High-Performance Networking
The University of Kansas EECS 881
Network Processors

James P.G. Sterbenz
Department of Electrical Engineering & Computer Science
Information Technology & Telecommunications Research Center
The University of Kansas

jpgs@eeecs.ku.edu

http://www.ittc.ku.edu/~jpgs/courses/hsnets
Network Processors

Outline

NP.1  Overview and Motivation
NP.2  Standard interfaces and system architecture
NP.3  Design alternatives and tradeoffs
NP.4  Example network processors
NP.5  Higher layer and active processing
NP.6  Outlook and future prospects
Network Processors

Outline

NP.1. Overview
NP.2. Standard interfaces
NP.3. Design alternatives
NP.4. Examples
NP.5. Higher layer processing
NP.6. Future prospects
Network Node Principle

Network nodes must support high-bandwidth low-latency, end-to-end flows, as well as their aggregation. High-speed network nodes should provide a scalable number or high-bandwidth, low delay interconnections.
Network Processors

NP.1 Overview and Motivation

NP.2 Standard interfaces and system architecture
NP.3 Design alternatives and tradeoffs
NP.4 Example network processors
NP.5 Higher layer and active processing
NP.6 Outlook and future prospects
Network Processors
Overview and Motivation

• Flexibility of programmability
  – embedded controller software
  – reduce design cycle for vendors
  – field upgradeability and migration for service providers
  
  vs.

• Performance of specialised hardware
  – custom VLSI and ASICS

Middle ground for networking?
Network Processors
Overview and Motivation

• Flexibility of programmability
  – embedded controller software
  – reduce design cycle for vendors
  – field upgradeability and migration for service providers

  *VS.*

• Performance of specialised hardware
  – custom VLSI and ASICS

• Middle ground:
  – network processors: special type of embedded controller
  – programmable hardware: FPGAs
Switch Implementation
Programmable Alternatives

• Alternative implementation technologies
  – fixed hardware (fast but inflexible)
  – programmable hardware (e.g. FPGAs)
  – software: flexible
    • switch control CPU
    • general purpose embedded controller
    • network processor:
      specialised embedded controller for packet processing
Switch Implementation

Programmable Input/Output Processing

- Network processors
  - compromise among cost, flexibility, performance
- Advantages of input/output programmability
  - reduce switch design and debug time
  - allow field upgrades
  - allow service providers to deploy new protocols/services
  - enable per port active networking
Network Processors
Overview and Motivation

• Network processor
  – domain-specific embedded controller
  – designed for high-performance packet processing

• Use for network processors
  – switch and router line cards
  – host-network interfaces \textit{Lecture ES}

• Design goals
  – high-performance packet processing
  – broad applicability
  – limited protocol specificity
Network Processors
Overview and Motivation

- Example NP functionality
  - header parsing and manipulation
  - longest prefix match lookup
  - multidimensional classification
  - error control (CRC/checksum generation and checking)
  - scheduling
    - policing, traffic shaping, fair queueing
Switch Implementation
Programmable Input/Output Processing

• NPs replace
  – input processing
    • lookup
    • classification
  – output processing
    • scheduling
  – additional
    • encryption
    • policing
  – large buffers
    • off NP chip
Network Processors
Overview and Motivation

• Example NPs  *Section NP.4*
  – Intel IXA: IXP1200, 2X00
  – IBM NP4GS3 (now Hifn 5NP4G)
Network Processors

Typical Organisation

- Embedded control processor core + memory
  - performs overall NP and packet dispatching
    - e.g. PowerPC, XScale, ARM
- Packet processing engines
  - typically called \textit{\mu}engines (Intel) or \textit{picoprocessors} (IBM)
  - interconnection between packet \mu engines
    - fixed or flexible; parallel or pipeline
  - packet buffers (shared or per \mu engine)
- Functional units (shared or per \mu engine)
  - hardware assists for critical-path functions
  - e.g. crypto, FEC, CRC, lookup, classification, scheduling
Network Processors

NP.2  Standard Interfaces & Architecture

NP.1  Overview and Motivation
NP.2  Standard interfaces and system architecture
NP.3  Design alternatives and tradeoffs
NP.4  Example network processors
NP.5  Higher layer and active processing
NP.6  Outlook and future prospects
Standard Interfaces

Motivation

- Multiple chip vendors
- Multiple switch and network-interface building blocks
  - network processors
  - search engines and TCAMs
  - classifiers
  - schedulers
  - switch fabric chips
  - queues
  - serdes (serialiser/deserialiser)
  - FIFOs, queues, buffers

Problem?
Standard Interfaces

Motivation

- Multiple chip vendors
- Multiple switch and network-interface building blocks
  - with different internal architectures
  - with different external interfaces
- Standard interfaces permit multivendor solutions
  - common in PC world
  - new to switch and network interface domain
Standard Interfaces

Standardisation Groups

- NPF – Network Processing Forum
  - merged into OIF June 2006
- OIF – Optical Internetworking Forum
  - www.oiforum.com
- Non-profit industry consortium
  - produces *implementation agreements* to...
  - promote interoperability among vendors
Standard Link Interfaces

Physical Interface: SxI

- SxI: common electrical characteristics
  - used by SFI, SPI, etc.
Standard Link Interfaces
Link–System Interface: SPI

- SPI (system packet interface)
  - interface between link framer and system
    - switch fabric or host interconnect
  - separate unidirectional transmit and receive paths
    - 16b data path
    - clock
    - control/data
    - status

- SPI rates
  - SPI-3: OC-48
  - SPI-4: OC-192
  - SPI-5: OC-768
Standard Link Interfaces

Link–System Interface: SPI
Standard Link Interfaces

Link Internal Interfaces: SFI

- **SFI** (serdes (serial-deserial) – framer interface)
  - interface among components in physical/link layer pipeline
  - separate unidirectional transmit and receive paths
    - 16b data path
    - clock, clock reference, deskew
    - status

- **SFI rates**
  - SFI-4: OC-192, OTU-2, 10GbE
  - SFI-5: OC-768, OTU-3, 40GbE
Standard Link Interfaces

Link Internal Interfaces: SFI

- Optical receive
- Data status transmit
- Data transmit [16b]
- Data [2 or 4b]

- Serdes 1:16
- FEC
- Framer SONET OTN Ethernet 802.16

- LA bus
- NPE co-processor
- NPE message
- NPE layer
- NPE [NPE-Framer] SPI
- NPE [NPE-NPE] SPI
- NPE [NPE-Fabric] SPI

- Switch fabric

 Technologies: FEC, 1:16, 802.16, SONET, OTN

© James P.G. Sterbenz
Standard NP Interfaces

NP Interfaces: NPSI and LA

- NPSI (network processor streaming interface)
  - between network processing element (NPE) and...
  - NPSI-framer: ...framing element
    - conformant to SPI
  - NPSI-NPE: ...another network processing element
  - NPSI-fabric: ...switch fabric

- LA (lookaside) interface
  - between NPE and...
  - memory
  - coprocessor

*Details later*
Standard NP Interfaces

NP Interfaces: NPSI and LA

- **Standard NP Interfaces**
- **NP Interfaces:** NPSI and LA
- **Switch Fabric**
- **Framer**
- **SONET, OTN, Ethernet**
- **FEC**
- **Serdes 1:16**
- **Transmission**
- **Receive Data**
- **Status Data**
- **Bus**
- **Co-processor Memory**
Standard NP Interfaces

NP Message Layer

• NP message layer
  – inter-NPE and NPE-coprocessor
  – defines message conveyance and formats
Standard NP Interfaces

NPSI Common Characteristics

- NPSI common characteristics
  - point-to-point connectivity between adjacent devices
  - unidirectional interface (two needed for full duplex)
  - 10 or 40 Gb/s
    - 16-bit data paths
    - 311 MHz or 1.3 GHz double-edge clocking
    - OC-192/10GbEthernet or OC-768 data rates
    - compatible with SPI-4 and SPI-5
  - Status flow control path
Standard NP Interfaces

NPSI: NPE-Framer Characteristics

- NPSI NPE-framer characteristics
  - point-to-point connectivity between NPE and framer
- SPI compliant
  - identical to SPI interface from framer to system
Standard NP Interfaces

NPSI: NPE-Framer Interface

- NPE-Framer Interface (NPSI)
- NPE-Fabric Interface (NPSI)
- NPE-NPE Interface (NPSI)

- Serial deserial (SFI)
- FEC decoder
- Framer
- SONET/OTN/Ethernet
- 802.16
- FEC encoder
- 1:16serializer
- Optical transmit/receive
- Data status [16b]
- Memory co-processor
- LA bus
- NPE message layer
- Switch fabric

© James P.G. Sterbenz
Standard NP Interfaces
NPSI: NPE-Fabric Characteristics

- NPSI NPE-fabric characteristics
  - point-to-point connectivity between NPE and switch fabric
  - 4096 switch fabric ports
  - 256 traffic classes
  - unicast and multicast
  - port and subport multiplexing

- Flow control (data ready indication)
Standard NP Interfaces

NPSI: NPE-Fabric Interface

Switch fabric

NPE

NPSI [NPE-Fabric]

LA bus

NPSI [NPE-NPE]

NPSI [NPE-Framer]

Framer

SONET OTN Ethernet 802.16

FEC

ser des 1:16

optical [1b]

receive transmit

data status [16b] [2 or 4b]

memory co-processor

message

layer

switch fabric

28 October 2010

KU EECS 881 – High-Speed Networking – Net Processors

HSN-NP-31
Standard NP Interfaces
NPSI:  NPE-NPE Characteristics

- NPSI NPE-NPE characteristics
  - point-to-point connectivity between NPEs
- 256 ports with expansion
- Message layer specification
Standard NP Interfaces

NPSI: NPE-NPE Interfaces

- NPSI [NPE-NPE]
- NPSI [NPE-Framer]
- SFI
- LA bus
- switch fabric
- optical [1b]
- transmit
- receive
- data status [16b] [2 or 4b]
- ser des 1:16
- SFI
- FEC
- NPSI
- framer
- SONET OTN Ethernet 802.16
- SPI
- NPE
- message
- co-processor
- memory
- NPE
- NPE layer
- NPE
- NPE-Fabric
- NPSI
Standard NP Interfaces
LA: Lookaside Characteristics

- LA (lookaside) bus characteristics
  - multidrop (≤500Mhz) or cascade (<300MHz) bus
    - master NPE
    - slave memory, lookup, classification, or other coprocessor
  - 10 or 40 Gb/s
    - 2×16-bit data + 2-bit parity data paths
      - concurrent read/write
    - up to 29-bit address lines
    - read, write (word, high- and low-byte) byte select lines
    - 3 clock lines
      - 500MHz double-edge clocking
Standard NP Interfaces

LA: Lookaside Interfaces

- Optical receive [1b]
- Optical transmit
- Data status [16b]
- [2 or 4b]

- 1:16 ser des
- FEC
- Framer
- SONET OTN Ethernet 802.16

- NPE co-processor
- Memory

- LA bus

- NPE framer
- SONET OTN Ethernet 802.16

- Switch fabric

- NPSI [NPE-Framer] SPI
- NPSI [NPE-NPE]
- NPSI [NPE-Fabric]
Network Processors

NP.3 Design Alternatives and Tradeoffs

NP.1 Overview and Motivation
NP.2 Standard interfaces and system architecture
NP.3 Design alternatives and tradeoffs
NP.4 Example network processors
NP.5 Higher layer and active processing
NP.6 Outlook and future prospects
NP Design Alternatives & Tradeoffs
Degree of Specialisation

- Set of relatively general purpose processors
  - e.g. Intel IXP
- Highly specialised functional units
  - checksum, lookup, classification, crypto etc.
  - in addition to general purpose packet processors
  - e.g. IBM NP4GS3

*Tradeoffs?*
NP Design Alternatives & Tradeoffs
Degree of Specialisation

• Set of relatively general purpose processors
  – e.g. Intel IXP

• Highly specialised functional units
  – checksum, lookup, classification, crypto etc.
  – in addition to general purpose packet processors
  – e.g. IBM NP4GS3

• Tradeoffs
  – ease of programming vs. performance
NP Design Alternatives & Tradeoffs

Scalability: Performance Measures

Performance measures?
NP Design Alternatives & Tradeoffs

Scalability: Performance Measures

- Performance measures
  - "line speed" or data rate
    - typically in SONET equivalents: 2.5, 10, 40 Gbps
  - packet processing rate in [packets/sec]
    - maximum necessary derived from rate and 40B packets
  - latency
    - generally not an issue except for very reach networks
NP Design Alternatives & Tradeoffs

NP Scalability: Techniques

Scaling of a single NP?
NP Design Alternatives & Tradeoffs

NP Scalability: Techniques

- Scaling of a single NP
  - higher clock rate
  - functional units

more?
NP Design Alternatives & Tradeoffs

NP Scalability: Techniques

- Scaling of a single NP
  - higher clock rate
  - functional units
    - greater number
    - more specialised: lookup, classification, scheduling, crypto, ...

additional techniques?
**NP Design Alternatives & Tradeoffs**

**NP Scalability: Techniques**

- Scaling of a single NP
  - higher clock rate
  - functional units
    - greater number
    - more specialised: lookup, classification, scheduling, crypto, ...
  - packet pipeline
    - each μengine stage operates on all packets
    - each μengine stage performs a fraction of packet processing
NP Design Alternatives & Tradeoffs

NP Scalability: Techniques

- Scaling of a single NP
  - higher clock rate
  - functional units
    - greater number
    - more specialised: lookup, classification, scheduling, crypto, ...
- packet pipeline
  - each µengine stage operates on all packets
  - each µengine stage performs a fraction of packet processing
- packet parallelism
  - each µengine operates on fraction of packets
  - each µengine performs processing functions
NP Design Alternatives & Tradeoffs

NP Scalability: Techniques

- Scaling of a single NP
  - higher clock rate
  - functional units
    - greater number
    - more specialised: lookup, classification, scheduling, crypto, ...
  - packet pipeline
    - each µengine stage operates on all packets
    - each µengine stage performs a fraction of packet processing
  - packet parallelism
    - each µengine operates on fraction of packets
    - each µengine performs processing functions
  - combination of techniques
NP Design Alternatives & Tradeoffs
NP Scalability: Techniques

Scaling performance with multiple NPs?
NP Design Alternatives & Tradeoffs

NP Scalability: Techniques

• Scaling performance with multiple NPs
  – specialised coprocessors
    • lookup
    • classification
    • scheduling
    • crypto
    • ...

28 October 2010
NP Design Alternatives & Tradeoffs

NP Scalability: Techniques

- Scaling performance with multiple NPs
  - specialised coprocessors
    - lookup, classification
    - scheduling
    - crypto, ...
  - packet pipeline
    - each NP stage operates on all packets
    - each NP stage performs a fraction of packet processing
NP Design Alternatives & Tradeoffs

NP Scalability: Techniques

- Scaling performance with multiple NPs
  - specialised coprocessors
    - lookup, classification
    - scheduling
    - crypto, ...
  - packet pipeline
    - each NP stage operates on all packets
    - each NP stage performs a fraction of packet processing
  - packet parallelism
    - each NP operates on fraction of packets
    - each NP performs processing functions
NP Design Alternatives & Tradeoffs

NP Scalability: Techniques

• Scaling performance with multiple NPs
  – specialised coprocessors
    • lookup, classification
    • scheduling
    • crypto, ...
  – packet pipeline
    • each NP stage operates on all packets
    • each NP stage performs a fraction of packet processing
  – packet parallelism
    • each NP operates on fraction of packets
    • each NP performs processing functions
  – combination of techniques
Network Processors

NP.4  Example Network Processors

NP.1  Overview and Motivation
NP.2  Standard interfaces and system architecture
NP.3  Design alternatives and tradeoffs
NP.4  Example network processors
NP.5  Higher layer and active processing
NP.6  Outlook and future prospects
## Network Processors

### Examples

<table>
<thead>
<tr>
<th></th>
<th>Interface Capacity</th>
<th>Packet Processors</th>
<th>Co-processors</th>
<th>Control Processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IXP 1200</td>
<td>OC-12</td>
<td>6 @ 166 MHz</td>
<td>hash</td>
<td>StrongARM 232 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4K × 40b I mem</td>
<td></td>
<td>8MB + 256MB+ external</td>
</tr>
<tr>
<td>IXP 2400</td>
<td>OC-48 14Mpps 40B</td>
<td>8 @ 600 MHz</td>
<td>hash</td>
<td>Xscale 600MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4K × 40b I mem</td>
<td></td>
<td>32KI$ [32+2]KD$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pipeline interconnection</td>
<td></td>
<td>2GB+32MB external</td>
</tr>
<tr>
<td>IXP 2850</td>
<td>OC-192 60Mpps 40B</td>
<td>16 @ 1.4 GHz</td>
<td>hash</td>
<td>Xscale 700MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4K×40b I+996w D mem</td>
<td></td>
<td>32KI$ [32+2]D$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pipeline interconnection</td>
<td></td>
<td>Ext 6GB+64MB ext</td>
</tr>
<tr>
<td>C-Port Motorola C-5</td>
<td>4 × OC-12 15Mpps</td>
<td>16 @ 266 MHz</td>
<td>queue, buffer,</td>
<td>XP 266 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>64KI + 12KD mem</td>
<td>fabric, table lookup</td>
<td>32KI + 32KD mem</td>
</tr>
<tr>
<td>IBM PowerNP 4GS3</td>
<td>OC-48 4.5Mpps</td>
<td>8 × 2 @ 133 MHz</td>
<td>8×10 e.g. tree search, checksum</td>
<td>PowerPC 405 133 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>picoprocessors</td>
<td></td>
<td>16KI$ + 16KD$</td>
</tr>
</tbody>
</table>

others are proprietary and require NDA for details
Network Processors
Example: Intel IXP 2800/2850

- Moderately specialised embedded controller
  - organised for packet processing
  - XScale ARM core controller + 16 microengines
  - some functional units
    - hash, crypto (AES, DES, SHA-1 on 2850)
    - CRC, pseudorandom generator per microengine

- Programming
  - requires *significant* knowledge of hardware organisation
Network Processors
Example: Intel IXP 2800/2850

- Processors
  - XScale core @ 700MHz
  - 16 microengines
- 16KB scratchpad mem
- Controllers
  - PCI
  - SRAM, DRAM
- Media/switch interface
- Hash unit
- Crypto (2850 only)
Network Processors
Example: Intel IXP 2800/2850 Microengine

- 16 @ 1.4 GHz
- 2 × 128 GPRs (A,B)
- ALU (A • B)
- 640 longword local mem
- {S|D}RAM regs
- 16 × 32+4b CAM
- CRC-{16,32} unit
Network Processors
Example: C-Port Motorola C-5

• Moderately specialised embedded controller
  – organised for packet processing
  – XP (executive processor) MIPS-1 core
  – 16 CPs (channel processors) MIPS-1
  – some functional units
    • table lookup and classification engine
    • fabric processor and interface
    • queue management unit interface to 128 Kword SRAM
    • buffer management unit interface to 128 MB SDRAM
    • external interface to traffic management coprocessor

• Programming
  – requires *significant* knowledge of hardware organisation
Network Processors
Example: C-Port Motorola C-5

- **Processors**
  - XP MIPS-1
    - 266 MHz
  - 16 CPs
    - 266 MHz

- **Coprocessors**

- **Interfaces**
  - PCI
  - S{D}RAM
  - fabric
Network Processors
Example: IBM PowerNP 4GS3

• Highly specialised embedded controller
  – organised for packet processing; many functional units
    • PowerPC 405 core @ 133 MHz
    • completion unit, dispatch unit, control store arbiter
      enqueue/dequeue unit, traffic scheduler/shaper
    • 8 DPPUs (dyadic protocol processor units)
      – 2 picoprocessors each @ 133 MHz
      – 10 coprocessors: checksum, bus control, ext coprocessor control,
        counter, data store, enqueue, policer, string copy, tree search,
        semaphore manager

• Programming
  – requires significant knowledge of hardware organisation

IBM has sold the NP to Hifn – datasheets and manuals no longer online
Network Processors

Example: IBM PowerNP 4GS3

**Processors**
- PowerPC @ 133 MHz
  - 16K I + 16K D cache
  - 8 DPPUs

**Memory**
- 32 KB instruction
- 128 KB SRAM buffer
- 113 KB SRAM ctl store

**Interface controllers**
- switch ingress+egress
- SRAM, SDRAM
- PCI

**Coprocessors/accelerators**
- traffic scheduler/shaper
- enqueue/dequeue
- completion, dispatch, ctl store
Network Processors
Example: IBM PowerNP 4GS3 EPC

- 8 DPPUs
  - 2 picoprocessors
    - 133 MHz
    - 16|32 GPRs
    - ALU
  - 10 coprocessors
- $8 \times 4$KB memory
Network Processors
Example: IBM PowerNP 4GS3 DPPU
Network Processors

NP.5 Higher Layer and Active Processing

NP.1 Overview and Motivation
NP.2 Standard interfaces and system architecture
NP.3 Design alternatives and tradeoffs
NP.4 Example network processors
NP.5 Higher layer and active processing
NP.6 Outlook and future prospects
Higher Layer and Active Processing
Motivation and Strategies

• Example motivation for processing in network nodes
  – firewalls  \textit{Lecture IS}
  – Web caches  \textit{Lecture AL}
  – active transcoders
  – multimedia stream mixing and merging

\textit{Strategies?}
Higher Layer and Active Processing
Motivation and Strategies

• Example motivation for processing in network nodes
  – firewalls \textit{Lecture IS}
  – Web caches \textit{Lecture AL}
  – active transcoders
  – multimedia stream mixing and merging

• Alternative strategies
  – application layer proxies
  – layer 4 and layer 7 switches
  – integration of function into layer 3 forwarding code
  – programmable active network nodes
Higher Layer and Active Processing
Programmable Networking

- Programmable network nodes
  - switch or router programmable
    - code for control processor
    - code in network processors on line cards
  - part of provisioning process
Higher Layer and Active Processing

Active Networking

- Programmable network nodes
  - switch or router programmable
    - code for control processor
    - code in network processors on line cards
  - part of provisioning process
- Active network nodes
  - nodes dynamically programmable
  - capsules (smart packets) carry mobile code
  - permits global or per flow deployment of services
  - code acts on conventional packets in flow
    - control plane or data plane
Higher Layer and Active Processing
DARPA Active Network Reference Model

- Reference model for active programmable nodes
  - developed by DARPA active nets program
- AA: active application carried in capsule
- EE: execution environment in which AA runs
  - MEE: management EE runs mgt. AAs to control node
- NodeOS: environment in which EEs operate
  - resource management and isolation
Higher Layer and Active Processing
DARPA Active Network Reference Model

NodeOS

MEE

MAAs

AAs

EEs

AAs

AAs

link MAC

switch fabric

physical receive

link MAC

physical transmit
Higher Layer and Active Processing
DARPA Active Network Reference Model

- Reference model for active programmable nodes
  - developed by DARPA active nets program
- AA: active application carried in capsule
- EE: execution environment in which AA runs
  - MEE: management EE runs mgt. AAs to control node
- NodeOS: environment in which EEs operate
  - resource management and isolation

Problem?
Higher Layer and Active Processing
DARPA Active Network Reference Model

- Reference model for active programmable nodes
  - developed by DARPA active nets program
- AA: active application carried in capsule
- EE: execution environment in which AA runs
  - MEE: management EE runs mgt. AAs to control node
- NodeOS: environment in which EEs operate
  - resource management and isolation
- Problem
  - high-performance switches not general purpose machines
  - critical path implemented on line cards
Higher Layer and Active Processing

Performance

- Active processing reasonable tradeoff
  - performance vs. flexibility
- Active processing must
  - not impede the normal fast forwarding path
  - occur with enough throughput so active queues don’t build

Active Network Processing

Active network processing should not impede the non-active fastpath; packet filters in the critical path must operate at line rate to pass non-active packets. The ability to perform per packet active processing requires sufficient processing power to sustain the require active packet throughput
Active Processing
Granularity and Node Architecture

- Processing granularity
  - global control plane
    - e.g. routing algorithm
  - per flow control plane
    - e.g. modify RSVP control
  - per packet control plane
    - e.g. congestion control
  - per packet data plane
    - e.g. transcoding
Active Processing
Network Processors

- Network processors
  - enable per port dynamic programmability
Network Processors

NP.6  Outlook and Future Prospects

NP.1  Overview and Motivation
NP.2  Standard interfaces and system architecture
NP.3  Design alternatives and tradeoffs
NP.4  Example network processors
NP.5  Higher layer and active processing
NP.6  Outlook and future prospects
Network Processors

Future Prospects

• Programmable network prospects
  – tied directly to deployment of open COTS NPs
Network Processors
Future Prospects

• Programmable network prospects
  – tied directly to deployment of open COTS NPs
• Future prospects of open COTS NPs in doubt
  – IBM exited market
  – Intel future in market uncertain
  – high end switch market dominated by single vendor
    • Cisco (+ Juniper)
Network Processors

Further Reading

• Comer,
  *Network Systems Design using Network Processors: Intel IXP 2xxx Version*
  http://ww.npbook.cs.purdue.edu

• Lekkas,
  *Network Processors: Architectures, Protocols, and Platforms*
  McGraw-Hill

• Carlson,
  *Intel Internet Exchange Architecture and Applications: A Practical Guide to IXP2XXX Network Processors*
  http://www.intel.com/intelpress/sum_ixa.htm
Network Processors

Acknowledgements

Some material in these foils comes from the textbook supplementary materials: