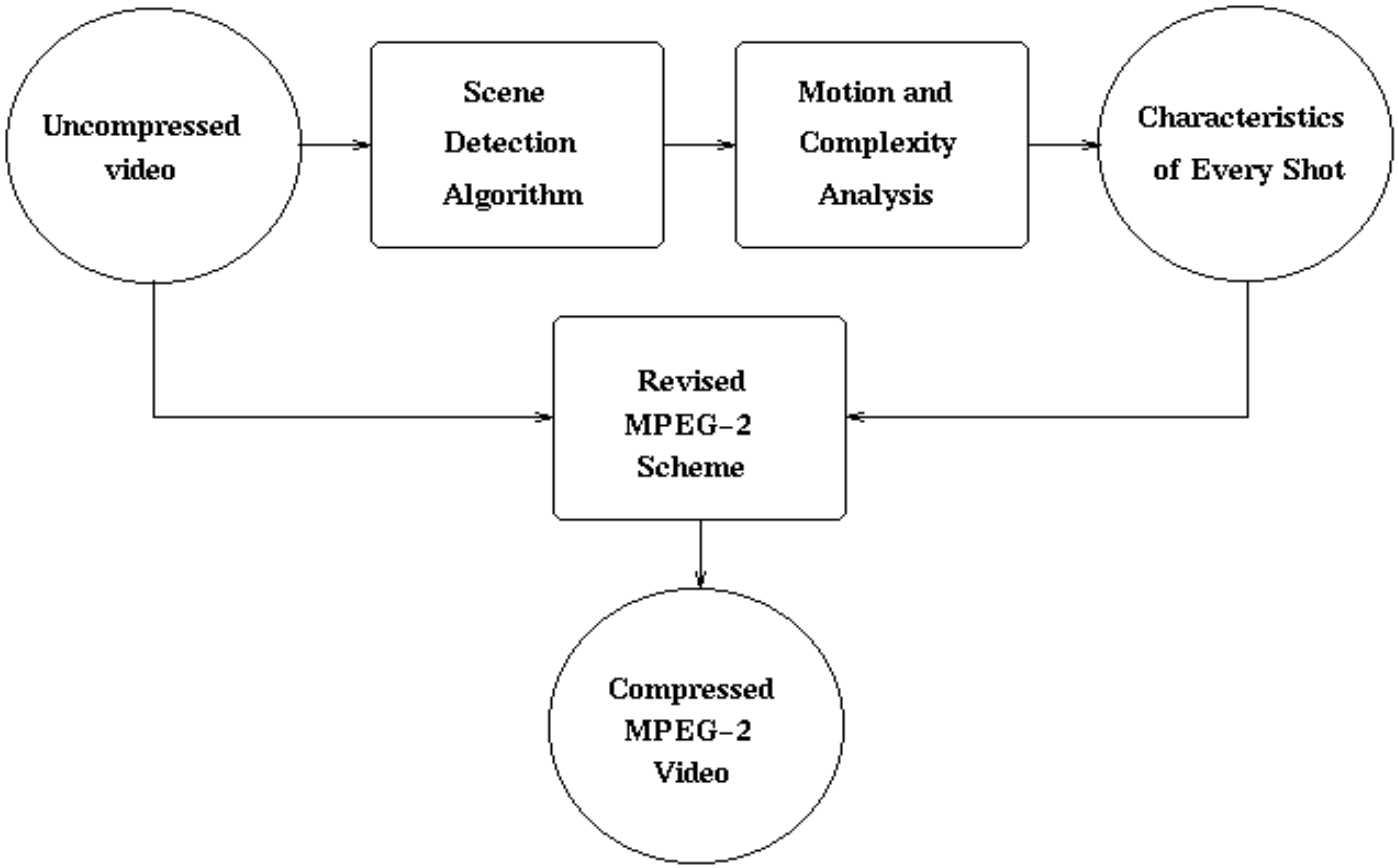
- Video Conferencing Service
  - Distance learning;
  - Company meetings;
- On-Demand Streaming Service
  - Compress the video off-line;
  - Store the video files in storage system;
- Concurrent users with guaranteed jitter-free quality via high-speed networks

Major System Components

- Compression/decompression schemes
- Network protocol at client/router/server
- Multimedia server design
- Intelligent storage system design
- Security and copyright processing
- Anything in-between the components

MPEG Compression Schemes

- Based on DCT transformation;
- Four types of frames: I, P, B and D
- MPEG-2: 2 Mbps - 100 Mbps
  - Computation intensive for HDTV quality
  - Our adaptive MPEG-2 encoder can save up to 48% of the bandwidth/storage
- MPEG-4: 64 Kbps – 4 Mbps



- J. Liu, ``Bandwidth-Efficient MPEG-2 Encoding", "Advances in Distributed Multimedia Systems", Edited by T. Znati, pp. 257--283, ISBN 981-02-3560-7, World Scientific Publishing, Jul. 1999.

(10-)Gigabit Ethernet LAN/MAN

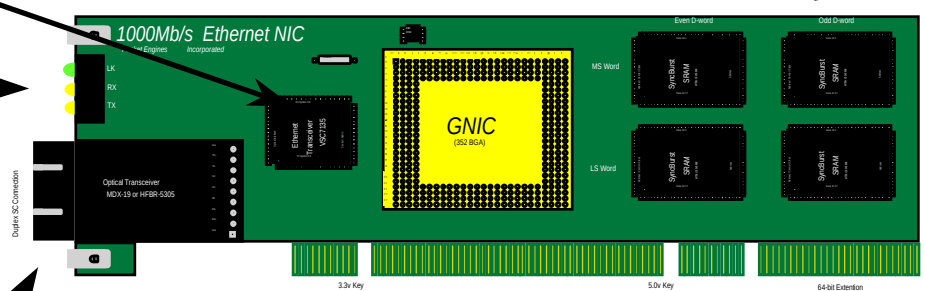
B. Daines, J. Liu and K. Sivalingam, ``Supporting Multimedia Communication over A Gigabit Ethernet Network", International Journal of Parallel and Distributed Systems and Networks, 4(2): 102--115,Jun. 2001

GbE PHY
Serializer/Deserializer

Yellowfin ASIC

512kB Local Packet Memory
32kB transmit & 480kB receive

LEDs
tx/rx/link valid



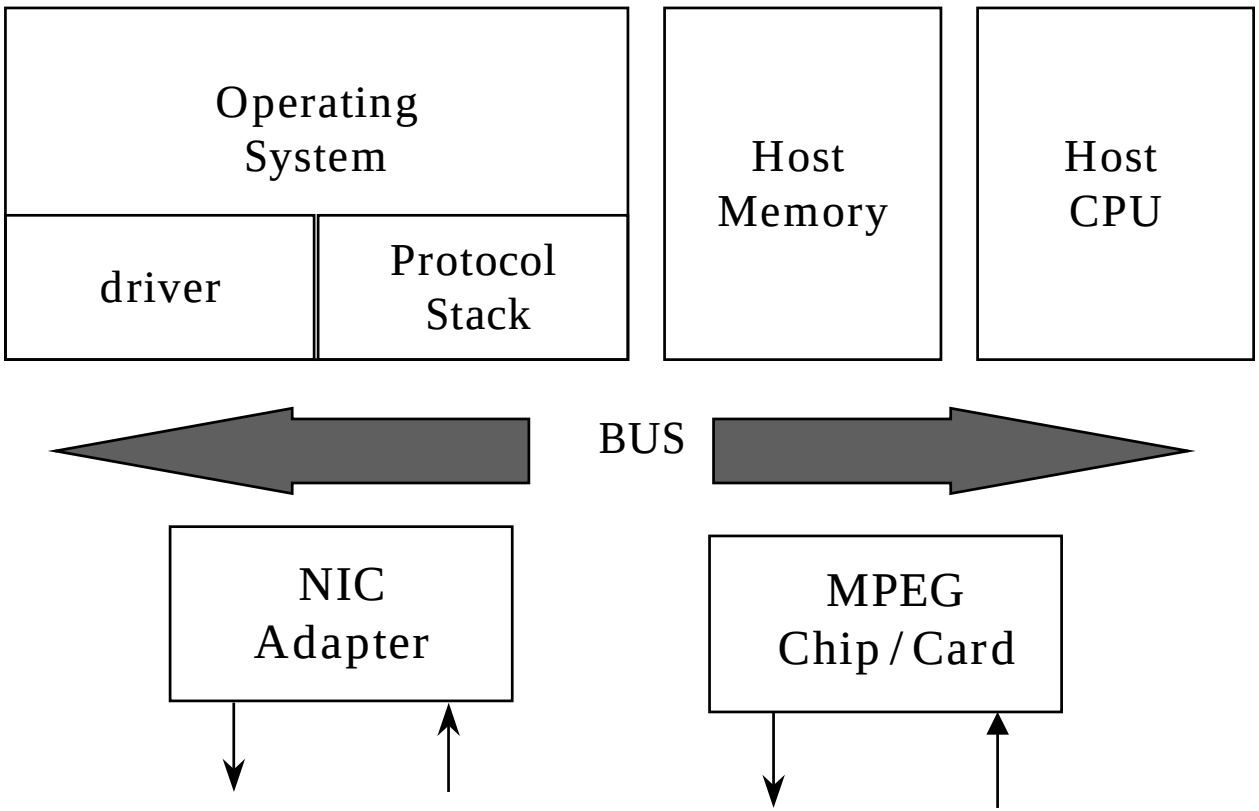
1000BASE-SX
Short Wavelength Laser
Optical Transceiver

PCI v2.1 Connector
Yellowfin ASIC- 'Auto-Negotiates"
32bit or 64 bit bus mode

Problem Nature

- MM Communication Application level: ??
- Operating System level:
  - Linux 19% vs. Windows 2000: 9%
- TCP/UDP/IP levels: 25%~30%
- Device Driver level: 35%~40%
- Hardware/firmware level: 70% - 80%
- Need an optimal design integration

Supporting the Mission??

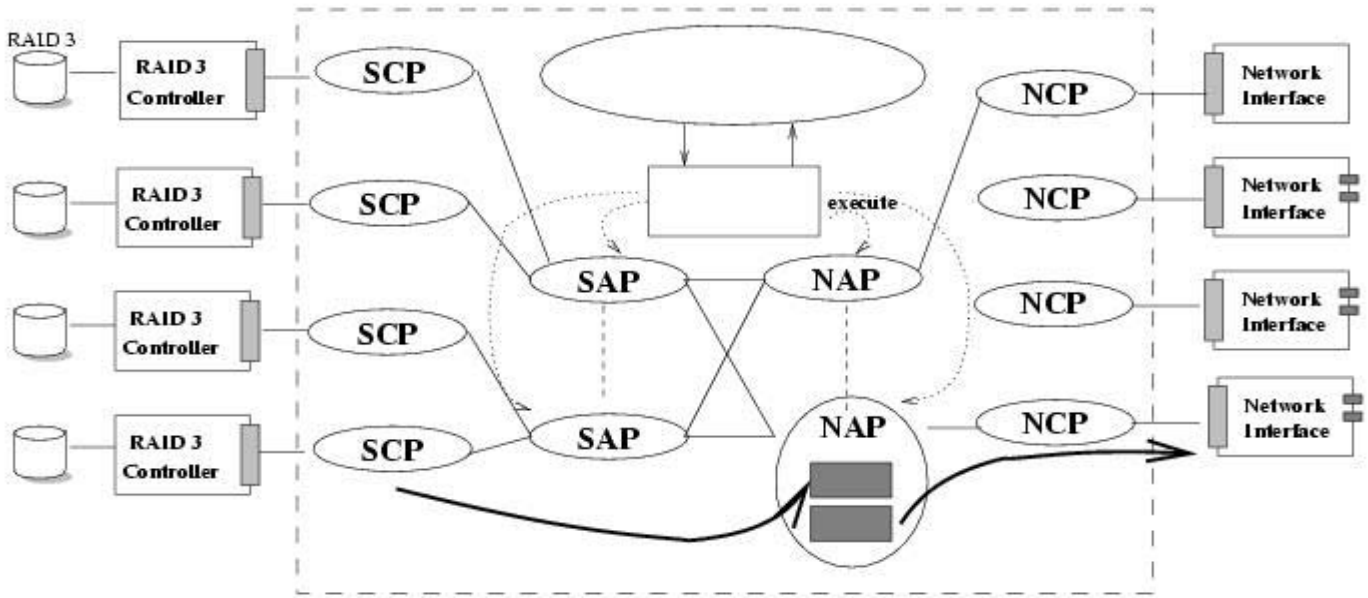


Possible Approaches

- Have the drivers implemented in firmware
- Have (part of) IP protocol in firmware
- Have (part of) UDP protocol in firmware
- Have (part of) TCP protocol in firmware
- (10-)Gigabit Ethernet cards do support eight priorities, but rarely utilized by software
- WHERE and HOW should we integrate them?

Server Design Considerations

- Needs to support a large number of concurrent accesses on stored videos
- Each stream should guarantee the application-level quality of 1-3% jitters
- What is the right software design within the multimedia servers?
- Block sizes (e.g., 16-frame) need to be adjusted for the number of accesses



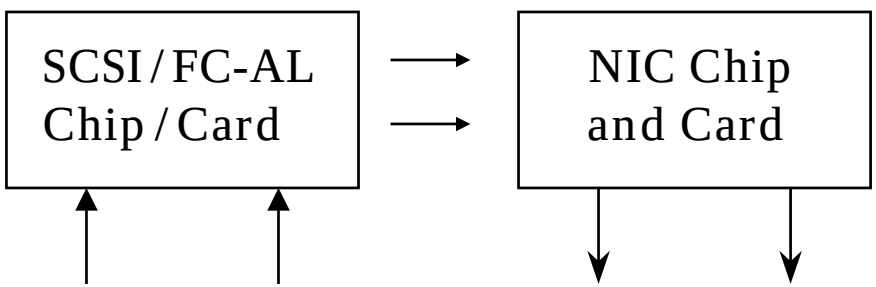
SCP: Storage Control Process
SAP: Storage Access Process
NCP: Network Control Process
NAP: Network Access Process

Prototype Status

- Currently running on the Linux OS as the system threads and processes
- Lesson learned: naïve software design caused system over-competition
- System-level coordination and scheduling helped in a significant degree
- Handles spatial and temporal locality

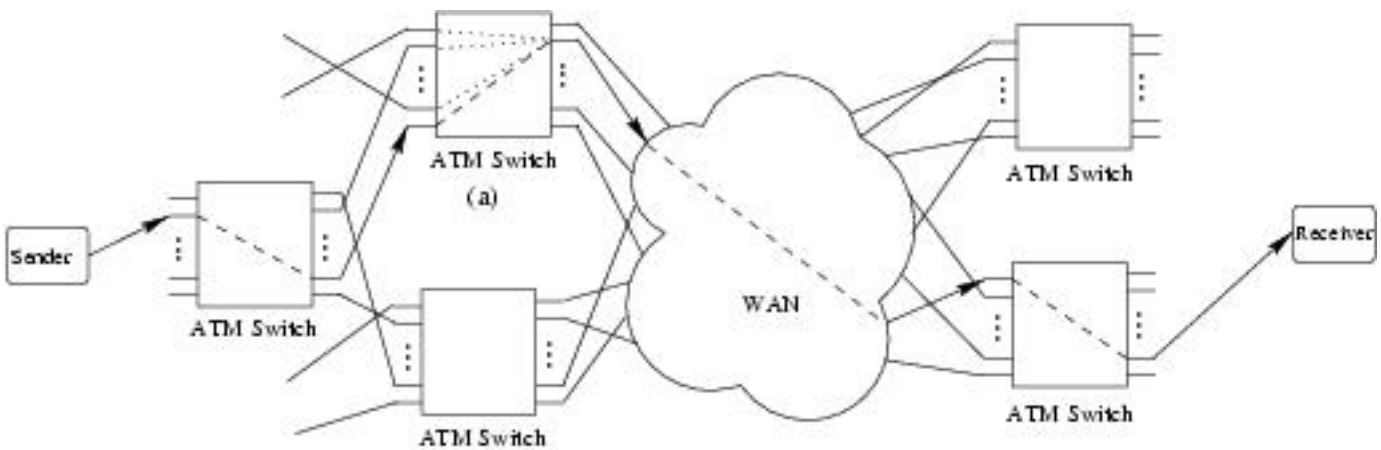
Further Improvements??

- The overall system CPU utilization is low (e.g., 5-10%) for normal operations
- The majority of the operations can be done by the jumbo-card design??

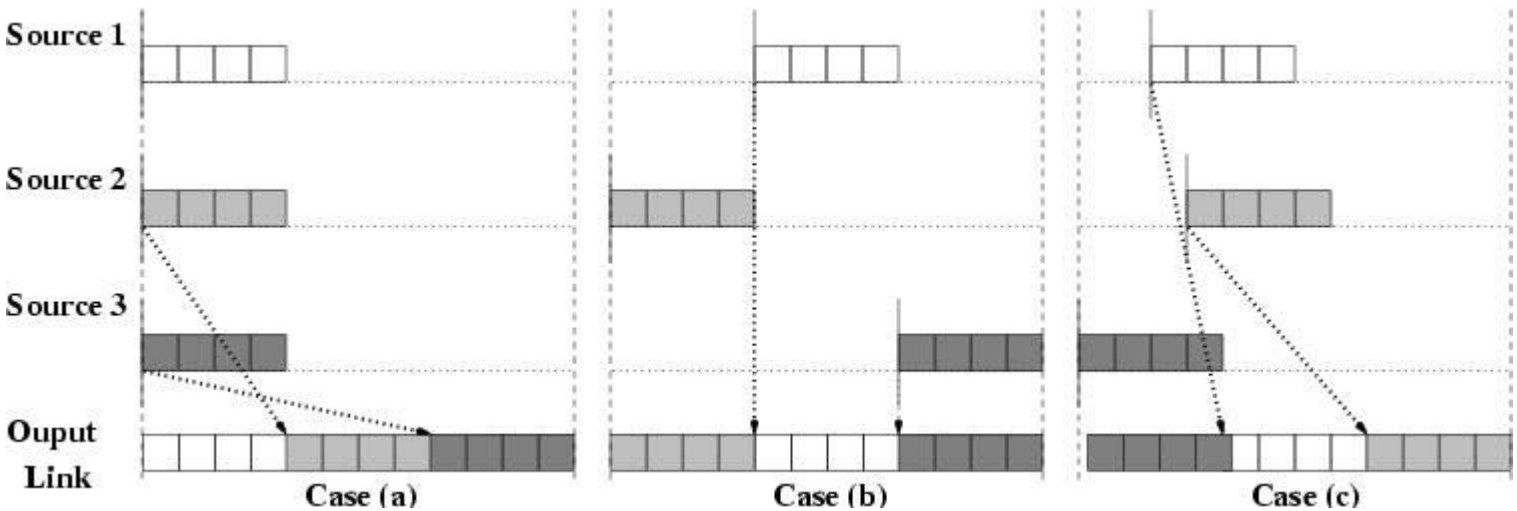


WAN Connection

- Eventually video streams require the support of ATM networks (or at least SONET) for long-distance connections.



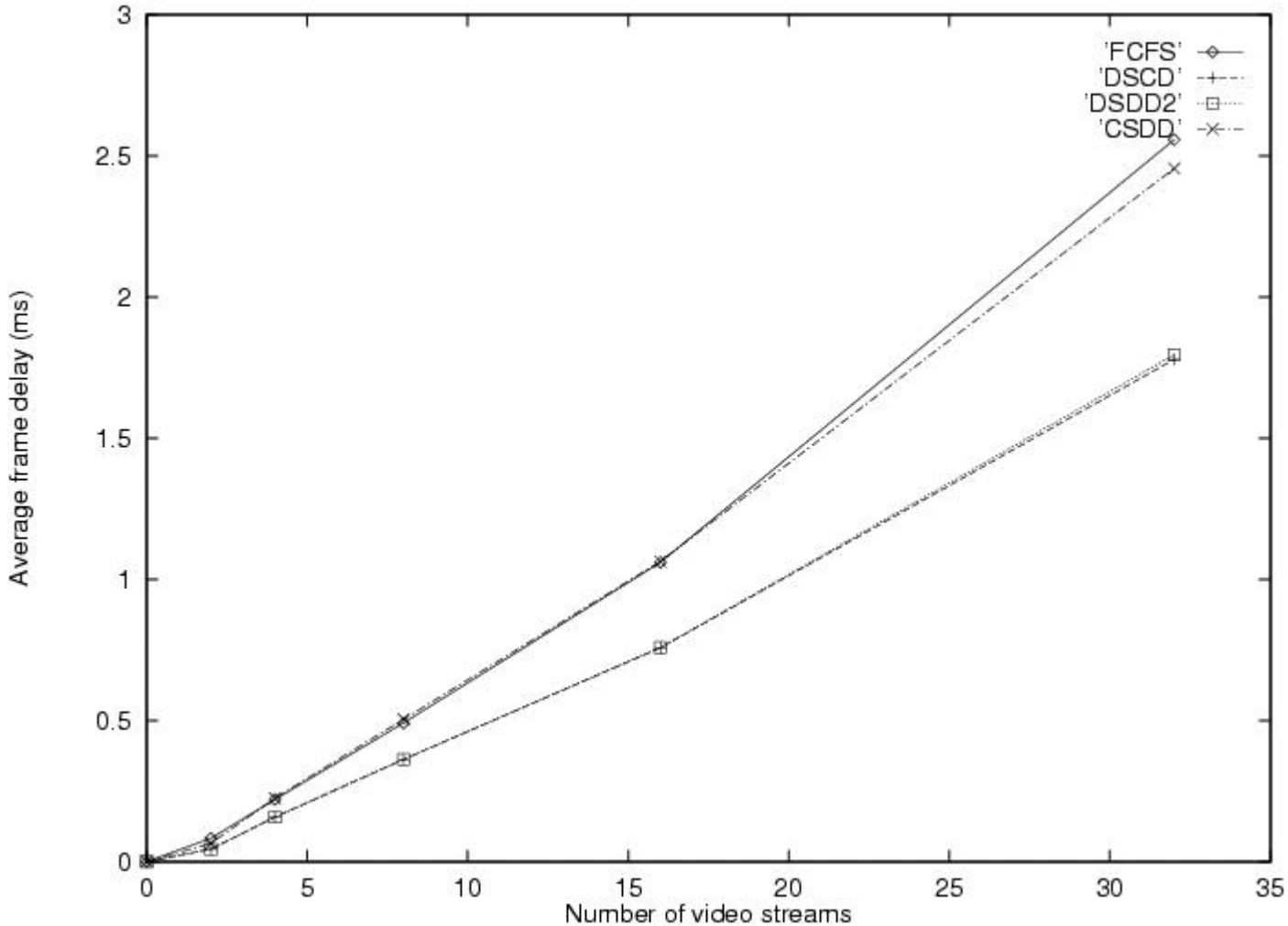
Problem Nature



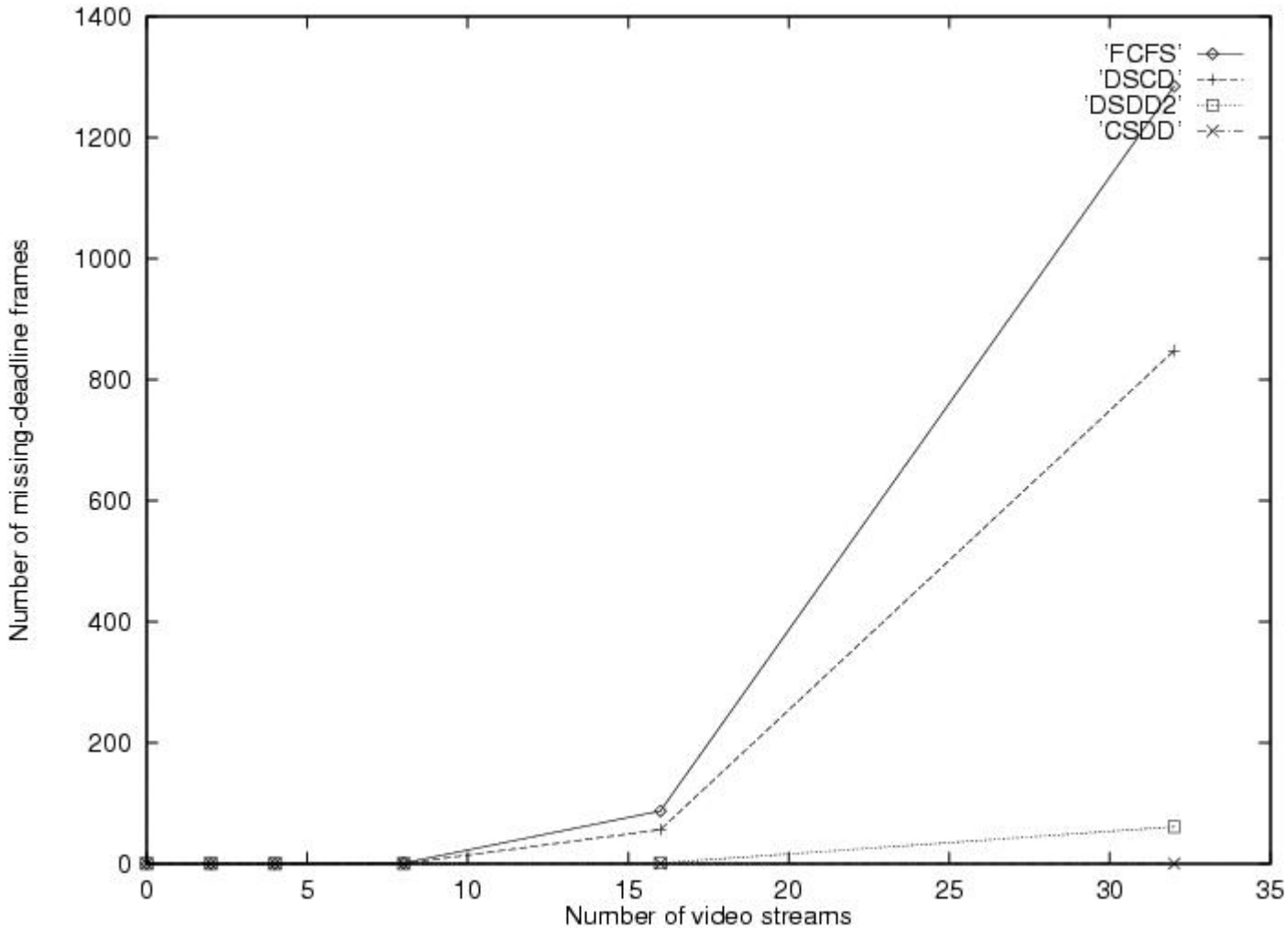
Proposed Algorithm

```
1  DSDD2(N, VC_i)
2  {
3        if (there is data in any of the N sub-buffer-queues)
4           for (i=1 to N)
5              Insert the head of sub-queue_i to the set CURRENT;
6           end for
7           FIRST = ∅;
8           repeat
9              oldsize =| FIRST |;
10             FIRST = ∅;
11             Find VC_i which has the maximal size in CURRENT;
12             for all VC_j such that (VC_j ∈ CURRENT) and (VC_j ≠ VC_i)
13                if (deadline_j − p_j < s_i);
14                   put VC_j into the set FIRST;
15                end if
16             end for
17             for all VC_j such that (VC_j ∈ CURRENT) and (VC_j ≠ VC_i)
18                if (deadline_i − p_i < s_j);
19                   put VC_i into the set FIRST and break;
20                end if
21             end for
22             CURRENT = FIRST;
23          until (| FIRST |== oldsize) or (| FIRST |< 2);
24          if (| FIRST |== 0)
25             Move the frame with the smallest size in CURRENT out of buffer queue;
26          else if (| FIRST |== 1)
27             Move the only frame in FIRST out of buffer-queue;
28          else /* (| FIRST |> 1 */
29             Move the frame with the smallest size in FIRST out of buffer-queue ;
30          end if
31          Compute Δ = Δ + (current time - start_time);
32          if ((deadline − p) < 0)
33             M = M + 1;
34          end if
35          served = served + 1;
36       Serve the frame on the output link;
37       end if
38       for (i=1 to N)
39          Scan for new incoming data for VC_i, and record the start_time;
40          Insert the new incoming frame to the end of sub-queue^i;
41       end for
```

Delay Performance



Deadline Performance



More Design Issues

- Integrated Priority Design
  - Application-level: ??
  - IPv6: 2^8=256 traffic classes
  - High-speed Ethernet: 2^3=8
  - ATM: 2^1=2
- Joint design with ADSL/Cable Modem
- Joint design with DVD players/recoders

Talk Summary

- Performance guarantee needs to be ensured all the way via all components
- System can benefit significantly from the software/firmware/hardware co-design
- Require the close collaboration between EE/CE/CS researchers in all fields