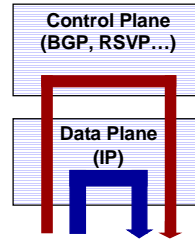


Preliminary Experiences Building an Extensible Router

Larry Peterson
Princeton University

Router Architecture

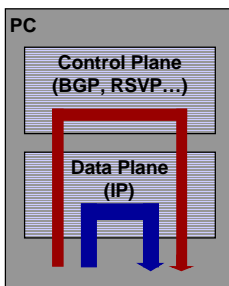


November 2001

Network Systems Group

2

Software-Based Router



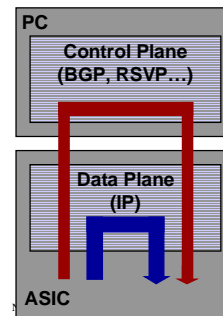
- + Cost
- + Programmability
- Performance (~300 Kpps)
- Robustness

November 2001

Network Systems Group

3

Hardware-Based Router



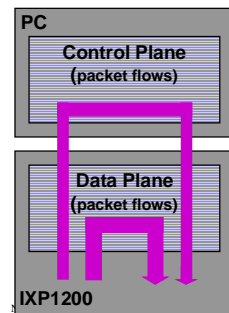
- Cost
- Programmability
- + Performance (25+ Mpps)
- + Robustness

2

Systems Group

4

Our Router Architecture



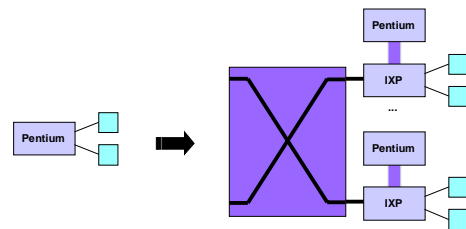
- + Cost (\$1500)
- + Programmability
- ? Performance
- ? Robustness

IXP1200

Systems Group

5

In General...

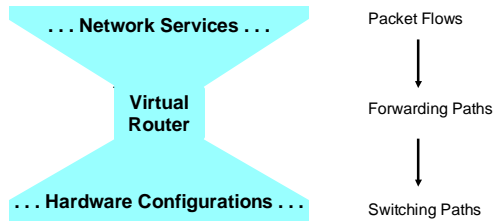


November 2001

Network Systems Group

6

Architectural Overview



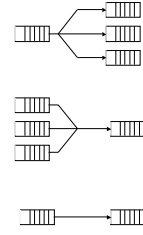
November 2001

Network Systems Group

7

Virtual Router

- Classifiers
- Schedulers
- Forwarders

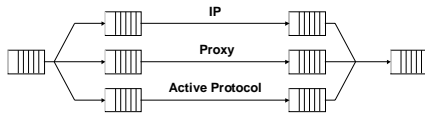


November 2001

Network Systems Group

8

Simple Example



November 2001

Network Systems Group

9

Research Problems

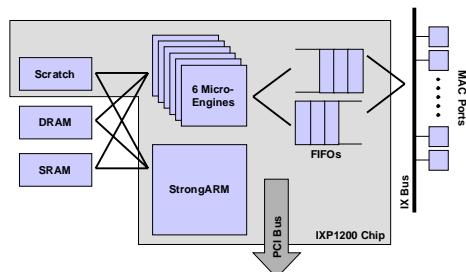
- How to program
 - granularity
 - trust
 - control
- Map forwarding paths onto switching paths
 - local/static: implement VR on one processor
 - local/dynamic: scheduling
 - global/static: function placement
 - global/dynamic: resource allocation

November 2001

Network Systems Group

10

Intel IXP

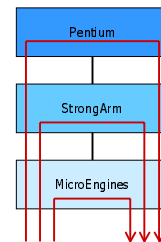


November 2001

Network Systems Group

11

Processor Hierarchy

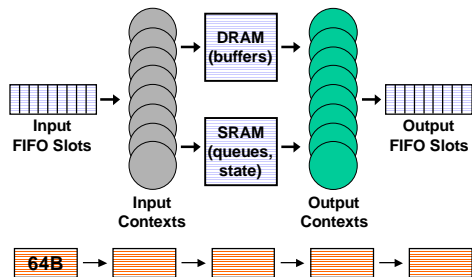


November 2001

Network Systems Group

12

Data Plane Pipeline



November 2001

Network Systems Group

13

Data Plane Processing

INPUT context loop

```
wait_for_data
copy in_fifo→regs
Basic_IP_processing
copy regs→DRAM
if (last_fragment)
  enqueue→SRAM
```

OUTPUT context loop

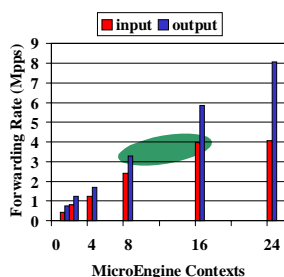
```
if (need_data)
  select_queue
  dequeue←SRAM
copy DRAM→out_fifo
```

November 2001

Network Systems Group

14

Pipeline Evaluation



Measured independently

100Mbps Ether \approx 0.142Mpps

November 2001

Network Systems Group

15

What We Measured

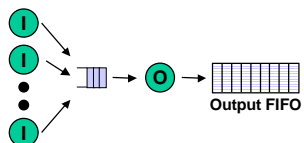
- Static context assignment
 - 16 input / 8 output
- Infinite offered load
- 64-byte (minimum-sized) IP packets
- Three different queuing disciplines

November 2001

Network Systems Group

16

Single Protected Queue



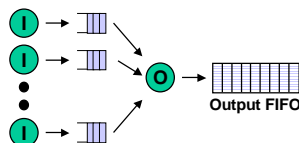
- Lock synchronization
- Max 3.47 Mpps
- Contention lower bound 1.67 Mpps

November 2001

Network Systems Group

17

Multiple Private Queues



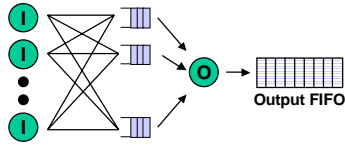
- Output must select queue
- Max 3.29 Mpps

November 2001

Network Systems Group

18

Multiple Protected Queues



- Output must select queue
- Some QoS scheduling (16 priority levels)
- Max 3.29 Mpps

November 2001

Network Systems Group

19

Data Plane Processing

INPUT context loop

```
wait_for_data
copy in_fifo→regs
Basic_IP_processing
copy regs→DRAM
if (last_fragment)
enqueue→SRAM
```

OUTPUT context loop

```
if (need_data)
select_queue
dequeue←SRAM
copy DRAM→out_fifo
```

November 2001

Network Systems Group

20

Cycles to Waste

INPUT context loop

```
wait_for_data
copy in_fifo→regs
Basic_IP_processing
nop
nop
...
nop
copy regs→DRAM
if (last_fragment)
enqueue→SRAM
```

OUTPUT context loop

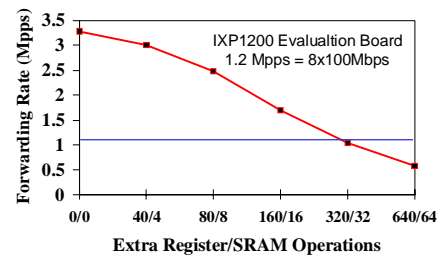
```
if (need_data)
select_queue
dequeue←SRAM
copy DRAM→out_fifo
```

November 2001

Network Systems Group

21

How Many "NOPs" Possible?



November 2001

Network Systems Group

22

Data Plane Extensions

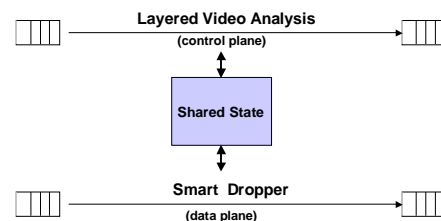
Processing	Memory Ops	Register Ops
Basic IP	6	32
TCP Splicer	6	45
TCP SYN Monitor	1	5
ACK Monitor	3	15
Port Filter	5	26
Wavelet Dropper	2	28

November 2001

Network Systems Group

23

Control and Data Plane



November 2001

Network Systems Group

24

What About the StrongARM?

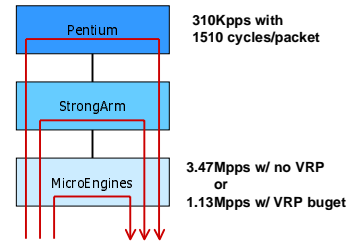
- Shares memory bus with MicroEngines
 - must respect resource budget
- What we do
 - control IXP1200 ↔ Pentium DMA
 - control MicroEngines
- What might be possible
 - anything within budget
 - exploit instruction and data caches
- We recommend against
 - running Linux

November 2001

Network Systems Group

25

Performance



November 2001

Network Systems Group

26

Pentium

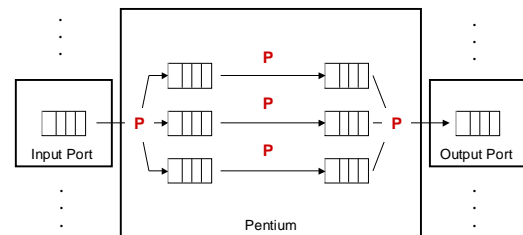
- Runs protocols in the control plane
 - e.g., BGP, OSPF, RSVP
- Run other router extensions
 - e.g., proxies, active protocols, overlays
- Implementation
 - runs Scout OS + Linux IXP driver
 - CPU scheduler is key

November 2001

Network Systems Group

27

Processes

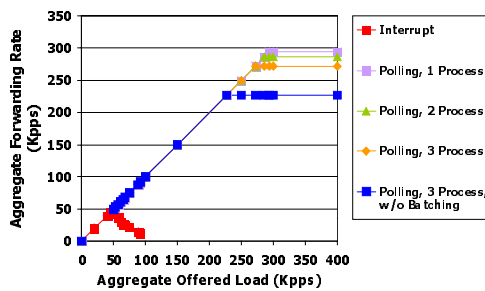


November 2001

Network Systems Group

28

Performance

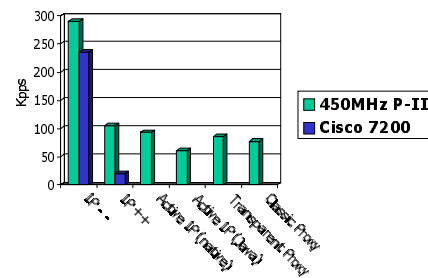


November 2001

Network Systems Group

29

Performance (cont)



November 2001

Network Systems Group

30

Scheduling Mechanism

- Proportional share forms the base
 - each process reserves a cycle rate
 - provides isolation between processes
 - unused capacity fairly distributed
- Eligibility
 - a process receives its share only when its source queue is not empty and sink queue is not full
- Batching
 - to minimize context switch overhead

November 2001

Network Systems Group

31

Share Assignment

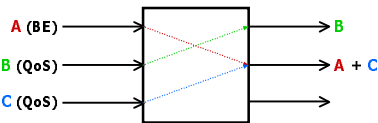
- QoS Flows
 - assume link rate is given, derive cycle rate
 - conservative rate to input process
 - keep batching level low
- Best Effort Flows
 - may be influenced by admin policy
 - use shares to balance system (avoid livelock)
 - keep batching level high

November 2001

Network Systems Group

32

Experiment



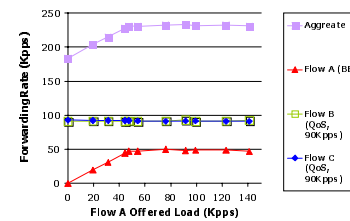
November 2001

Network Systems Group

33

Mixing Best Effort and QoS

- Increase offered load from A



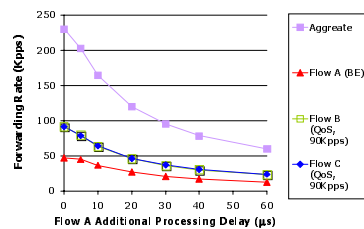
November 2001

Network Systems Group

34

CPU vs Link

- Fix A at 50Kpps, increase its processing cost

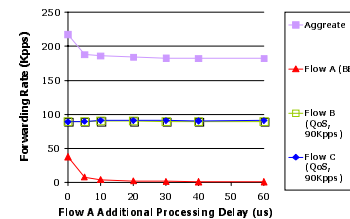


November 2001

Network Systems Group

35

Turn Batching Off



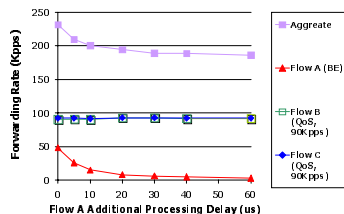
- CPU efficiency: 66.2%

November 2001

Network Systems Group

36

Enforce Time Slice



- CPU efficiency: 81.6% (30us quantum)

November 2001

Network Systems Group

37

Batching Throttle

- Scheduler Granularity: G
 - flow processes as many packets as possible w/in G
- Efficiency Index: E , Overhead Threshold: T
 - keep the overhead under $T\%$, then $1 / (1+T) < E$
- Batch Threshold: B_i
 - don't consider Flow i active until it has accumulated at least B_i packets, where $C_{sw} / (B_i \times C_i) < T$
- Delay Threshold: D_i
 - consider a flow that has waited D_i active

November 2001

Network Systems Group

38

Dynamic Control

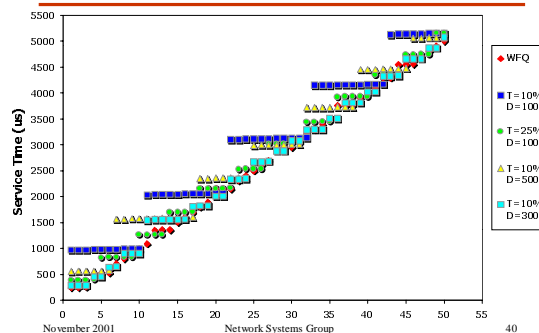
- Flow specifies delay requirement D
- Measure context switch overhead offline
- Record average flow runtime
- Set E based on workload
- Calculate batch-level B for flow

November 2001

Network Systems Group

39

Packet Trace



November 2001

Network Systems Group

40

Acknowledgements

- Graduate Students
 - Andy Bavier
 - Yitzhak Gottlieb
 - Scott Karlin
 - Tammo Spalink
 - Xiaohu Qie
- More Information
 - www.cs.princeton.edu/nsg/

November 2001

Network Systems Group

41

Research Agenda

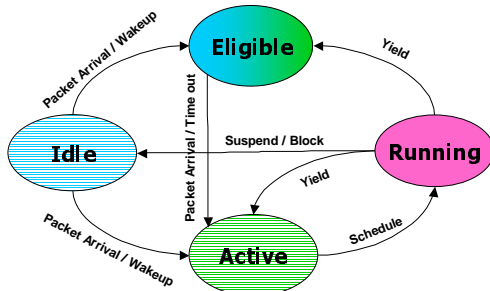
- Evaluate the use of commodity components to build IP routers.
- Design interfaces and mechanisms needed to support extensibility.

November 2001

Network Systems Group

42

State Machine



November 2001

Network Systems Group

43

Simulation

- Setup

- CPU: 1MHz
- Context-Switch: 3μs
- 16 QoS Flows: 10Kpps, 3μs/pkt
- 1 BE Flow: 100Kpps, 3μs/pkt

In order to forward all packets, E must be at least
 $10 \times 10^3 \times 3 \times 10^{-6} \times 16 + 100 \times 10^3 \times 3 \times 10^{-6} = 78\%$

- Parameters:

- G=100μs, T=10% / 25%
- QoS Flows: $D_i = 1\text{ms} / 0.5\text{ms} / 0.3\text{ms}$

November 2001

Network Systems Group

44