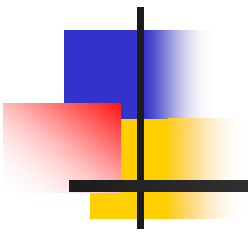
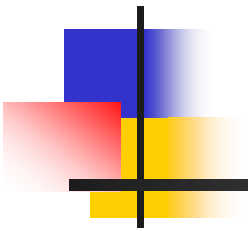


A First Example: The Bump in the Wire

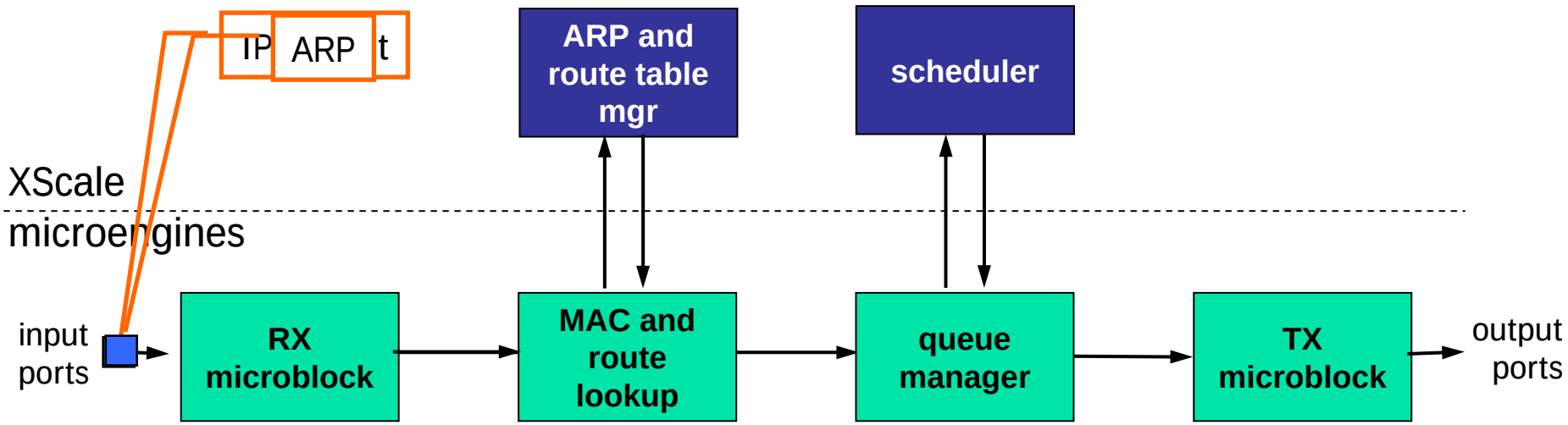
8/9 - 2006



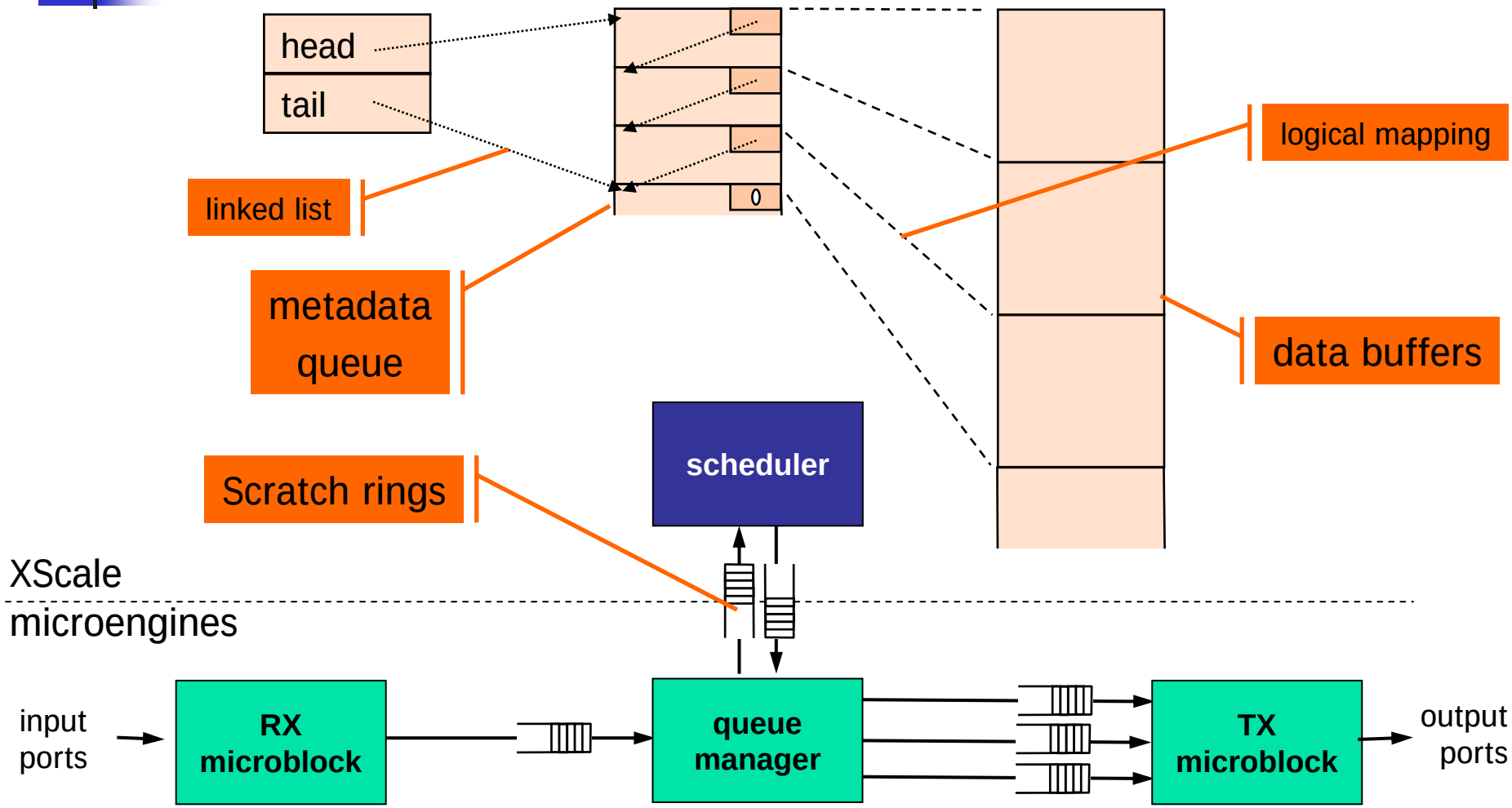
Using IXP2400

Programming Model

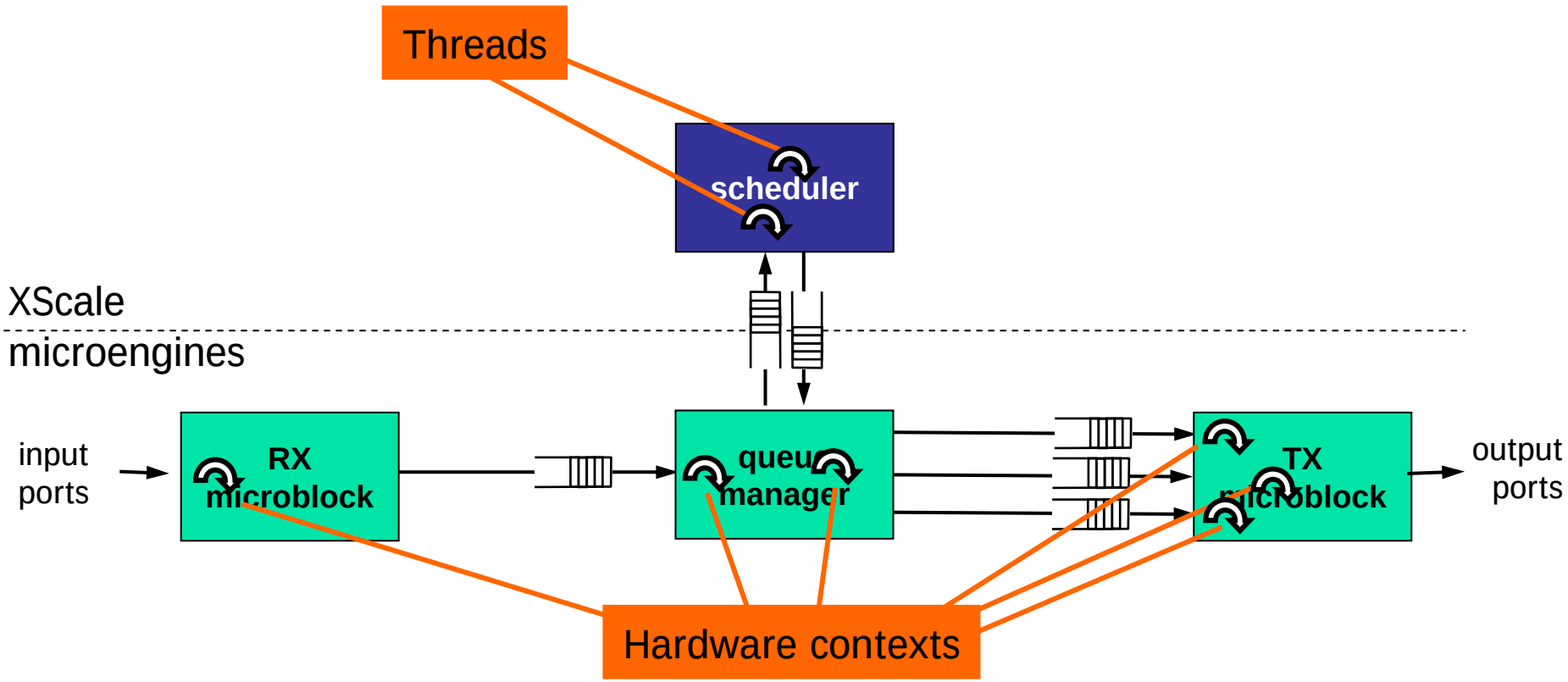
- ✓ Packet flow illustration for IP forwarding



Programming Model

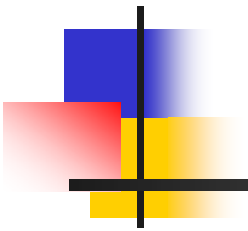


Programming Model



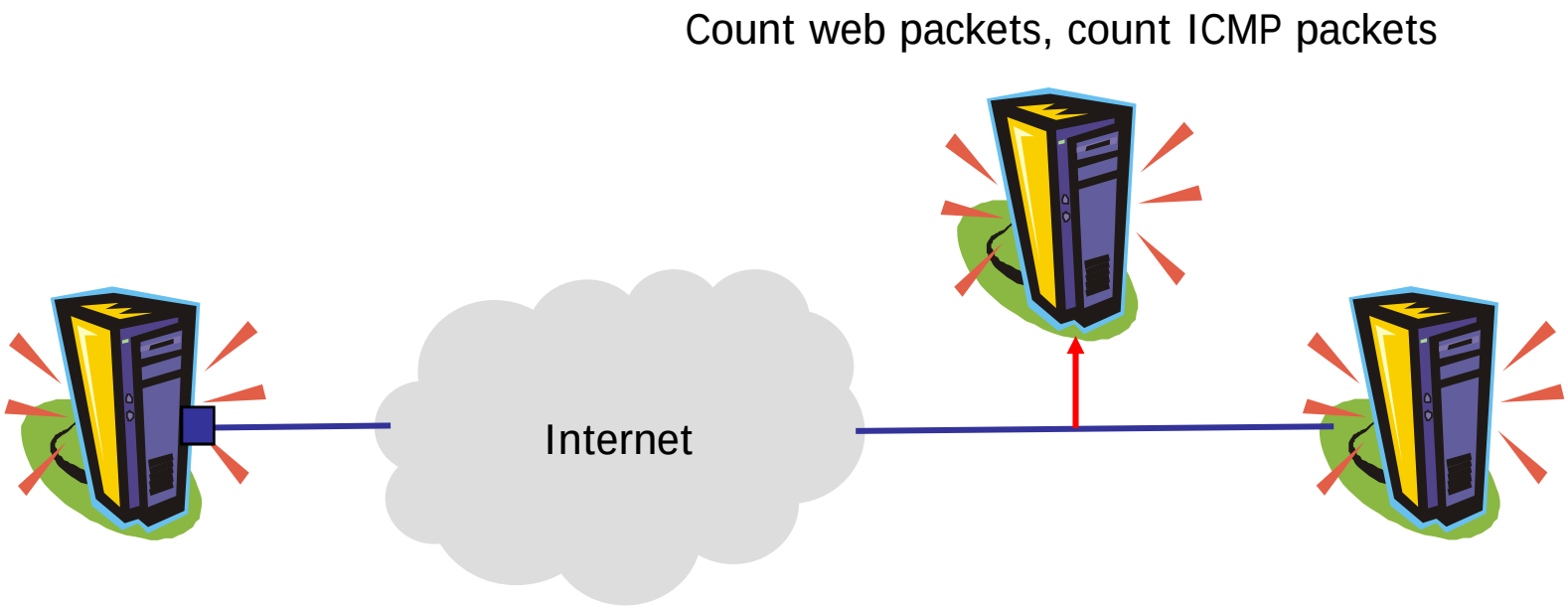
Framework

- 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_

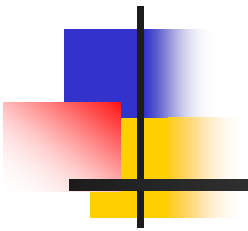


Bump in the Wire

Bump in the Wire

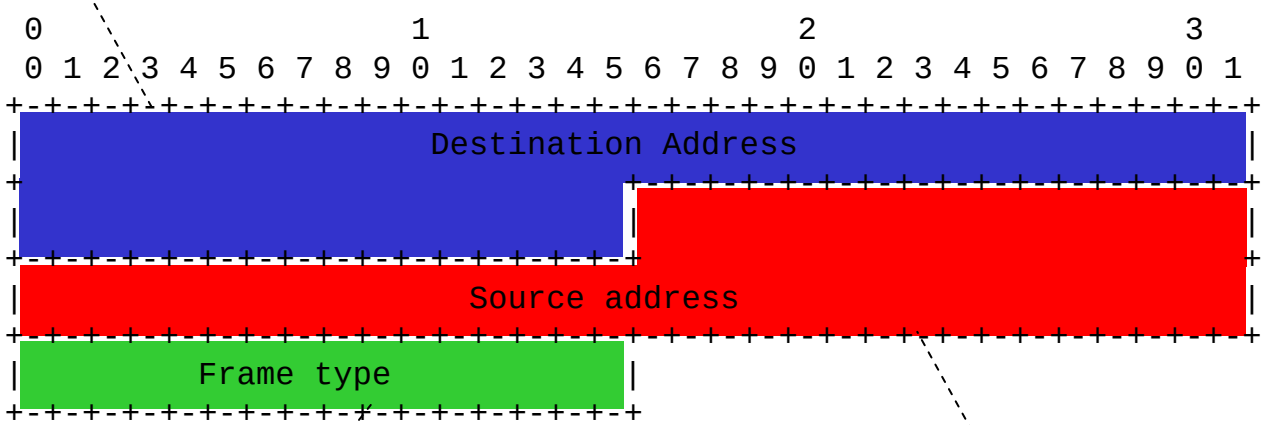


Packet Headers and Encapsulation



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

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

Internet Protocol version 4 (IPv4)

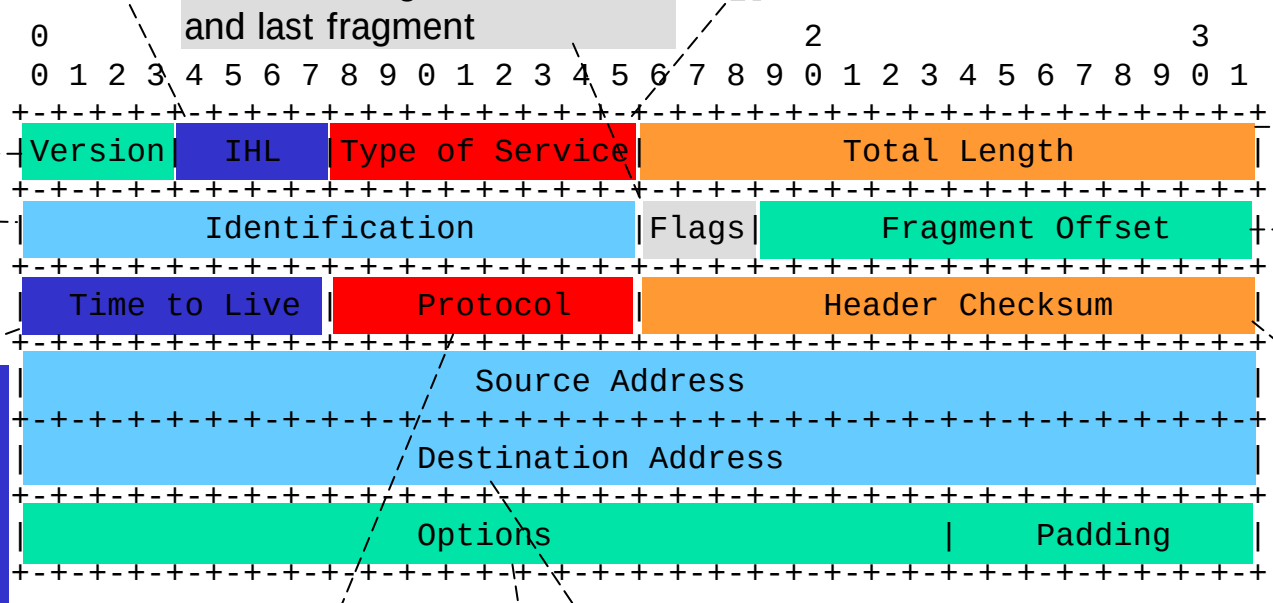
indicates the format of the internet header, i.e., version 4

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

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

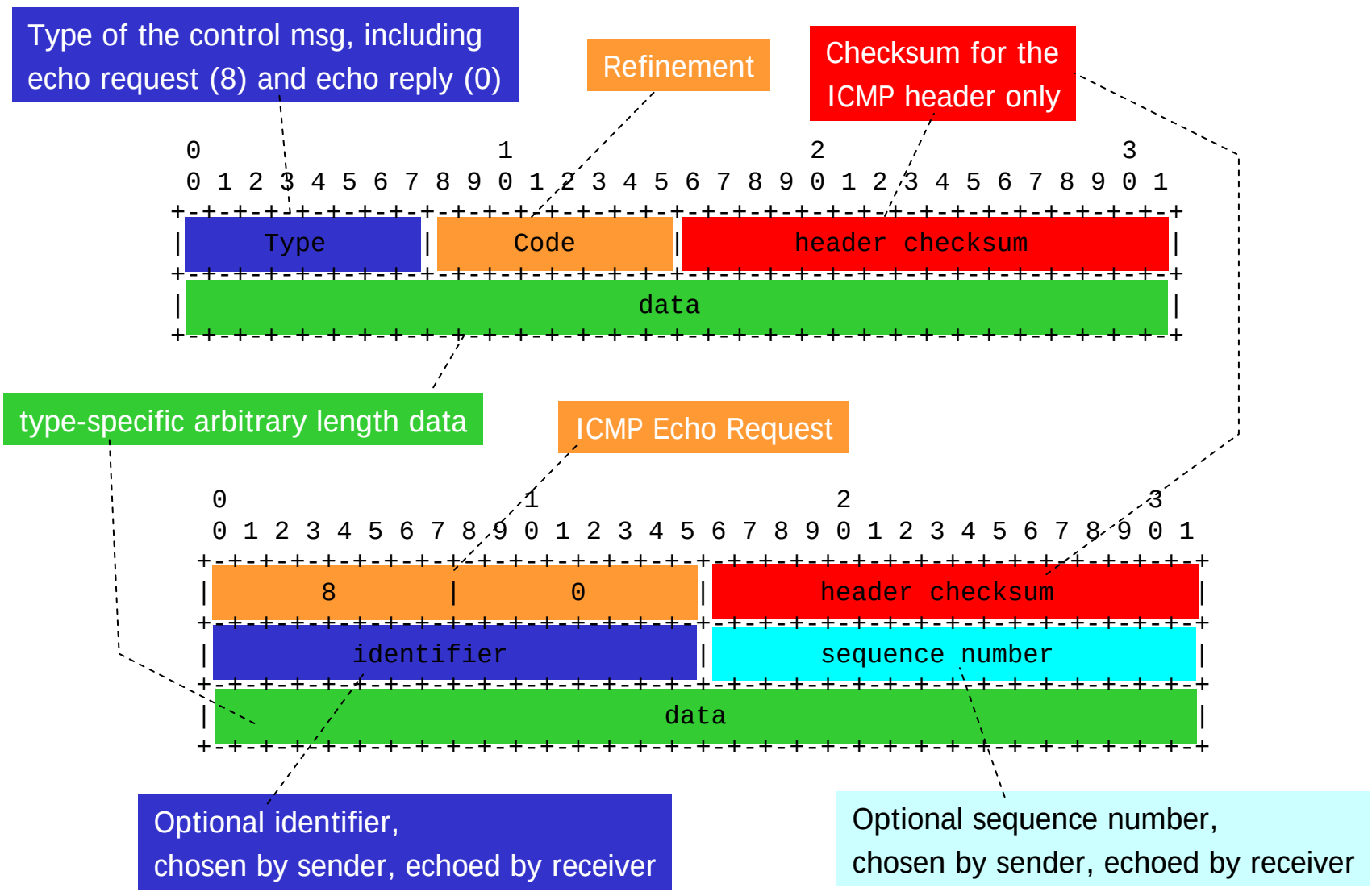
checksum on the header only – TCP, UDP over payload. Since some header fields change(TTL), this is recomputed and verified at each point

indicates used transport layer protocol

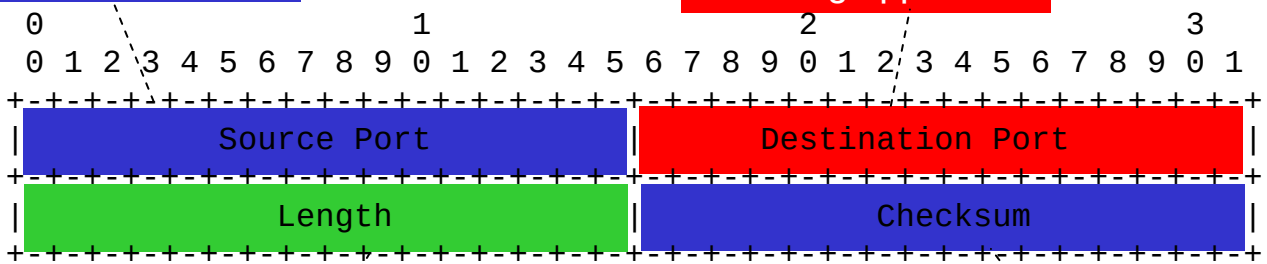
32-bit address fields. May be configured differently from small to large networks

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

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

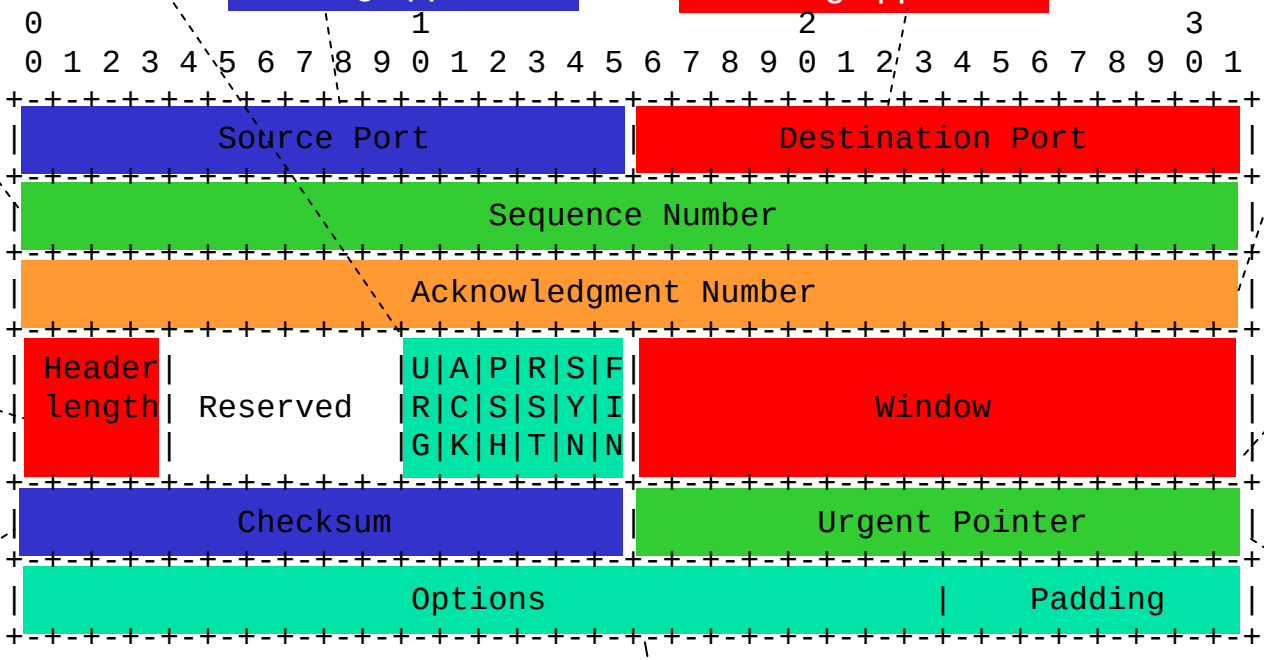
code bits: urgent, ack, push, reset, syn, fin

sequence number for data in payload

port to identify the sending application

port to identify receiving application

acknowledgement for data received



header length in 32 bit units

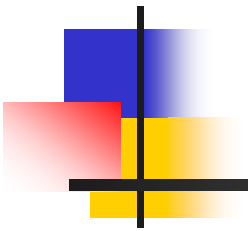
receiver's buffer size for additional data

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

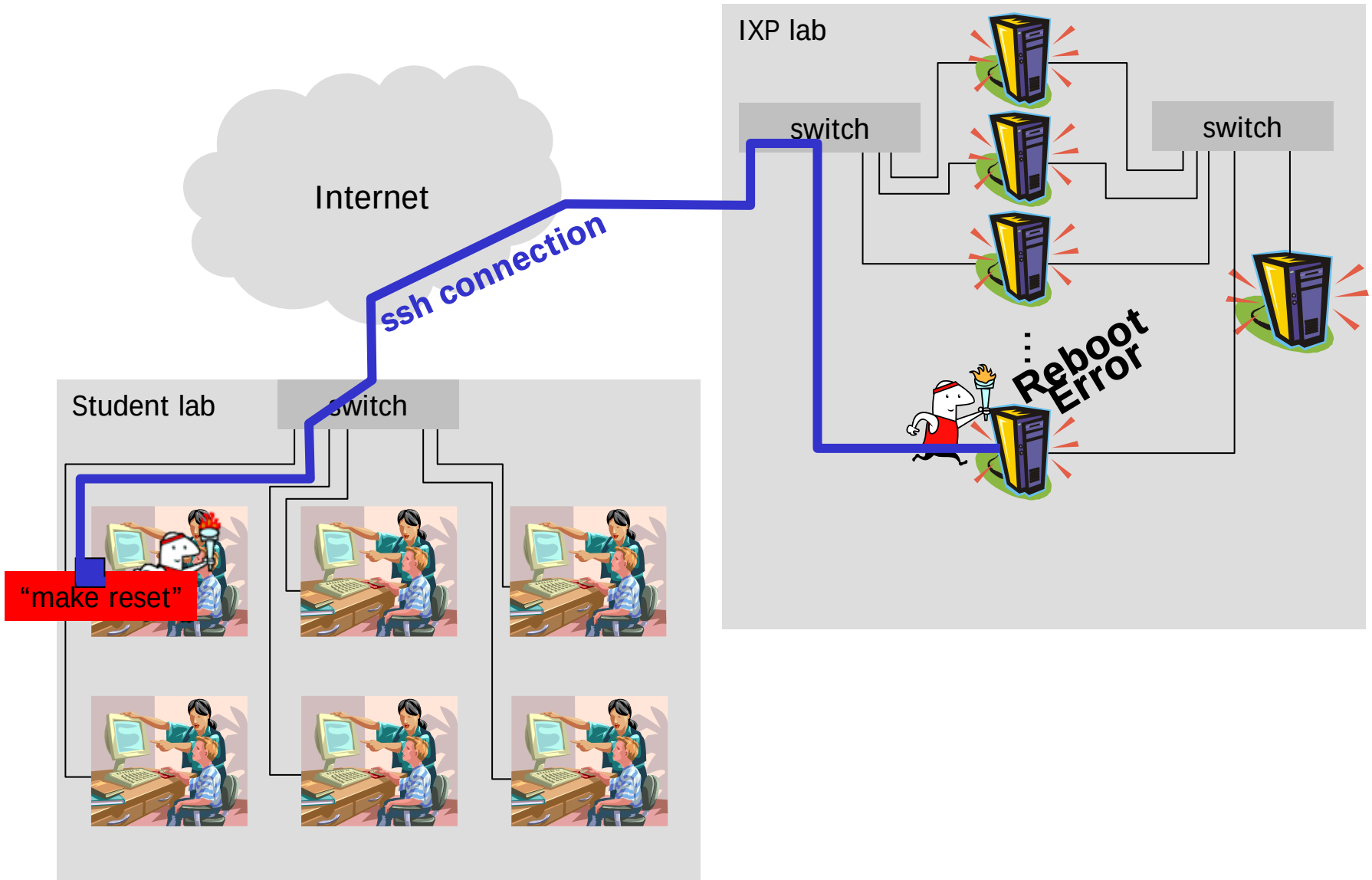
options may extend the header. If the options do not end on a 32-bit boundary, the remaining fields are padded in the padding field (0's)

Encapsulation

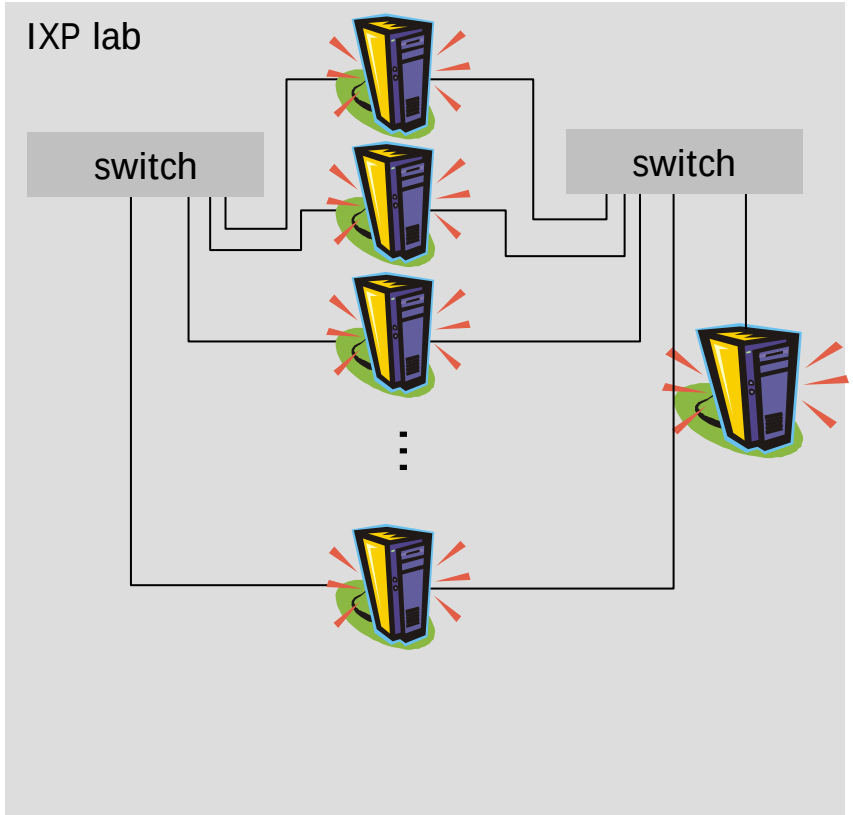


Lab Setup

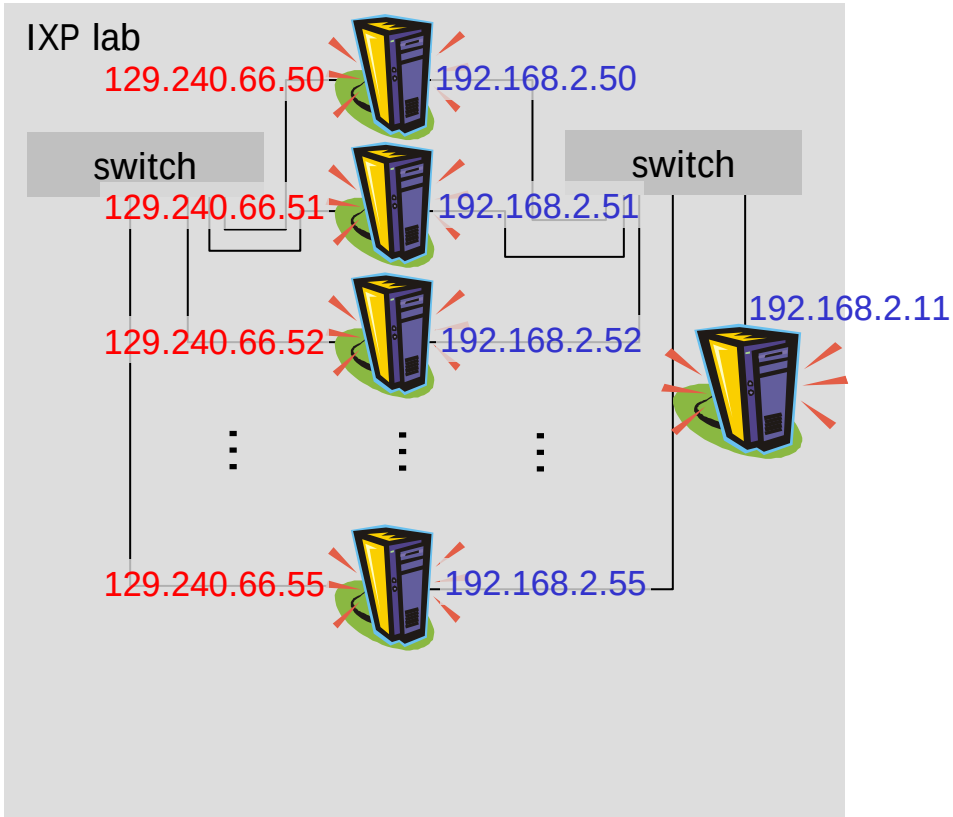
Lab Setup



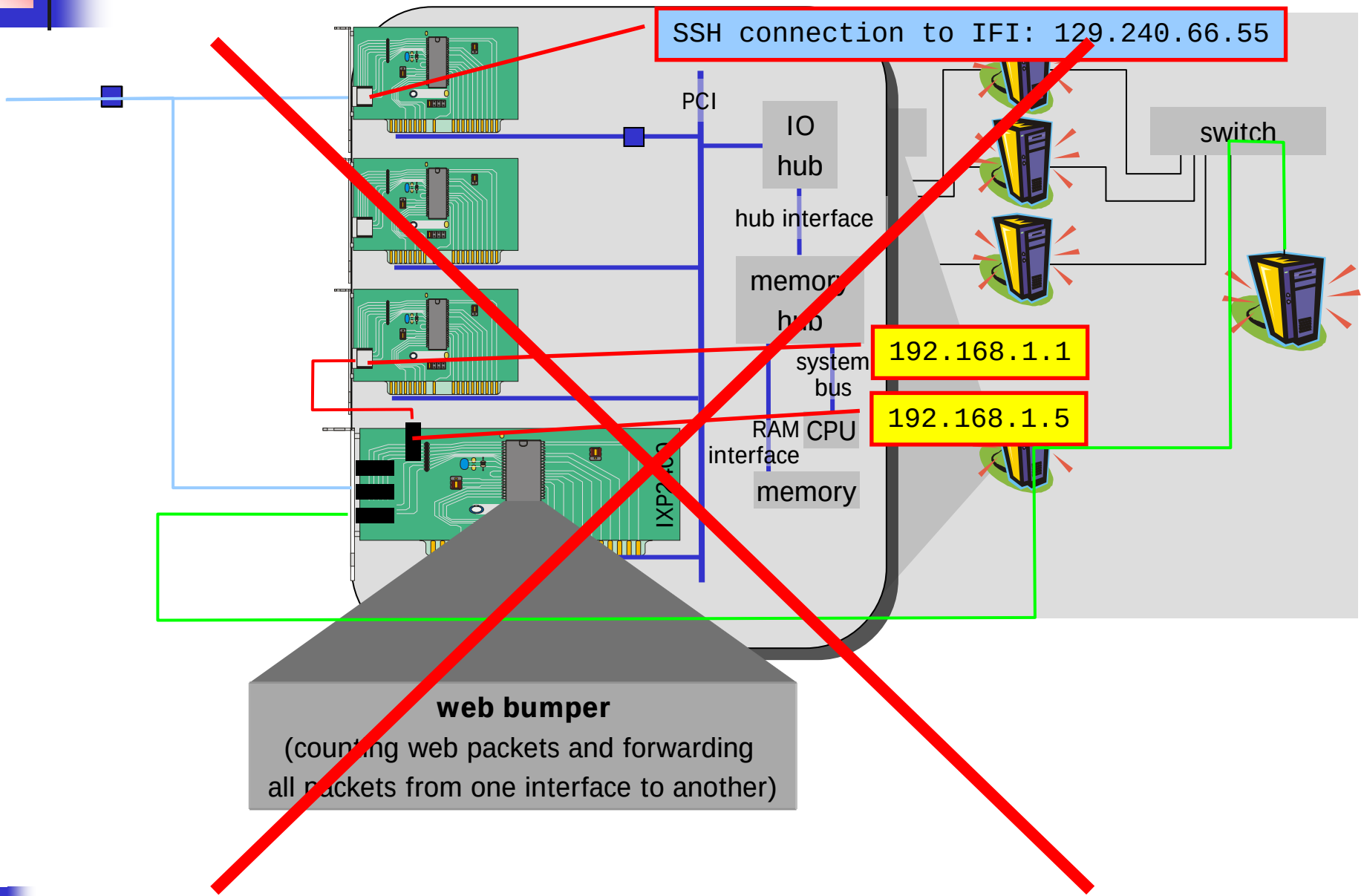
Lab Setup - Addresses



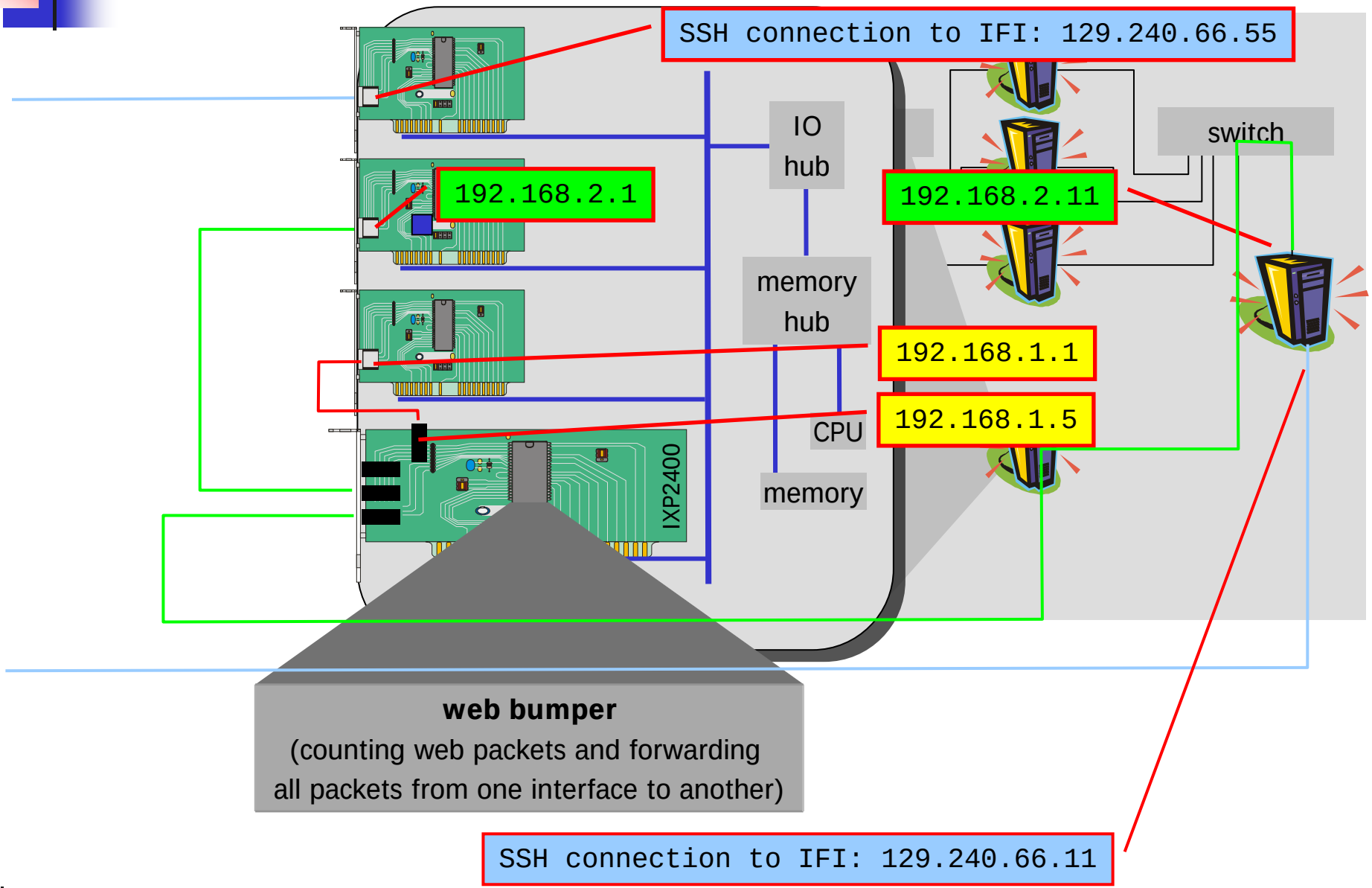
Lab Setup - Addresses



Lab Setup – Data Path



Lab Setup – Data Path



The Web Bumper

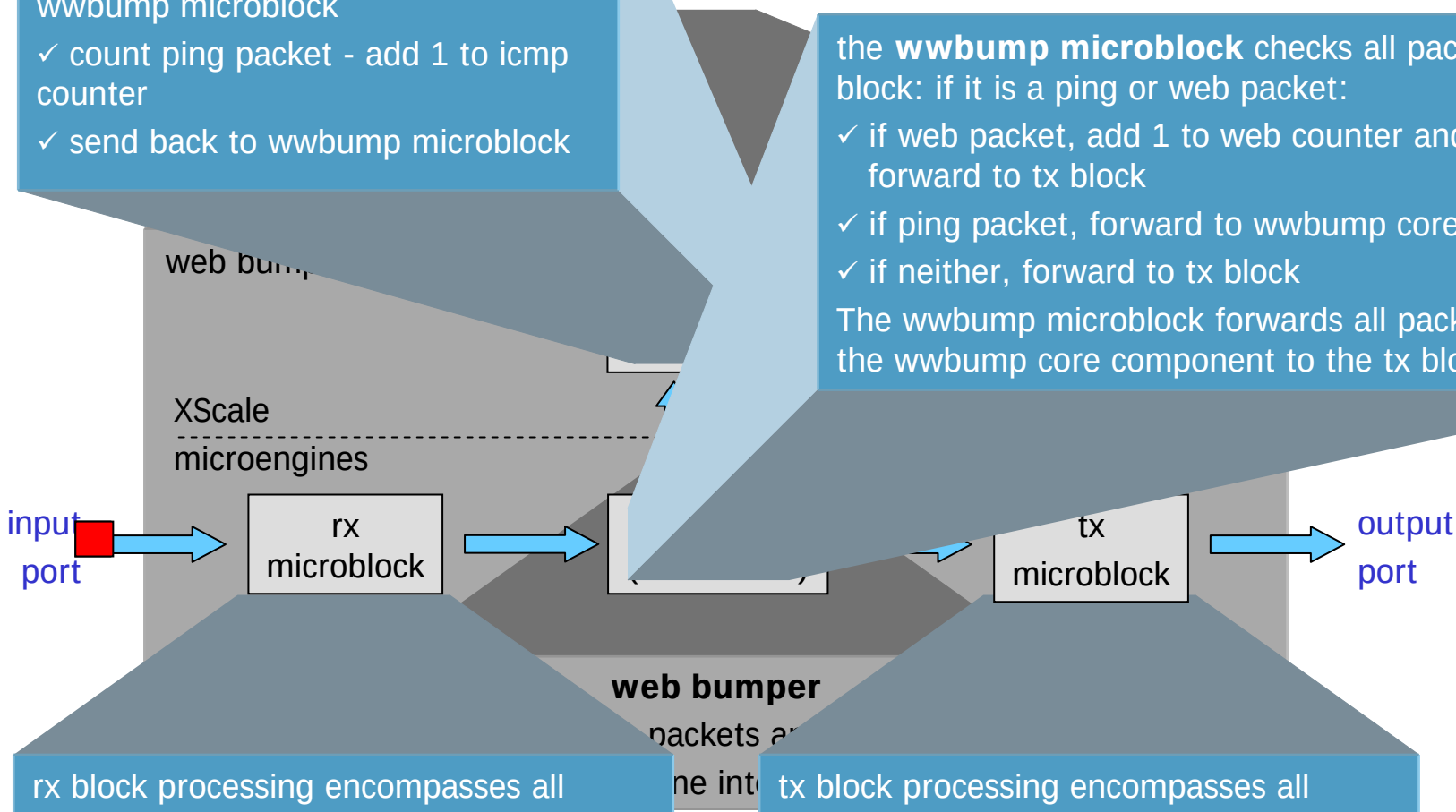
the **wwbump core components** checks a packet forwarded by the wwbump microblock

- ✓ count ping packet - add 1 to icmp counter
- ✓ send back to wwbump microblock

the **wwbump microblock** checks all packets from rx block: if it is a ping or web packet:

- ✓ if web packet, add 1 to web counter and forward to tx block
- ✓ if ping packet, forward to wwbump core component
- ✓ if neither, forward to tx block

The wwbump microblock forwards all packets from the wwbump core component to the tx block



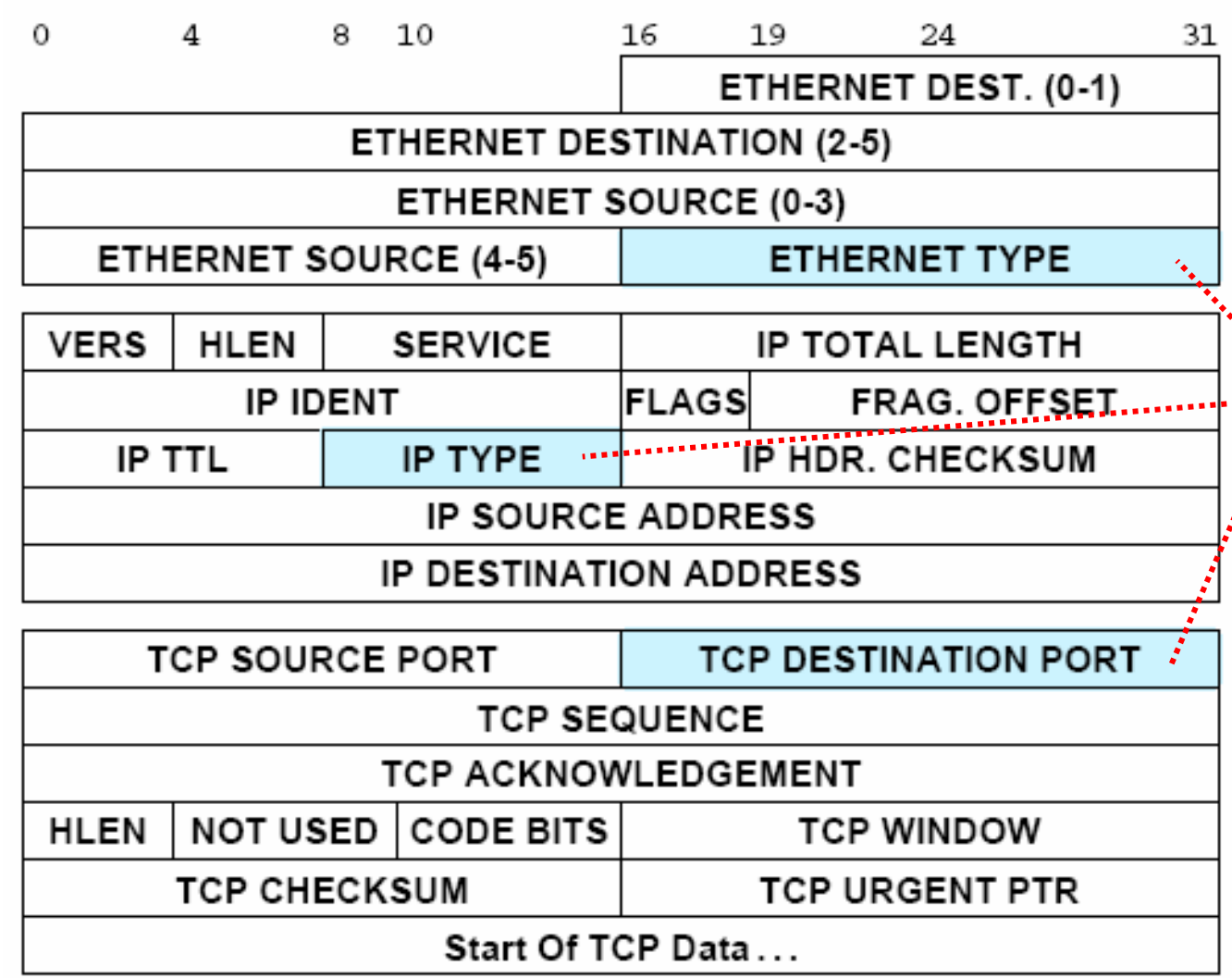
rx block processing encompasses all operations performed as packets arrive

tx block processing encompasses all operations applied as packets depart

Starting and Stopping

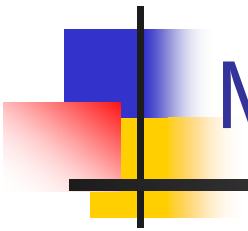
- On the host machine
  - Location of the example: **`/root/ixa/wwpingbump`**
  - Rebooting the IXP card: **`make reset`**
  - Installing the example: **`make install`**
  - Telnet to the card:
    - **`telnet 192.168.1.5`**
    - **`minicom`** ☺
- On the card
  - To start the example: **`./wwbump`**
  - To stop the example: **`CTRL-C`**

Identifying Web Packets



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

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

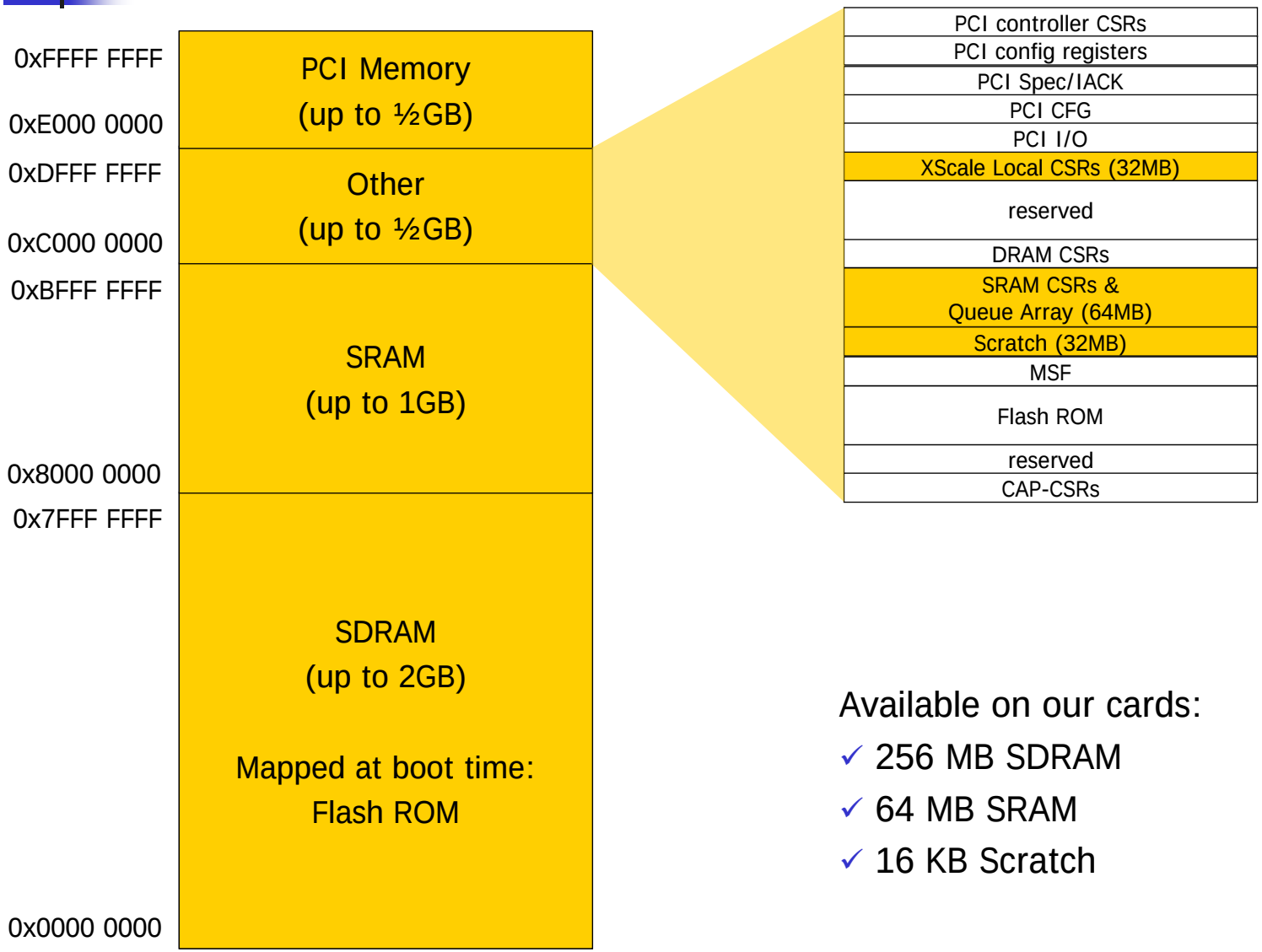


Memory

Intel IXP2400 Hardware Reference Manual

This page intentionally left blank.

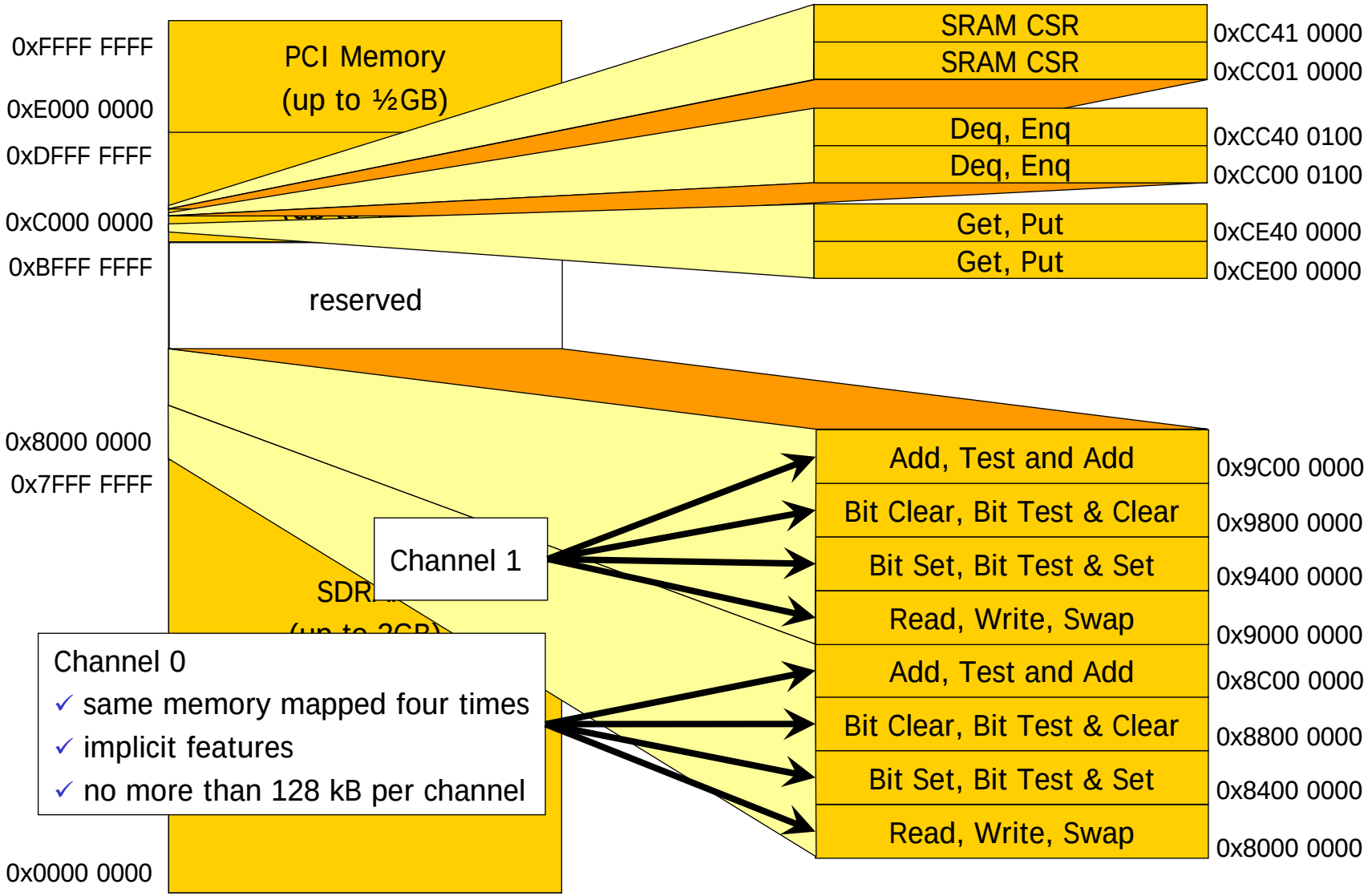
XScale Memory Mapping



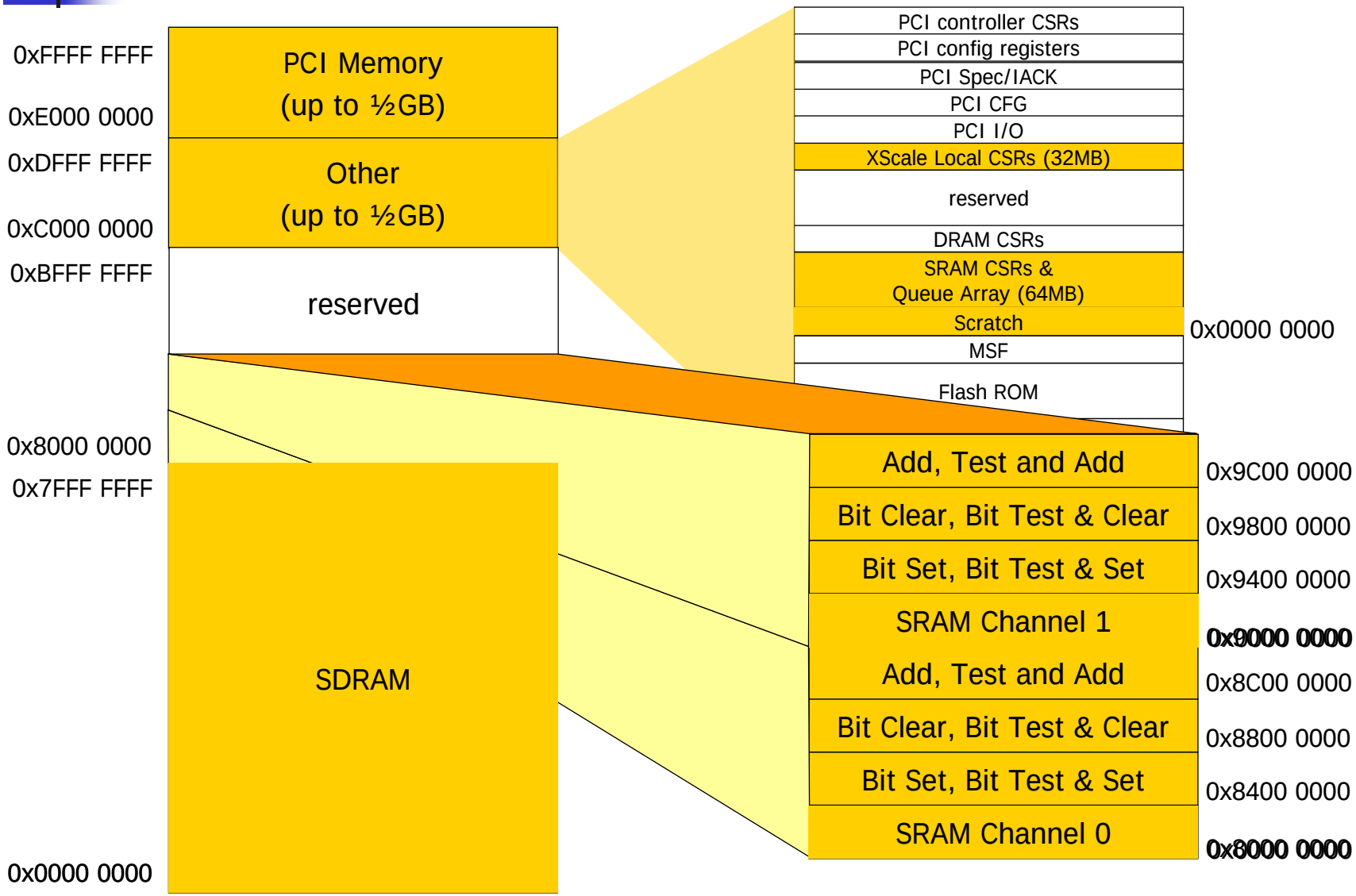
Available on our cards:

- ✓ 256 MB SDRAM
- ✓ 64 MB SRAM
- ✓ 16 KB Scratch

XScale Memory Mapping



Microengine Memory Mapping



XScale Memory

- A general purpose processor
  - With MMU
    - In use!
  - 32 Kbytes instruction cache
    - Round robin replacement
  - 32 Kbytes data cache
    - Round robin replacement
    - Write-back cache, cache replacement on read, not on write
  - 2 Kbytes mini-cache for data that is used once and then discarded
    - To reduce flushing of the main data cache
  - Instruction code stored in SDRAM

Microengine Memory

- 256 general purpose registers
  - Arranged in two banks
- 512 transfer registers
  - Transfer registers are not general purpose registers
  - DRAM transfer registers
    - Transfer in
    - Transfer out
  - SRAM transfer registers
    - Transfer in
    - Transfer out
  - Push and pull on transfer registers usually by external units
- 128 next neighbor registers
  - New in ME V2
  - Dedicated data path to neighboring ME
  - Also usable inside a ME
  - SDK use: message forwarding using rings
- 2560 bytes local memory
  - New in ME V2
  - RAM
  - Quad-aligned
  - Shared by all contexts
  - SDK use: register spill in code generated from MicroC

SDRAM

- Recommended use
  - XScale instruction code
  - Large data structures
  - Packets during processing
- 64-bit addressed (8 byte aligned, quadword aligned)
- Up to 2GB
  - Our cards have 256 MB
  - Unused higher addresses map onto lower addresses!
- 2.4 Gbps peak bandwidth
  - Higher bandwidth than SRAM
  - Higher latency than SRAM
- Access
  - Instruction from external devices are queued and scheduled
  - Accessed by
    - XScale
    - Microengines
    - PCI

SRAM

- Recommended use
  - Lookup tables
  - Free buffer lists
  - Data buffer queue lists
- 32-bit addressed (4 byte aligned, word aligned)
- Up to 16 MB
  - Distributed over 4 channels
  - Our cards have 8 MB, use 2 channels
- 1.6 Gbps peak bandwidth
  - Lower bandwidth than SDRAM
  - Lower latency than SDRAM
- Access
  - XScale
  - Microengines
- Accessing SRAM
  - XScale access
    - Byte, word and longword access
  - Microengine access
    - Bit and longword access only

SRAM Special Features

- ✓ Atomic bit set and clear with/without test
- ✓ Atomic increment/decrement
- ✓ Atomic add and swap
- ✓ Atomic enqueue, enqueue_tail, dequeue
  - Hardware support for maintaining queues
  - Combination enqueue/enqueue_tail allows merging of queues
  - Several modes
    - Queue mode: data structures at discontiguous addresses
    - Ring mode: data structures in a fixed-size array
    - Journaling mode: keep previous values in a fixed-size array

Scratch Memory

- Recommended use
  - Passing messages between processors and between threads
  - Semaphores, mailboxes, other IPC
- 32-bit addressed (4 byte aligned, word aligned)
- 4 Kbytes
- Has an atomic autoincrement instruction
  - Only usable by microengines

Scratchpad Special Features

- ✓ Atomic bit set and clear with/without test
- ✓ Atomic increment/decrement
- ✓ Atomic add and swap
- ✓ Atomic get/put for rings
  - Hardware support for rings links SRAM
  - Signaling when ring is full