

---

# DiffServ over Network Processors: Implementation and Evaluation

---

Authors: Ying-Dar Lin, Yi-Neng, Shun-Chin  
Yang, and Yu-Shen Lin  
Speaker: Yi-Neng Lin

Department of Computer and Information Science  
National Chiao Tung University  
Hsinchu, Taiwan

---

## Outline

- ❑ Motivation
- ❑ Introduction
- ❑ Hardware architecture of IXP1200
- ❑ Design and implementation of DiffServ over IXP1200
- ❑ External benchmark
- ❑ Internal benchmark
- ❑ Conclusions and future works

## Motivation

- ❑ Scalable data-plane processing in Firewall, DiffServ, and WebSwitch
- ❑ Solutions
  - ❑ General processor
  - ❑ General processor + ASIC
  - ❑ General processor + co-processors (NP)
- ❑ Offload the data-plane processing to co-processors

2002/8/22

7

## Introduction(1/2)

- ❑ Why Network Processor?
  - ❑ Scalability and programmability
  - ❑ Hardware-based threads in co-processors, zero context swap overhead
  - ❑ Specifically designed instruction set for networking purpose

Instructions of EXP1200	Instruction description	Instructions of x86 processor
ALU	Perform ALU with shift in one instruction	ALU (ADD or SUB) + shift
IMMED	Load an immediate value with shift	Load + shift
FIND_BSET LOAD_BSET	1. Determine the position of the first set bit in a 16-bit field of a register 2. Shift option provided	At least 5 instructions to test one single bit
BR_BSET	Branch if the specified bit in a register is set	Shift + bit test + JUMP
HASH1_64	Perform one 64-bit hash operation	Many instructions needed

2002/8/22

7

## Introduction(2/2)

- ❑ Related work--IP forwarding over IXP1200 [Spalink, SOSP'18]
  - ❑ Identify that **SDRAM** is the bottleneck in **IP forwarding**
  - ❑ May not be generalized for other complex applications
- ❑ Investigated issues
  - ❑ Map DiffServ modules to IXP1200
  - ❑ Flow scalability
  - ❑ Aggregate throughput
  - ❑ Internal simulations
  - ❑ Bottlenecks of IXP1200 in DiffServ

2002/8/22

7

## Hardware Architecture

2002/8/22

7

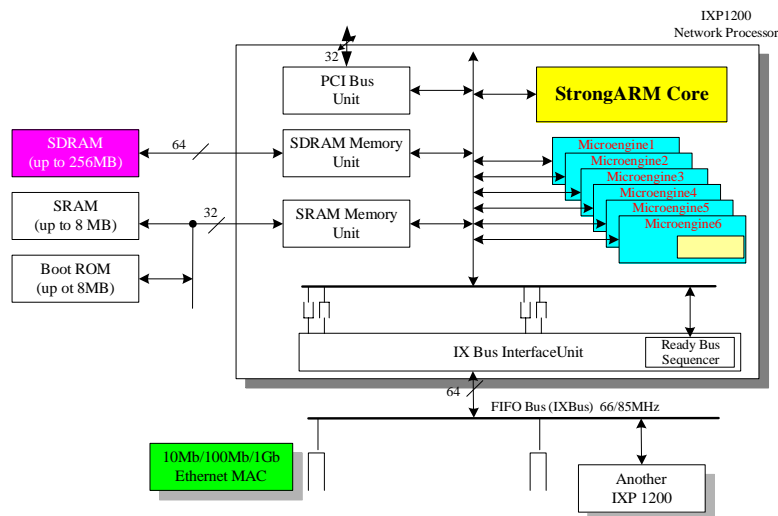
# Components of IXP1200

- ❑ Processors
  - ❑ Integrated StrongArm Core for control-plane
  - ❑ Six integrated programmable microengines for data-plane. Each has four threads
- ❑ Interfaces and Storage
  - ❑ 64-bit IX Bus, 4.2Gbps peak bandwidth
  - ❑ 32MB, 64-bit SDRAM interface (up to 256MB)
  - ❑ 2MB, 32-bit SRAM interface (up to 8MB)
  - ❑ 2K instruction headroom (named “control store”) for each microengine

2002/8/22

7

# IXP1200 Block Diagram



2002/8/22

8

## Design and Implementation of DiffServ over IXP1200

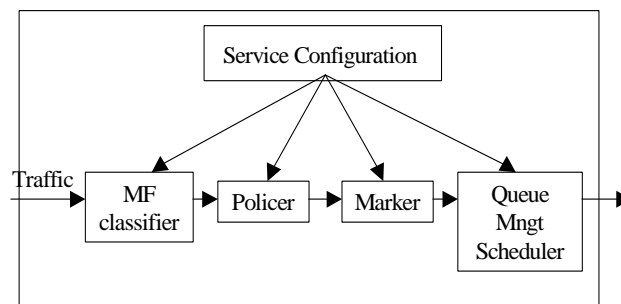
- ❑ Packet flow in a DiffServ edge router
- ❑ Data-Plane Architecture
- ❑ Detail packet flow chart
- ❑ Algorithm description

2002/8/22

7

## Packet Flow in a DiffServ Edge Router

DiffServ (Differentiated Services) edge router

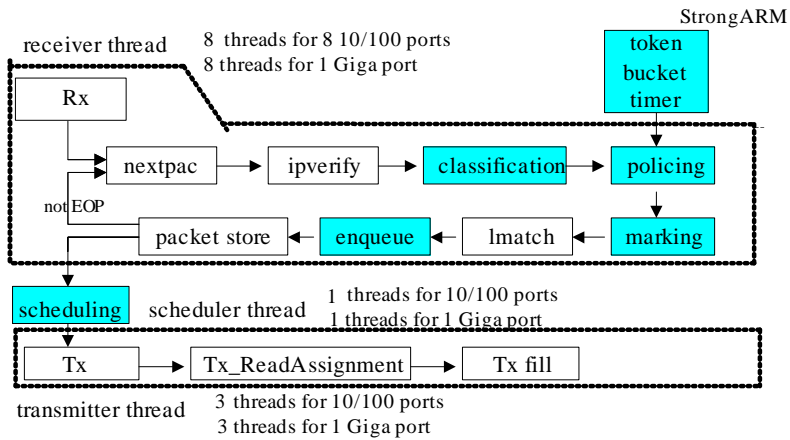


2002/8/22

8

# Data-Plane Architecture

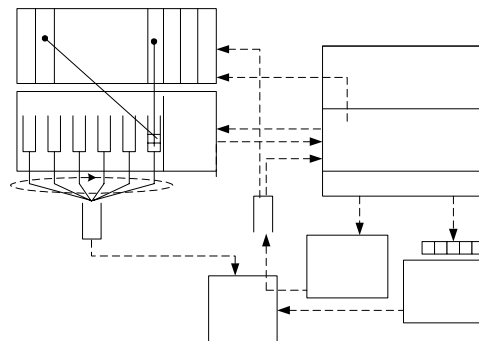
Thread allocation: consider the size of the control store (2K)



2002/8/22

7

# DiffServ Packet Flow in IXP1200



The basic unit in IXP1200 is the 64 bytes MAC Packet (MP)

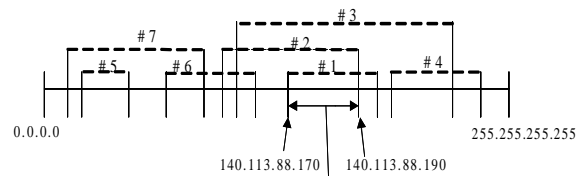
2002/8/22

2

## Multi-Dimensional Range Matching [Lakshman, 1998]

- A rule is a pair of IP addresses which specifies a range
- Use Bit Vector table to record the overlapped rules in an interval
- FIND\_BSET can be applied for finding the first matched rule

Source IP dimension



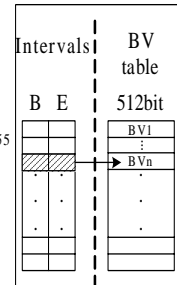
140.113.88.172

Src\_ip of the packet

Time complexity:  $O(\log n)$

Space complexity:  $O(n^2)$

SRAM

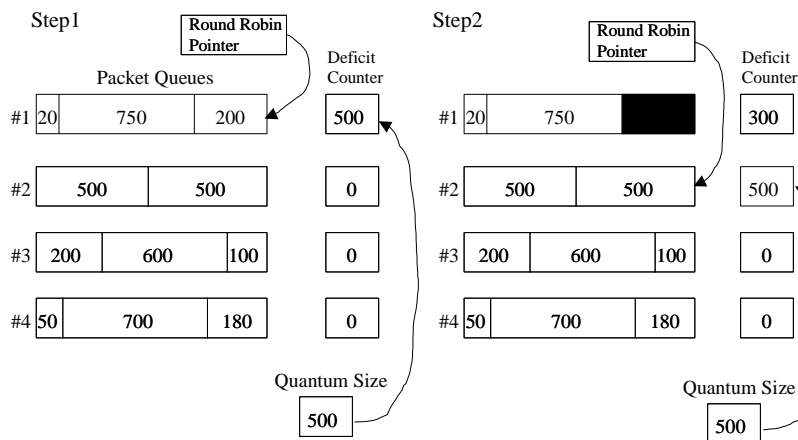


BVn 0 0 1 1 1 bit vector for this interval  
512 511 ... 3 2 1

2002/8/22

7

## Deficit Round Robin [Shreedhar, 1996]

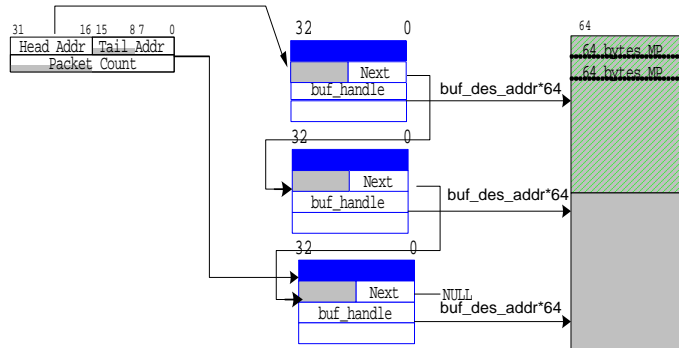


2002/8/22

7

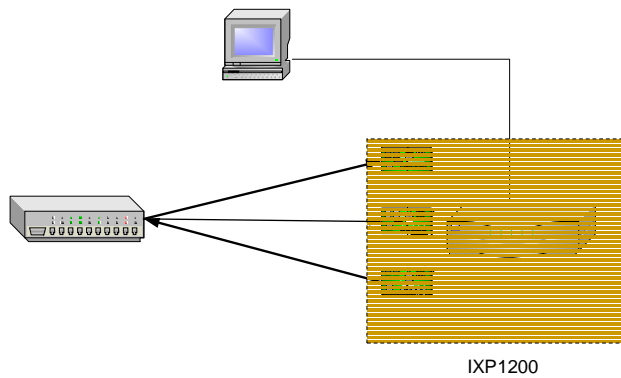
# Packet in the Queue

SRAM\_QUEUE\_DESCRIPTOR\_BASE      SRAM\_BUFF\_DESCRIPTOR\_BASE      SDRAM\_PKT\_BUFF\_BASE  
Per Queue, Per Port                  PACKET\_FREELIST                  Actual Packet Storage



# External Benchmark

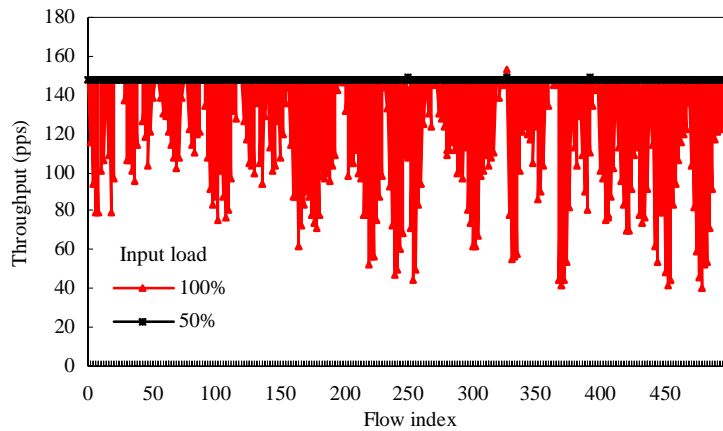
- Scalability test
- Aggregate throughput test



## Scalability Test – 500 EF Flows

Maximum load that results no packet loss = 58%

Flow fairness test (Len=64bytes, input port x1, 500 flows, BW=74400/500=148pps)

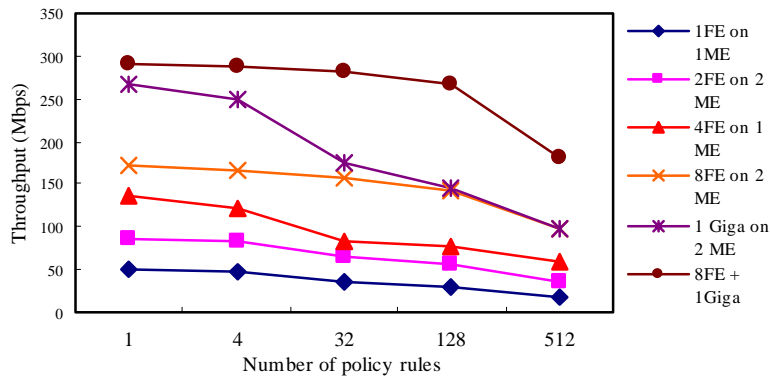


2002/8/22

7

## Aggregate Throughput

Aggregate throughput (Len=64bytes, worst case)



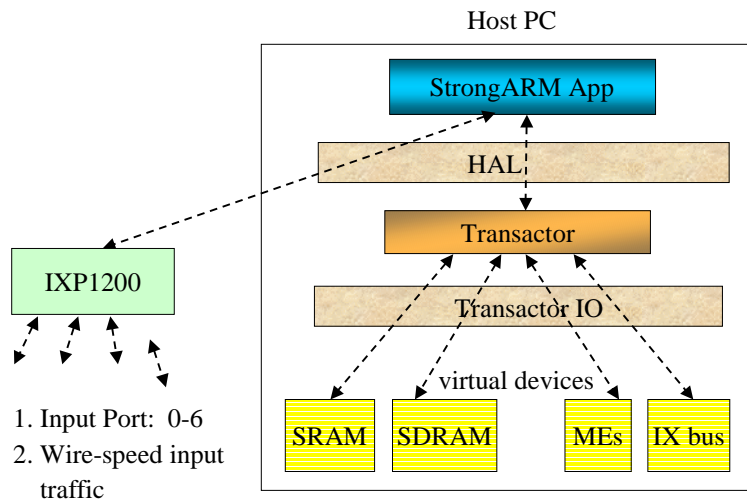
Wire speed (1.8Gbps) in IP forwarding while 290Mbps in DiffServ?

-- The complex computation and the delay of SRAM access

2002/8/22

8

## Internal Benchmark



2002/8/22

7

## Simulation Results

Algorithm for classifier	SRAM Util.	ME Util.	SDRAM Util.	Bottleneck
Linear Search	55%	80%	9%	SRAM
Range Matching	35.3%	100%	13%	ME

### SRAM bottleneck

- A bottleneck unit needs not to be 100% utilized
- Bursty SRAM access
- May also cause an idle microengine

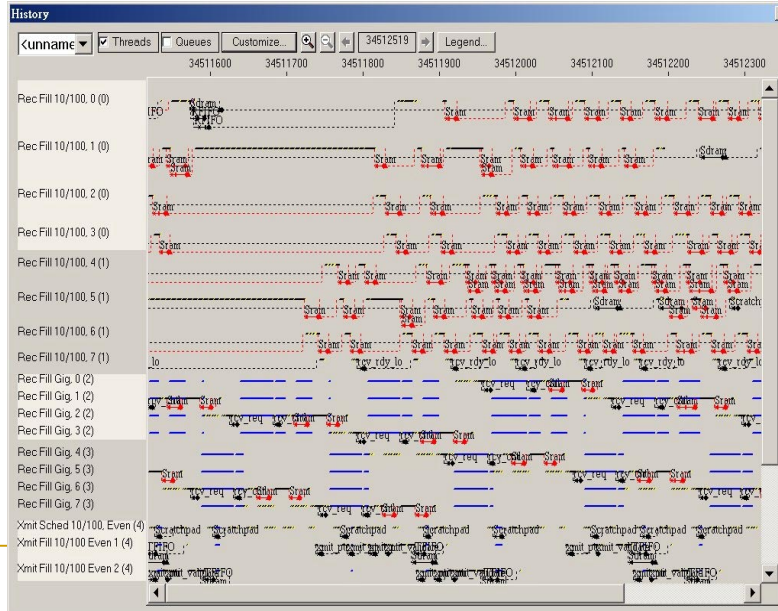
### ME bottleneck

- Complex data calculations in Range Matching

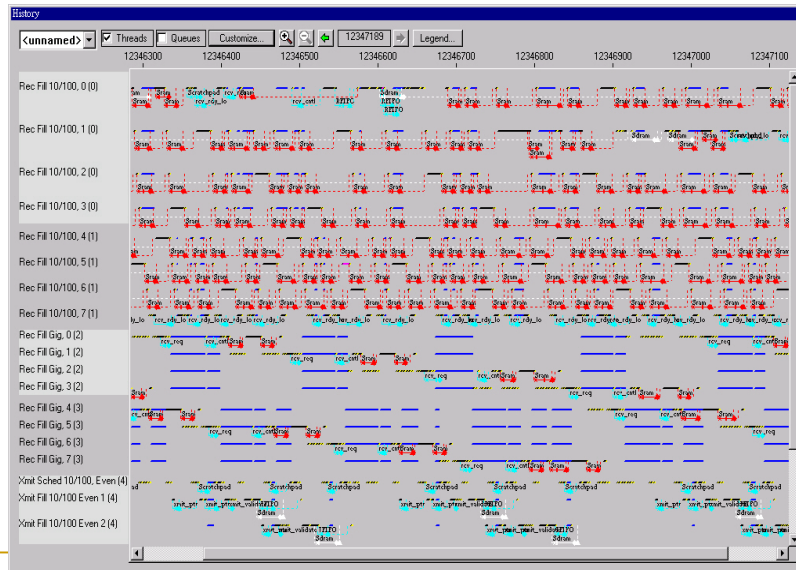
2002/8/22

20

# Performance Statistics– History(RM)



# Performance statistics – History(LN)



## Possible Solutions for Bottlenecks (1/2)

- ❑ SRAM bottleneck
  - ❑ Divide one large SRAM into smaller banks
    - ❑ Each has an interface for accessing a portion of the address space
    - ❑ An arbitrator decides which bank to go
  - ❑ Redundant memory modules
  - ❑ Another memory architecture
    - ❑ QDR SRAM (1.6Gbps, 2~3 times faster than ordinary SRAM)
  - ❑ Additional cache memory for exploiting “locality”

2002/8/22

27

## Possible Solutions for Bottlenecks (2/2)

- ❑ ME bottleneck
  - ❑ Fair, dynamic workload assignment
  - ❑ Hardware upgrade
  - ❑ Programming optimization

2002/8/22

27

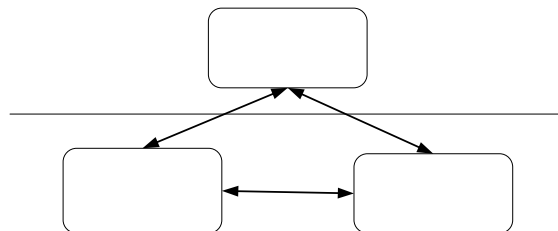
## Conclusion and Future Works (1/2)

- Implement an benchmark error network
- Information bottleneck network
  - Small control file allocation
  - Bottleneck information
  - *Bottleneck shifts*
- Core file
  - Name component
  - Check information for threads

2002/8/22

27

## Conclusions and Future Works (2/2)



**Application-specific** bottleneck: bottleneck may shift!

2002/8/22

27

## References

- [1] IXP1200 Data Sheet, Intel document number 278298-004, May 2000.
- [2] T. Spalink, S. Karlin, L. Peterson, and Y. Gottlieb, "Building a Robust Software-Based Router Using Network Processors," *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP)*.
- [3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," RFC 2475, Dec 1998.
- [4] P. Gupta, and N. McKeown, "Packet Classification on Multiple Fields," *ACM SIGCOMM'99*.
- [5] V. Srinivasan, G. Varghese, S. Suri, and M. Waldvogel, "Fast and Scalable Layer Four Switching," *ACM SIGCOMM'98*.
- [6] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queuing Algorithm," *ACM SIGCOMM'89*.
- [7] T.V. Lakshman, and D. Stiliadis, "High-Speed Policy-based Packet Forwarding Using Efficient Multi-dimensional Range Matching," *ACM SIGCOMM'98*.
- [8] M. Shreedhar, and G. Varghese, "Efficient Fair Queuing Using Deficit Round-Robin," *IEEE/ACM Transactions on Networking*, June 1996, vol. 4, no. 3, pp. 375-385.
- [9] L.V. Nguyen, T. Evers, and J.F. Chicharo, "Differentiated Service Performance Analysis," *Fifth IEEE Symposium on Computers and Communications*, 2000, pp. 328 -333.
- [10] J.K. Muppala, T. Banerjee, and A. Tyagi, "VoIP Performance on Differentiated Services Enabled Network," *IEEE International Conference on Network*, 2000, pp. 419 -423.
- [11] J. Harju, and P. Kivimaki, "Co-operation and Comparison of DiffServ and Intserv: performance measurements," *25th Annual IEEE Conference on Local Computer Networks*, 2000, pp. 177 -186.
- [12] Z. Di, and H.T. Mouftah, "Performance Evaluation of Per-Hop Forwarding Behaviors in the DiffServ Internet," *Fifth IEEE Symposium on Computers and Communications*, 2000, pp. 334-339.