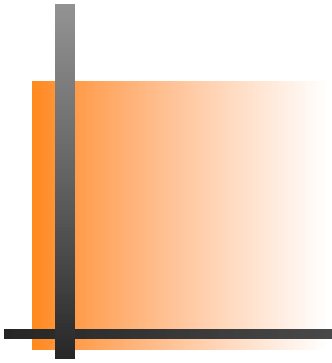
INF5063:
Programming Asymmetric Multi-Core Processors



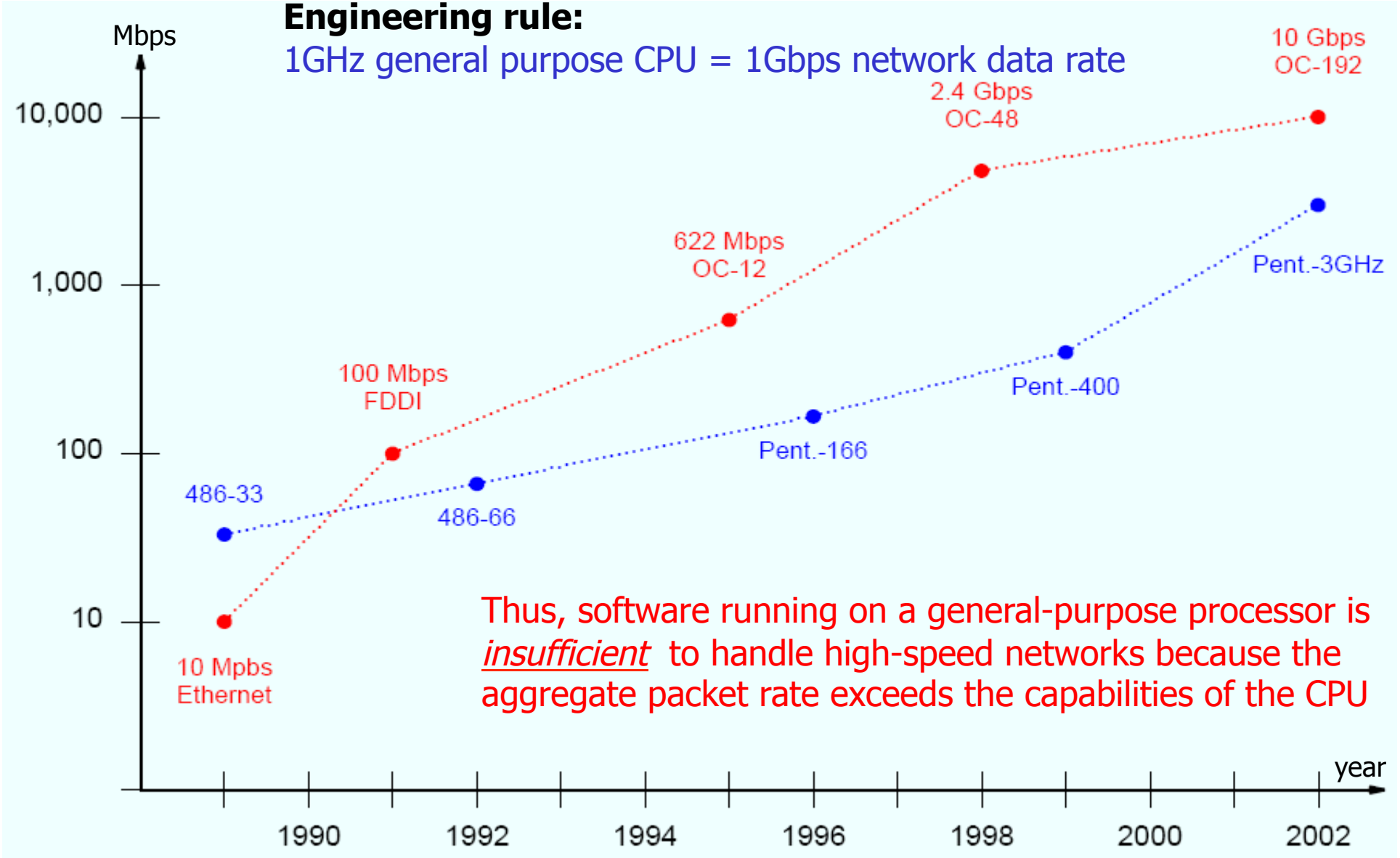
IXP: Bump in the Wire

September 13, 2010

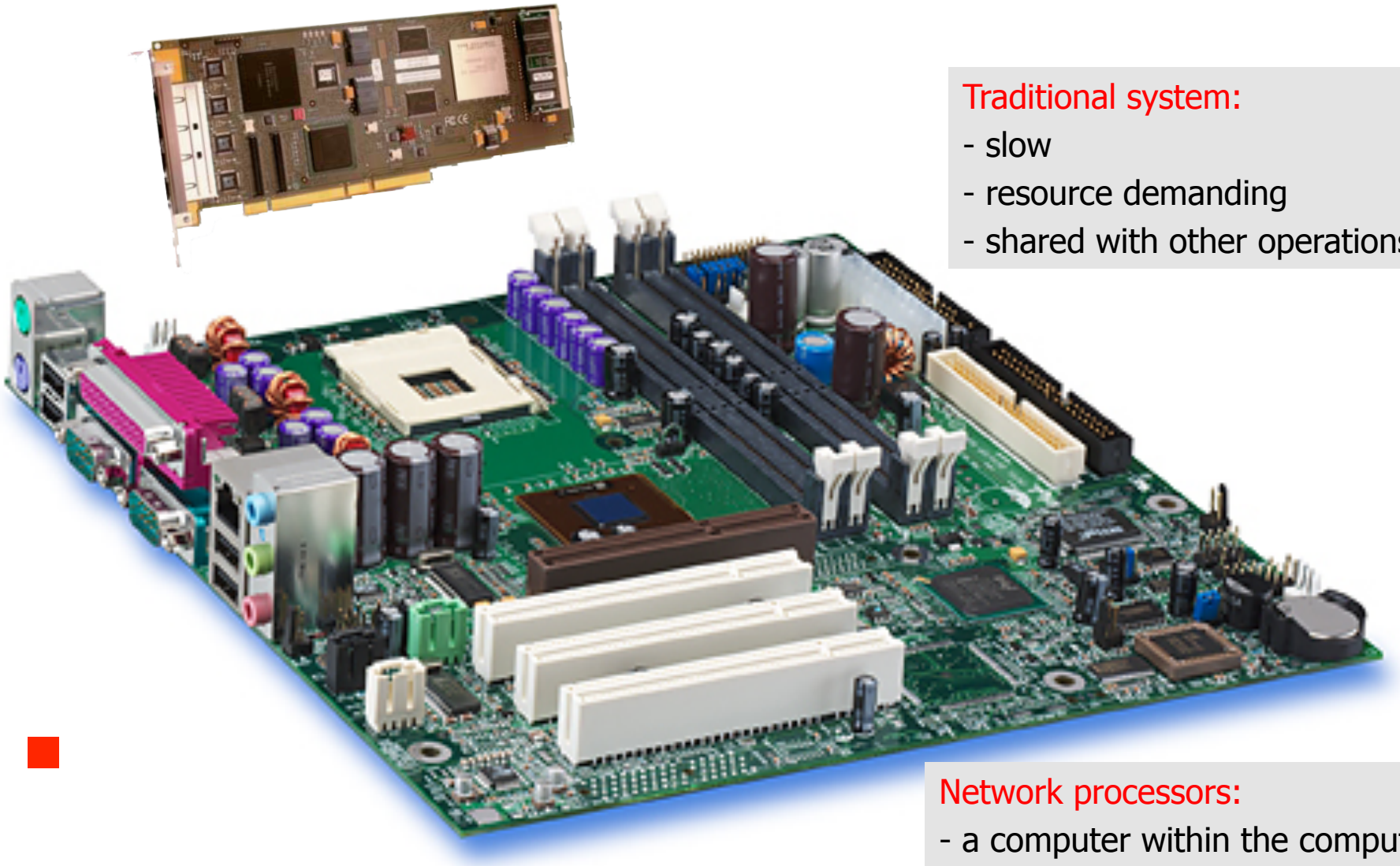
Question:

- Which is growing faster?
  - network bandwidth
  - processing power
- Note: if network bandwidth is growing faster
  - CPU may be the bottleneck
  - need special-purpose hardware
- Note: if processing power is growing faster
  - no problems with processing
  - network/busses will be bottlenecks

Growth Of Technologies



Network Processors: Main Idea



Traditional system:
- slow
- resource demanding
- shared with other operations

Network processors:
- a computer within the computer
- special, programmable hardware
- offloads host resources

Explosion of Commercial Products

- 1990 → 2000: network processors transformed from interesting curiosity to mainstream product
  - reduction in both overall costs and time to market
  - 2002: over 30 vendors with a vide range of architectures
  - e.g.,
    - Multi-Chip Pipeline (Agere)
    - Augmented RISC Processor (Alchemy)
    - Embedded Processor Plus Coprocessors (Applied Micro Circuit Corporation)
    - Pipeline of Homogeneous Processors (Cisco)
    - Pipeline of Heterogeneous Processors (EZchip)
    - Configurable Instruction Set Processors (Cognigine)
    - Extensive And Diverse Processors (IBM)
    - Flexible RISC Plus Coprocessors (Motorola)
    - Internet Exchange Processor (Intel)
    - …

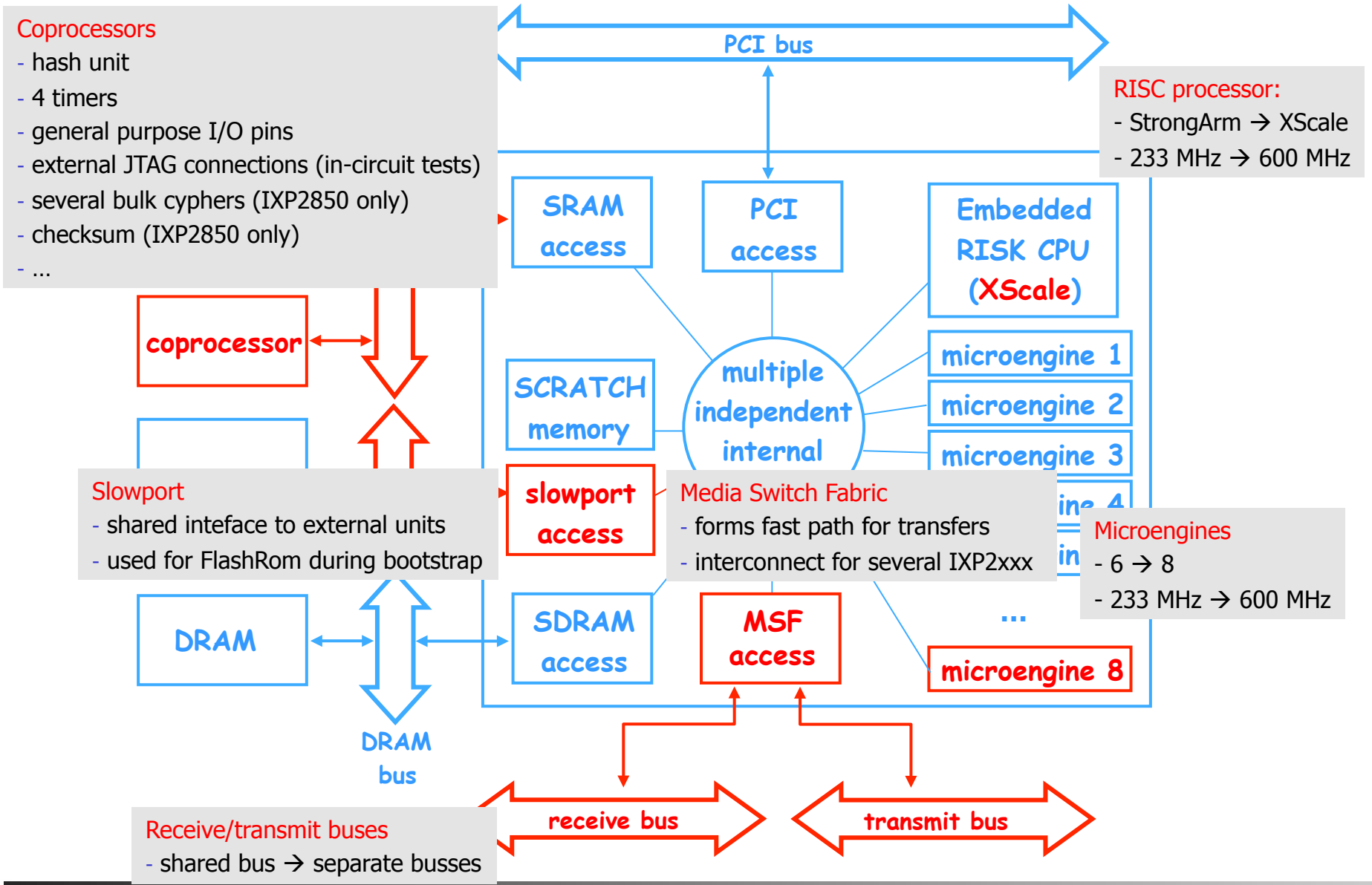
Intel IXP1200 / 2400: A Short Overview

IXA: Internet Exchange Architecture

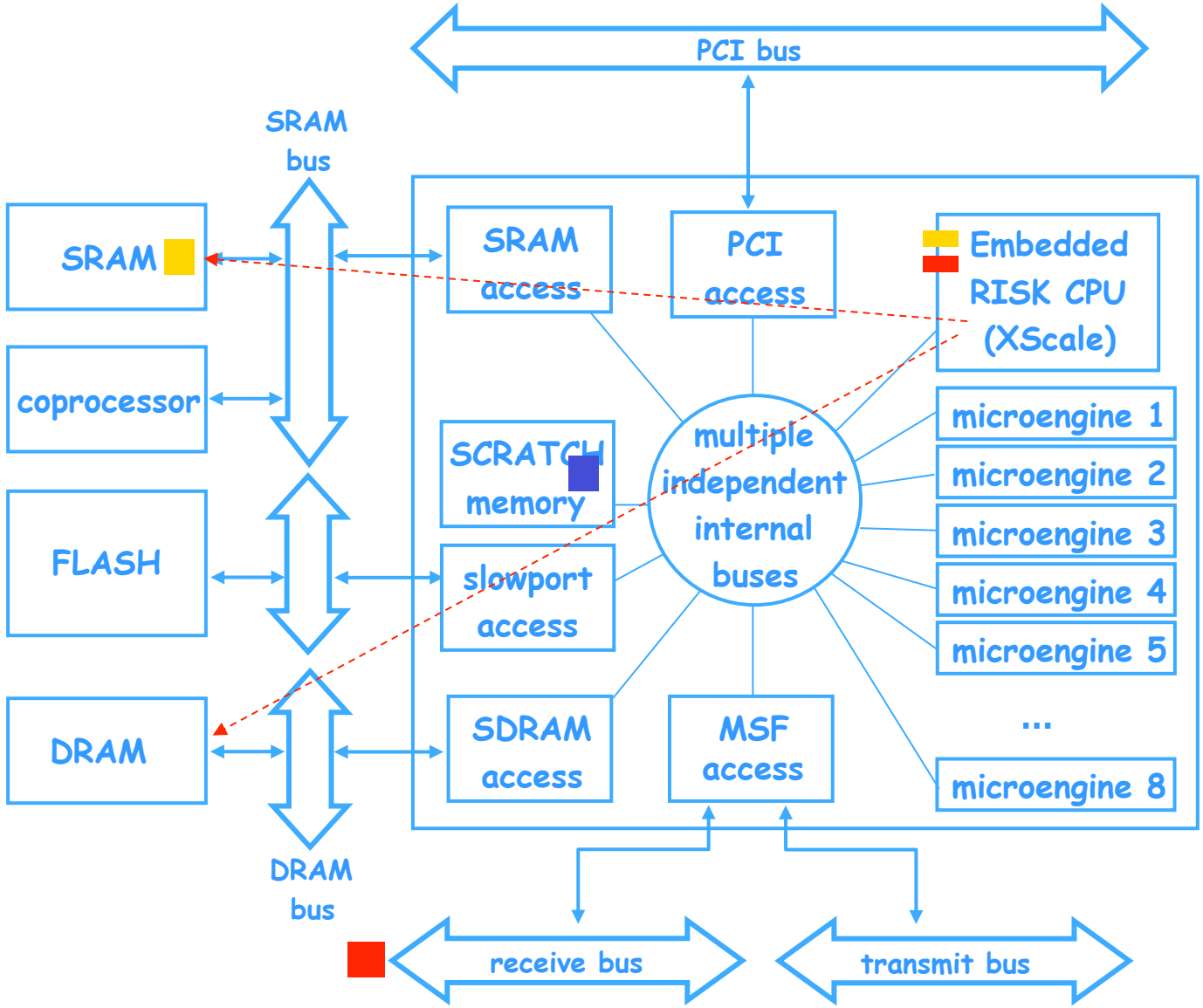
- IXA is a broad term to describe the Intel network architecture (HW & SW, control- & data plane)
- IXP: Internet Exchange Processor
  - processor that implements IXA
  - IXP1200 is the first IXP chip (4 versions)
  - IXP2xxx has now replaced the first version

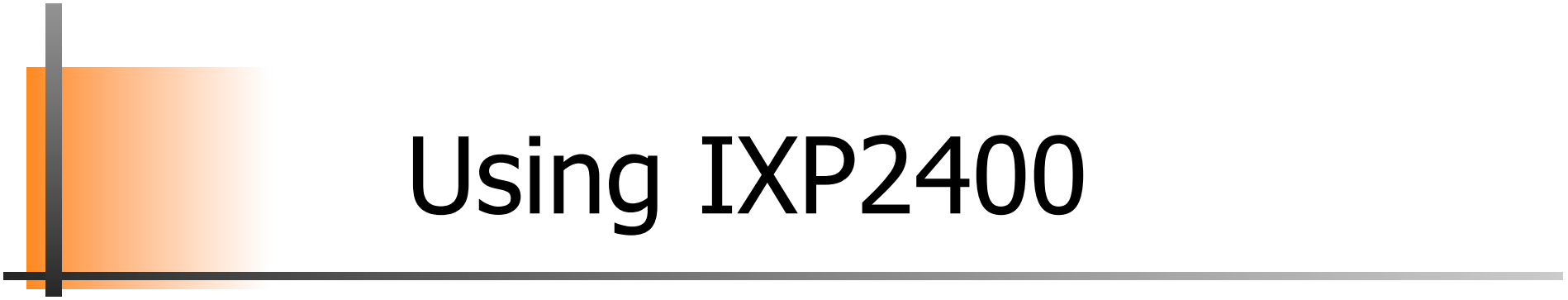
- **IXP1200** basic features
  - 1 embedded 232 MHz StrongARM
  - 6 packet 232 MHz *µengines*
  - onboard memory
  - 4 x 100 Mbps Ethernet ports
  - multiple, independent busses
  - low-speed serial interface
  - interfaces for external memory and I/O busses
  - …
- **IXP2400** basic features
  - 1 embedded 600 MHz XScale
  - 8 packet 600 MHz *µengines*
  - 3 x 1 Gbps Ethernet ports
  - …

IXP2400 Architecture



IXP2400 Basic Packet Processing



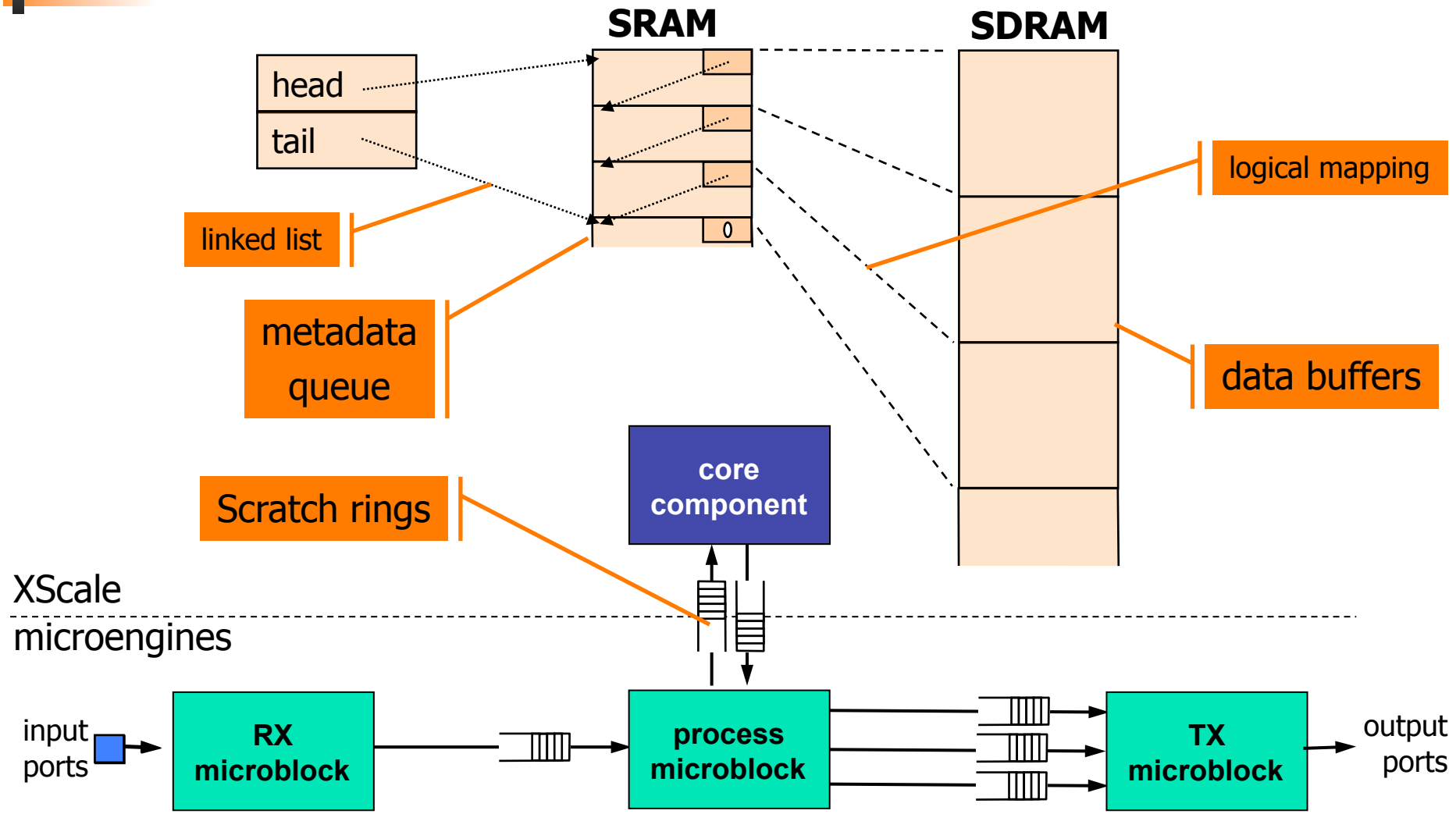


Using IXP2400

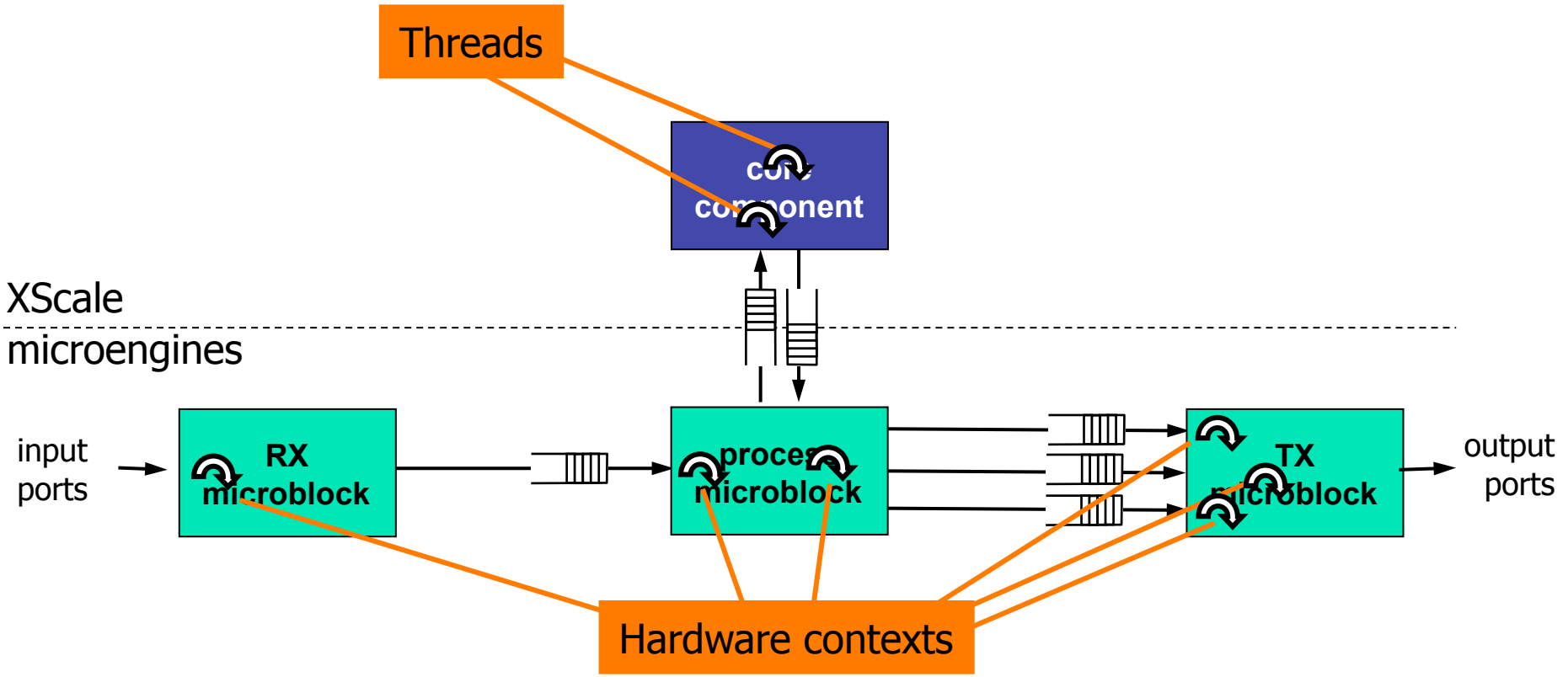
Intel IXP2400 Hardware Reference Manual

This page intentionally left blank.

Programming Model



Programming Model

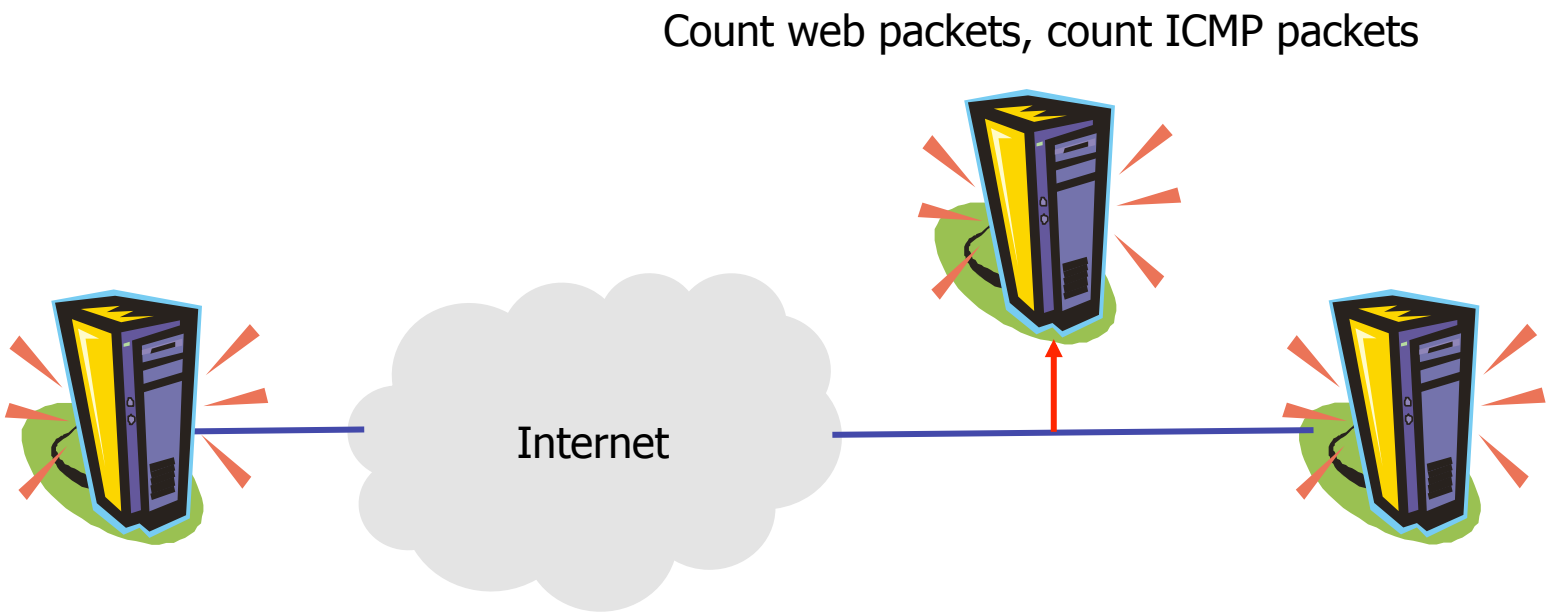


Framework

- uclo
  - Microengine loader
  - Necessary to load your microengine code into the microengines at runtime
- hal
  - Hardware abstraction layer
  - Mapping of physical memory into XScale processes' virtual address space
  - Functions starting with hal
- ossl
  - Operating system service layer
  - Limited abstraction from hardware specifics
  - Functions starting with ix_
- rm
  - Resource manager
  - Layered on top of uclo and ossl
  - Memory and resource management
    - all memory types and their features
    - IPC, counters, hash
  - Functions starting with ix_

Bump in the Wire

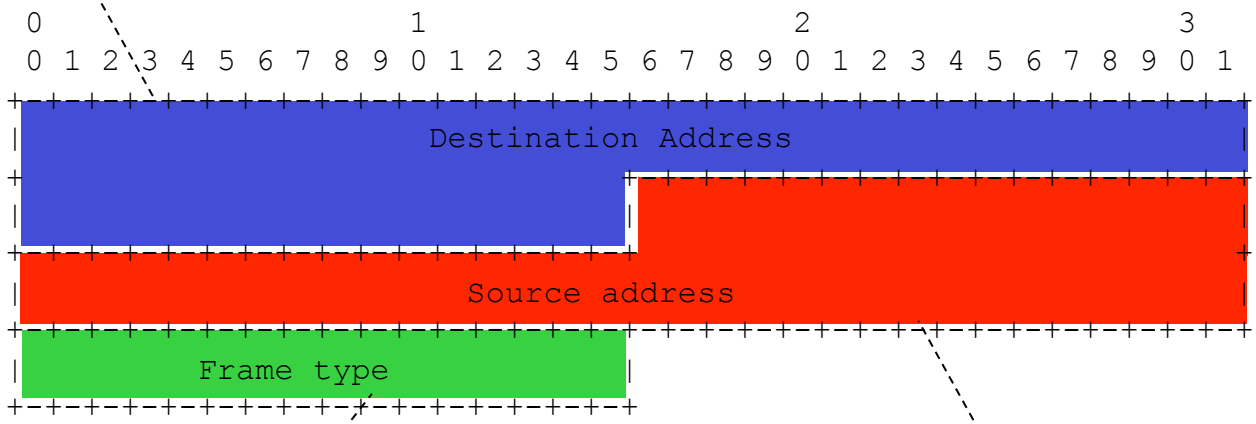
Bump in the Wire



Packet Headers and Encapsulation

Ethernet

48 bit address configured to an interface on the NIC on the receiver



describes content of ethernet frame, e.g., 0x0800 indicates an IP datagram, 0x0806 indicates an ARP packet

48 bit address configured to an interface on the NIC on the sender

Internet Protocol version 4 (IPv4)

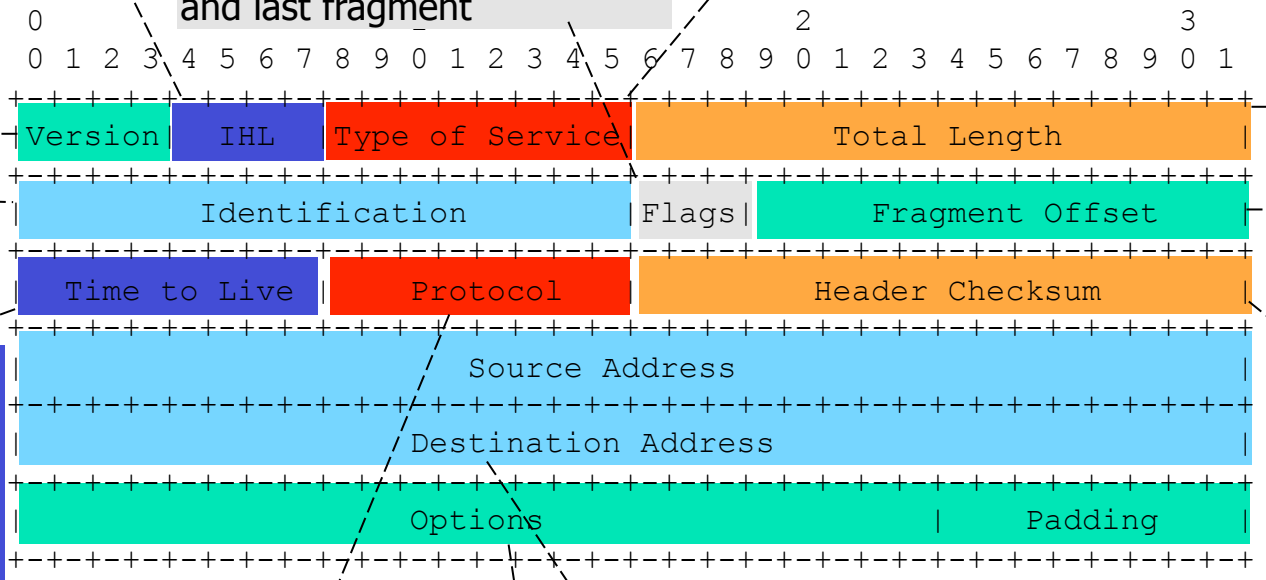
indicates the format of the internet header, i.e., version 4

length of the internet header in 32 bit words, and thus points to the beginning of the data (minimum value of 5)

indication of the abstract parameters of the **quality of service** desired – somehow treat high precedence traffic as more important – tradeoff between low-delay, high-reliability, and high-throughput – NOT used, bits now reused for differential services code point

first zero, fragments allowed and last fragment

datagram length (octets) including header and data - allows the length of 65,535 octets



identifying value to aid assembly of fragments

indicate where this fragment belongs in datagram

disable a packet to circulate forever,decrease value by at least 1 in each node – discarded if 0

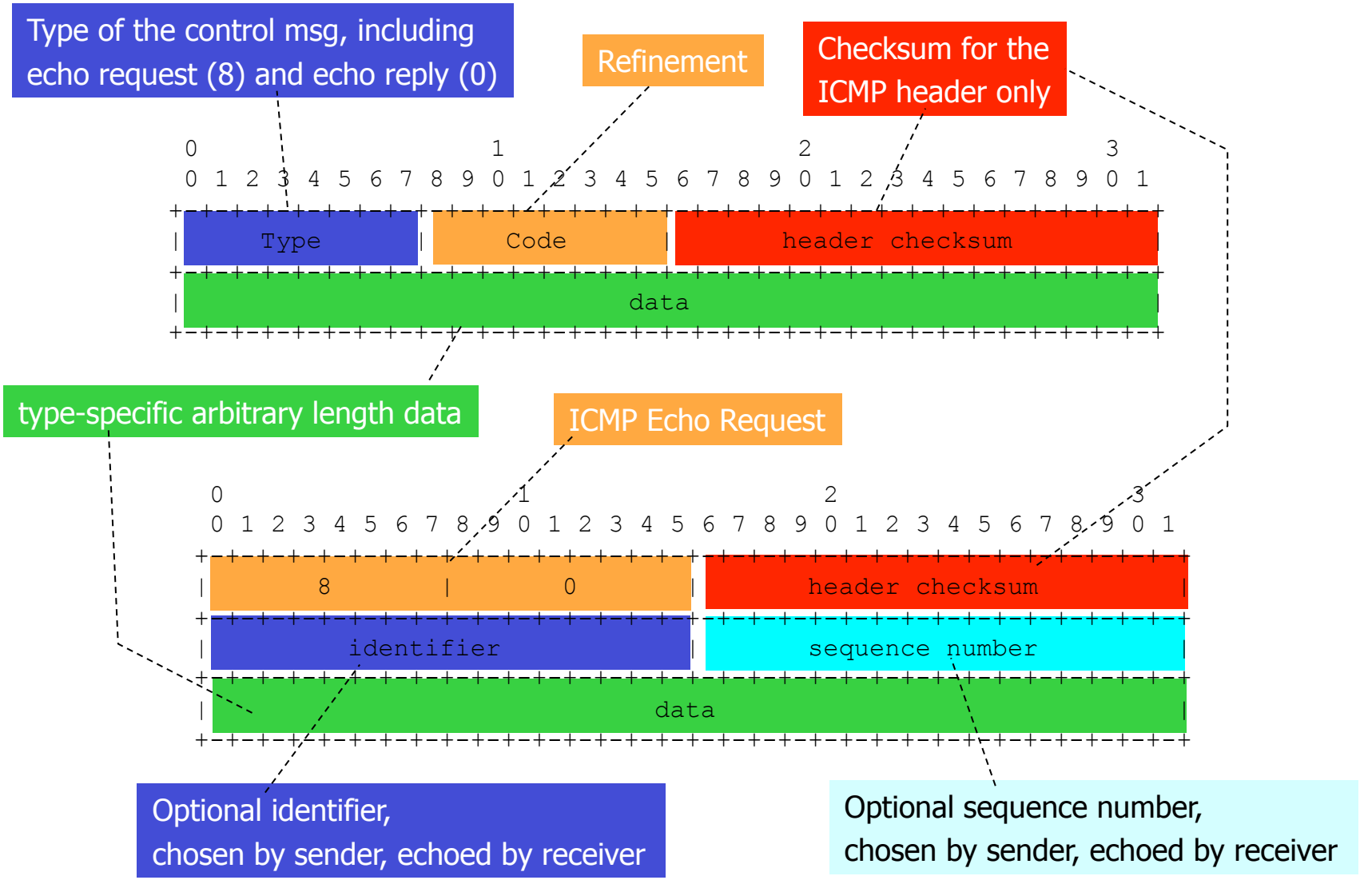
checksum on the header only – TCP, UDP over payload. Since some header fields change(TTL), this is recomputed and verified at each point

indicates used transport layer protocol

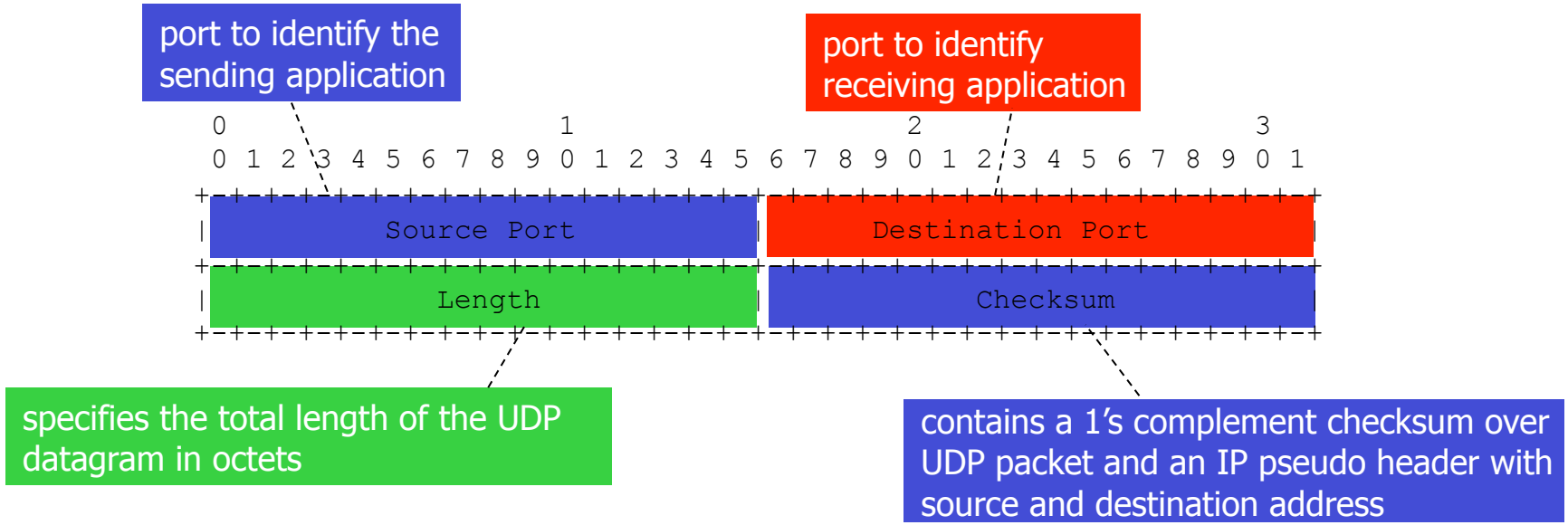
32-bit address fields. May be configured differently from small to large networks

options may extend the header – indicated by IHL. If the options do not end on a 32-bit boundary, the remaining fields are padded in the padding field (0's)

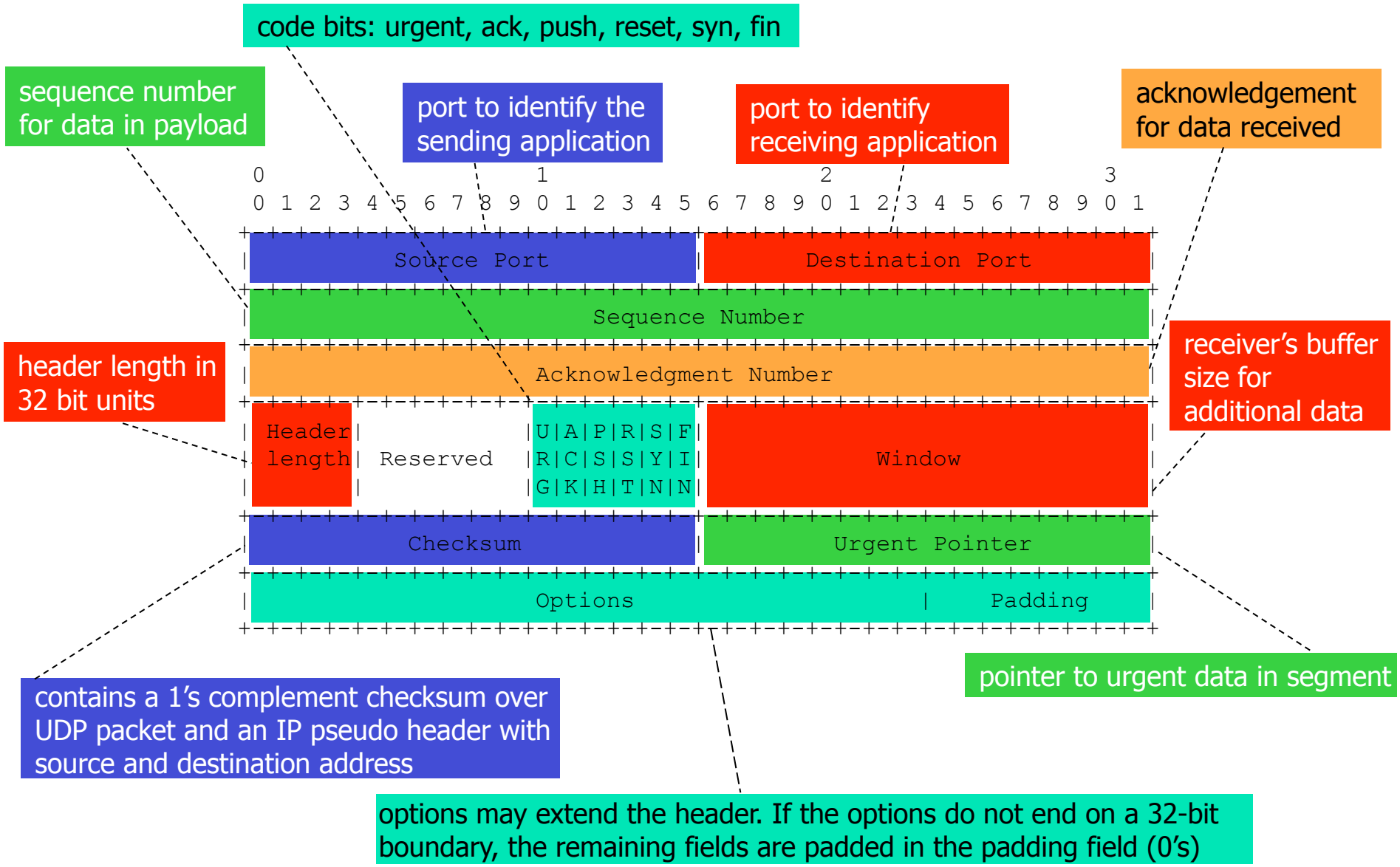
Internet Control Message Protocol (ICMPv4)



UDP

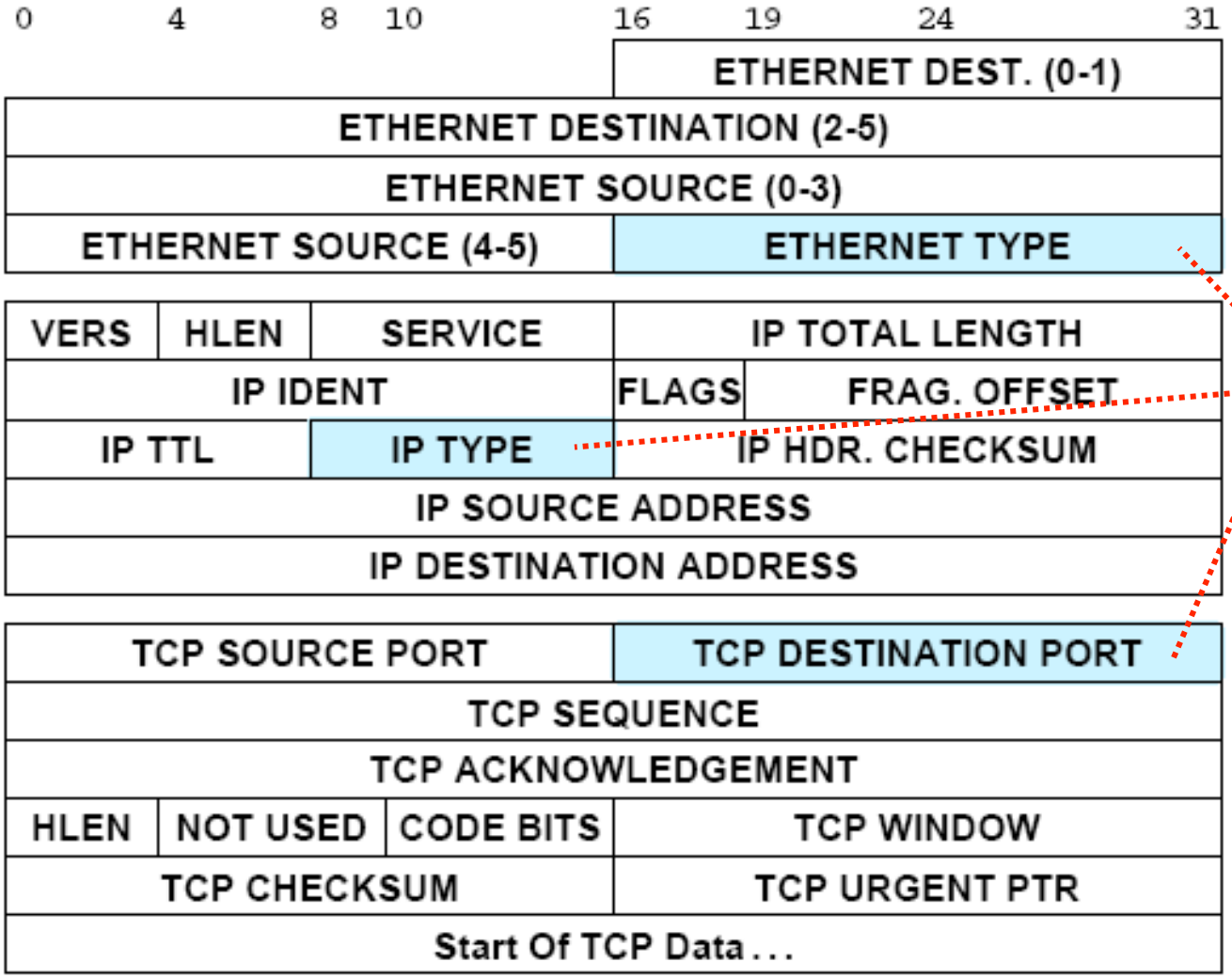

port to identify the sending application
port to identify receiving application
0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Source Port
Destination Port
Length
Checksum
specifies the total length of the UDP datagram in octets
contains a 1's complement checksum over UDP packet and an IP pseudo header with source and destination address

TCP


code bits: urgent, ack, push, reset, syn, fin
sequence number for data in payload
port to identify the sending application
port to identify receiving application
acknowledgement for data received
0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Source Port
Destination Port
Sequence Number
Acknowledgment Number
header length in 32 bit units
receiver's buffer size for additional data
Header length
Reserved
U|A|P|R|S|F
R|C|S|S|Y|I
G|K|H|T|N|N
Window
Checksum
Urgent Pointer
Options
Padding
contains a 1's complement checksum over UDP packet and an IP pseudo header with source and destination address
pointer to urgent data in segment
options may extend the header. If the options do not end on a 32-bit boundary, the remaining fields are padded in the padding field (0's)

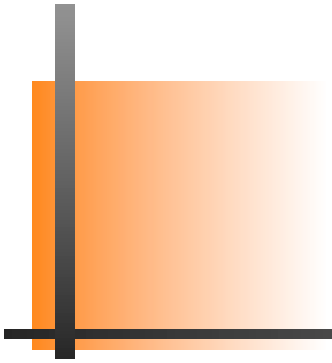
Encapsulation

Identifying Web Packets



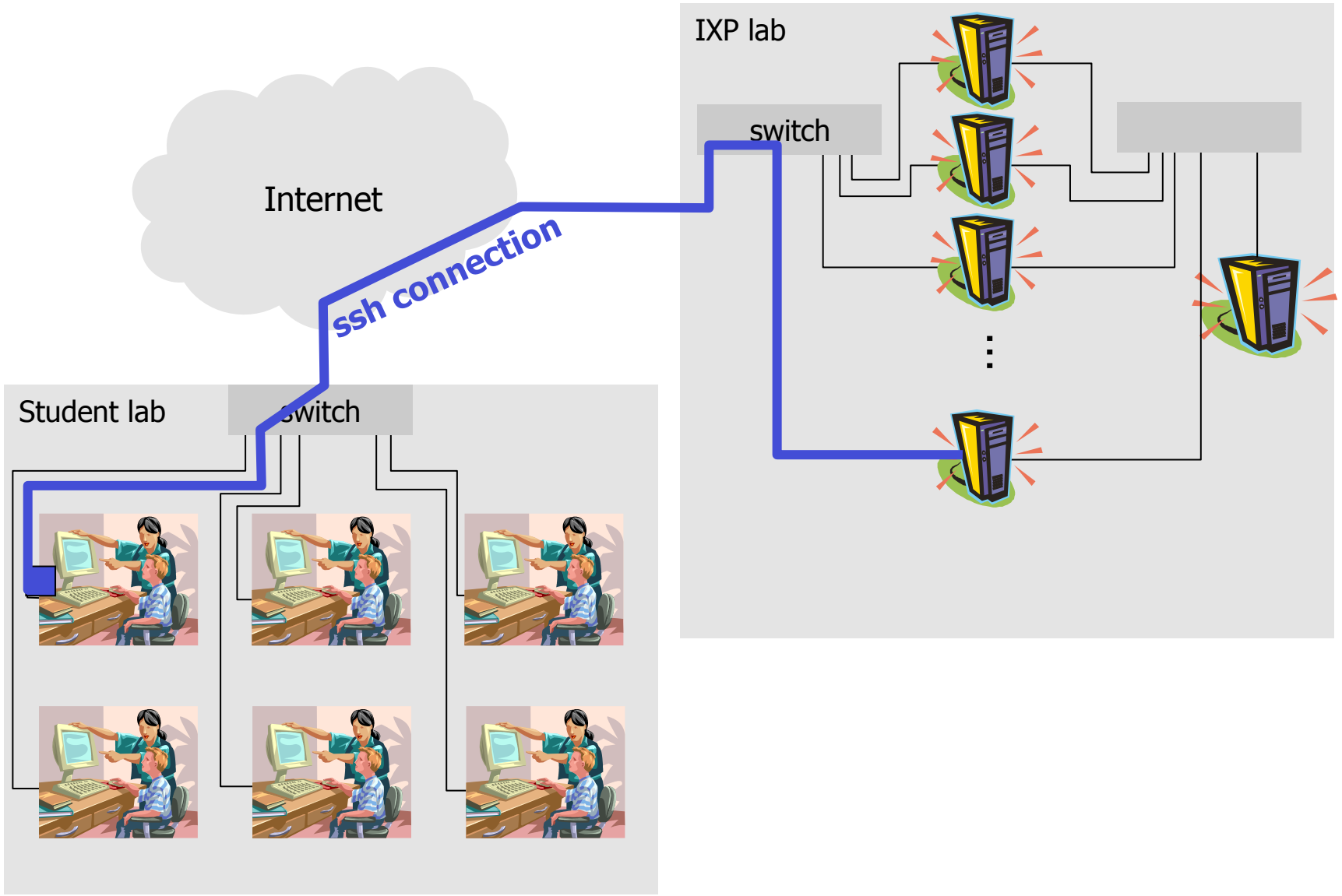
These are the header fields you need for the web bumper:

- ✓ Ethernet type **0x800**
- ✓ IP type **6**
- ✓ TCP port **80**

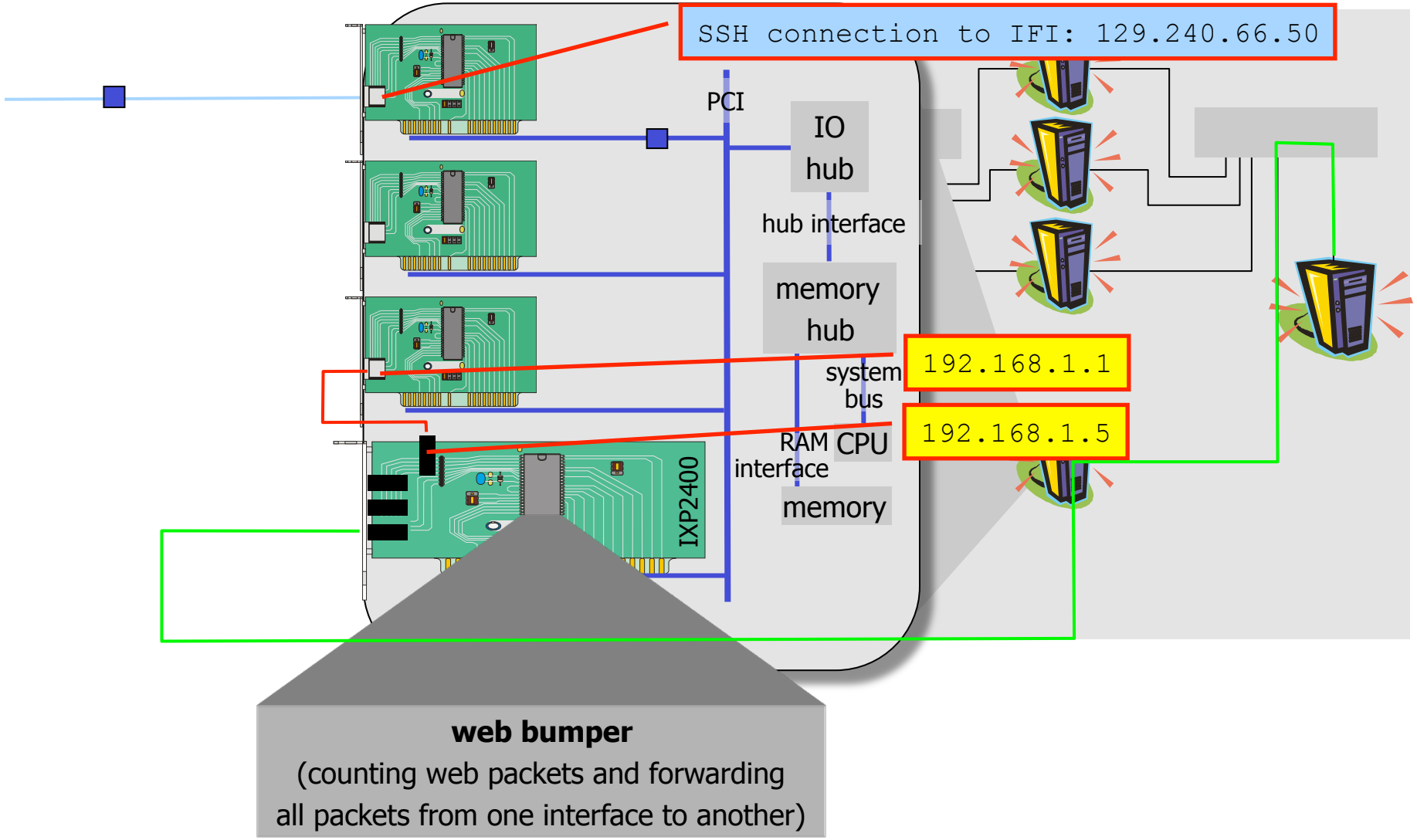


Lab Setup

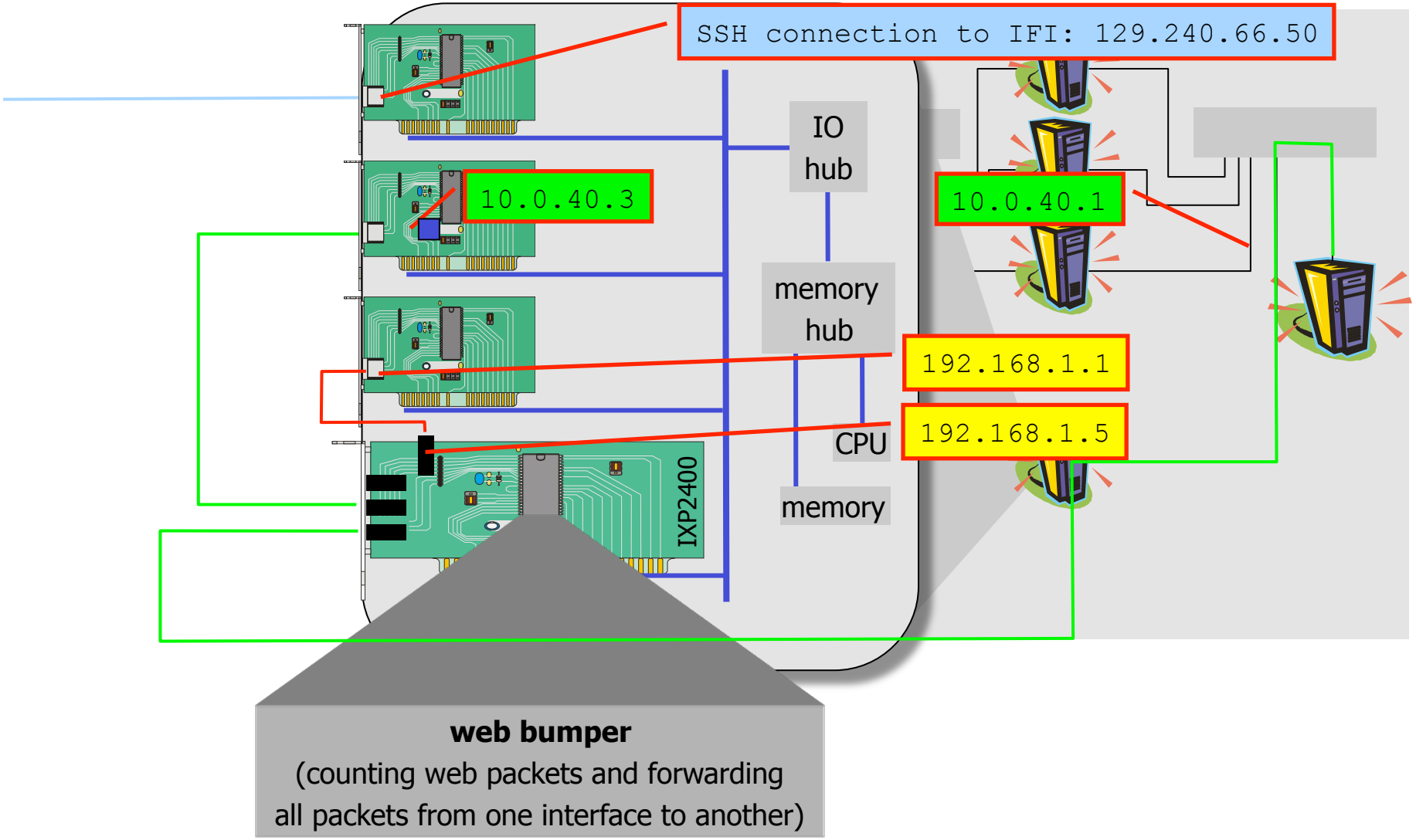
Lab Setup



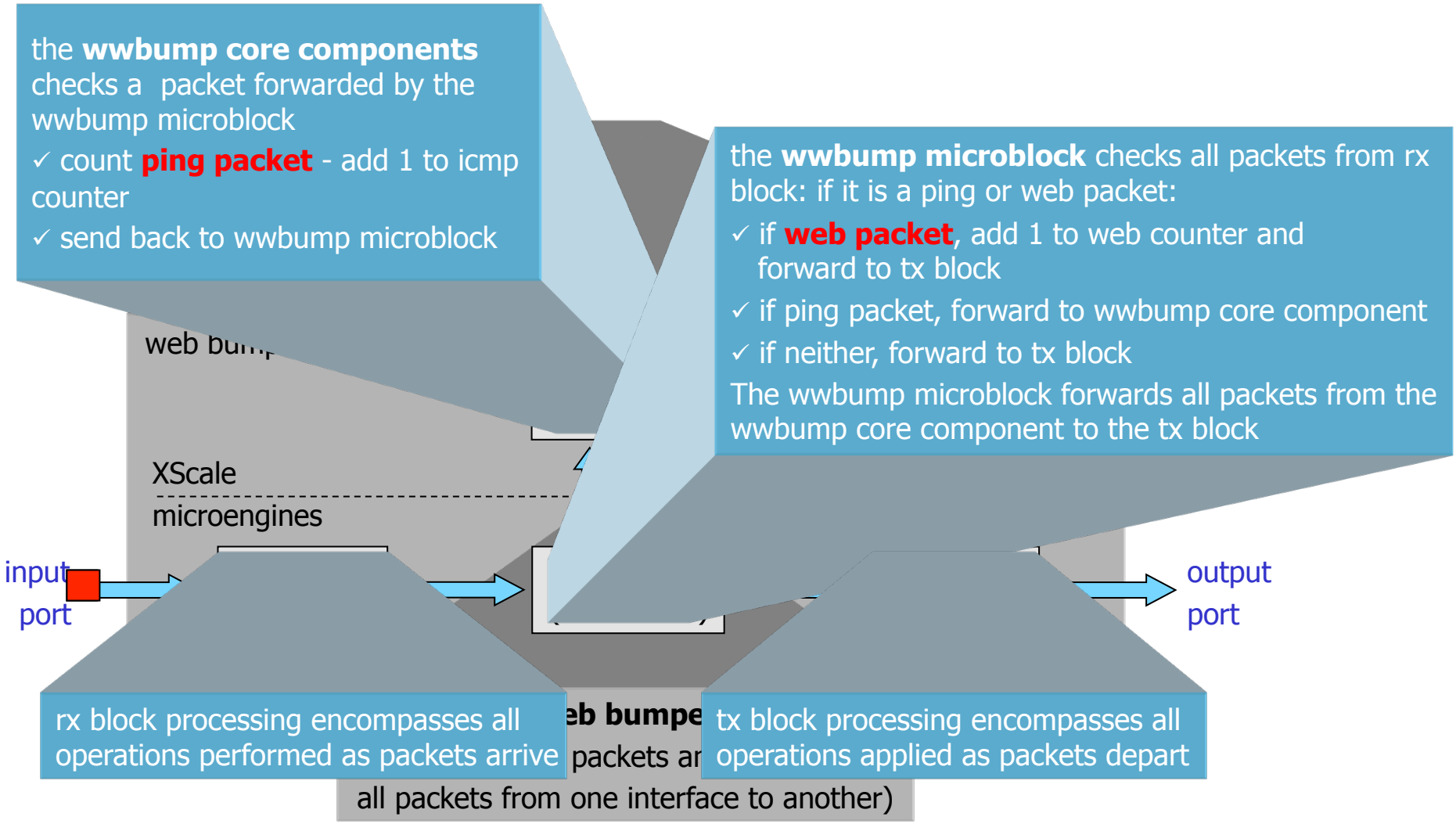
Lab Setup – Data Path



Lab Setup – Data Path



The Web Bumper



Starting and Stopping

- On the host machine
  - Location of the example: **`/root/ixa/wwpingbump`**
  - Rebooting the IXP card: **`make reset`**
  - Installing the example: **`make install`**
  - Telnet to the card:
    - **`telnet 192.168.1.5`**
    - **`minicom`**
    - **`screen /dev/ttyS0 57600`**
- On the card
  - To start the example: **`./wwbump`**
  - To stop the example: **`CTRL-C`**
- Let's look at an example...