Data Networks

Introduction to Networking

Dan Octavian Savu
&
Stefan Stancu

CERN

ISOTDAQ 2014, Budapest

Outline

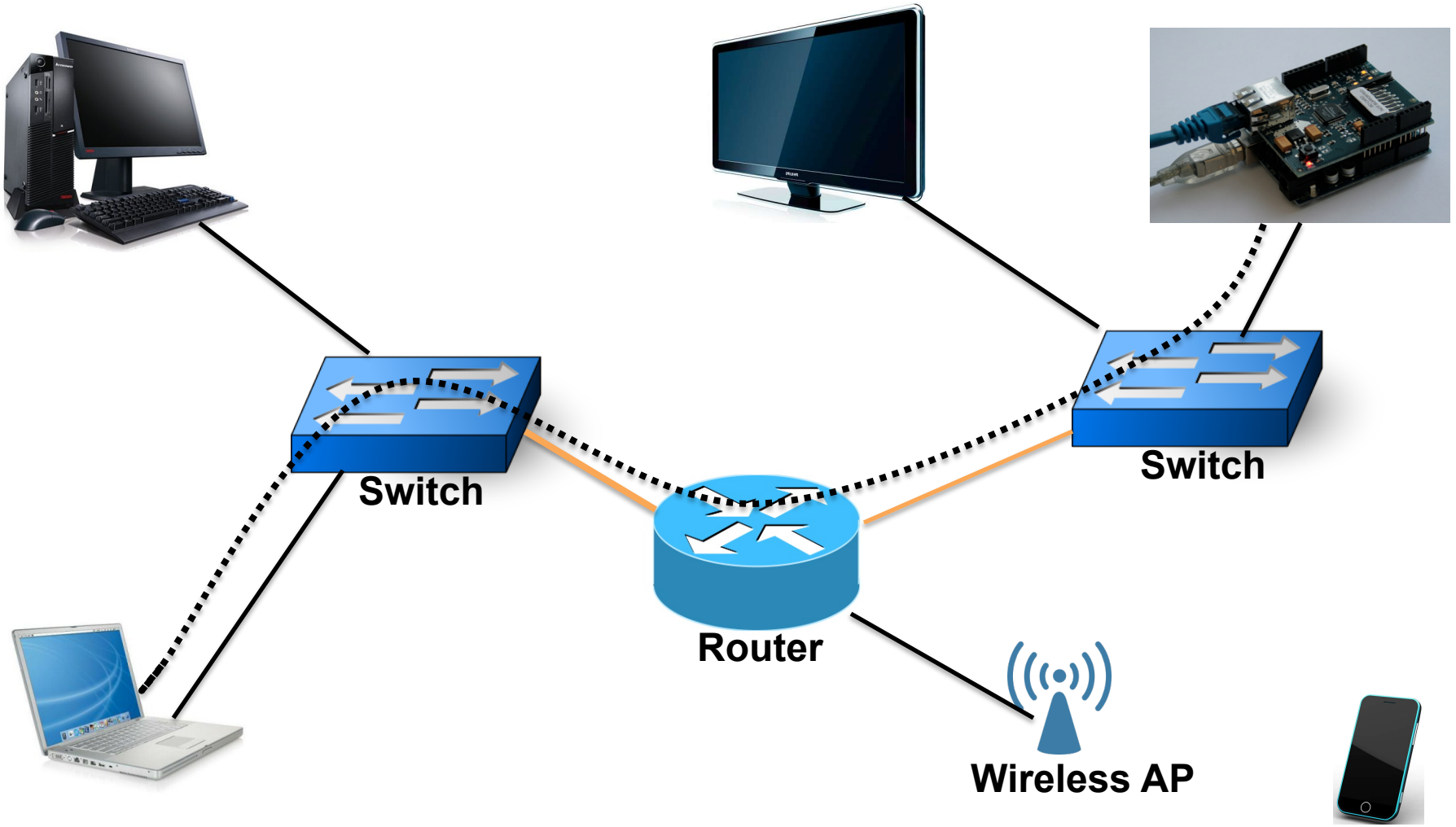
- Introduction
  - Networking basics
  - OSI reference model
- Technologies and protocols
  - Ethernet
  - Internet Protocol (IP)
  - Routers and routing
- Network monitoring
- Software defined networking

What is a network ?

- A ***network*** is simply two or more computers connected together so they can exchange information. At the same time it can be a complex interconnected system of objects and people (Internet)

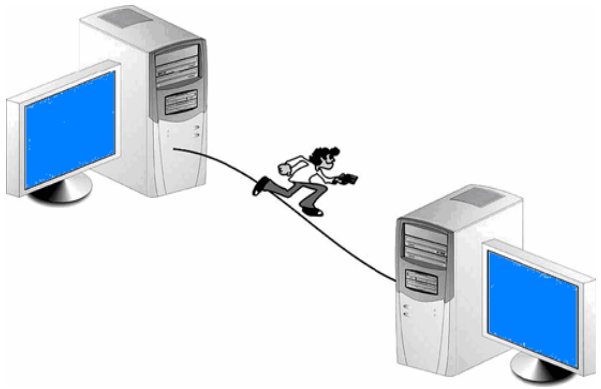
- ***End-host devices*** are hosts attached to a network
- A ***source host*** is the place where the data originally comes from
- A ***destination host*** is the place where the data is being sent to
- ***Networking devices*** are waypoints along paths for data to travel along
- **Links** are direct data paths between adjacent devices
- A **route** is the path between any two network points

What is a network ?



Why do we need a network ?

- Sneaker Net
  - Inefficient data communication;
  - Many copies of the same file;
  - Reliability, scalability, flexibility… issues.



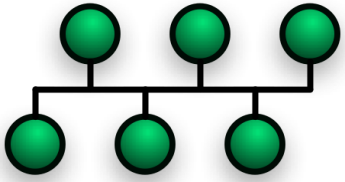
- (High speed) networks connecting all hosts help address slow transmission of information
- Interconnected datacenter servers help minimize redundant copy of files
- File sharing, resource sharing, communication & collaboration, group organization, remote access, data backups etc

Network types

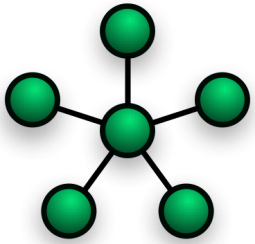
- Networks have different varieties to suit different purposes and needs
- **LAN** (small size, high speed, physical proximity)
- **WAN** (long distance, lower data transfer rates)
- **MAN** (metropolitan area network)
- **PAN** (immediate space around a person)
- **SAN** (connecting storage farms, high speed)
- **VPN** (private network extension across a shared or a public network)

Network structure

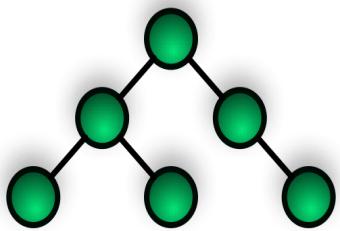
- The structure of a network is known as the ***topology***
  - **Physical** = The way the network is cabled
  - **Logical** = The way devices use the network to communicate



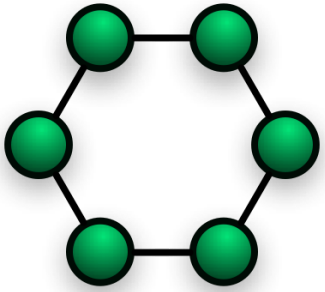
Bus Topology



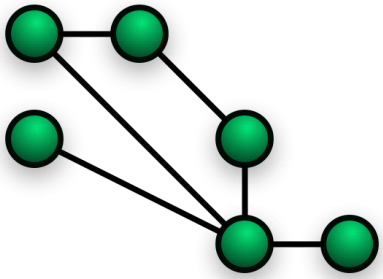
Star Topology



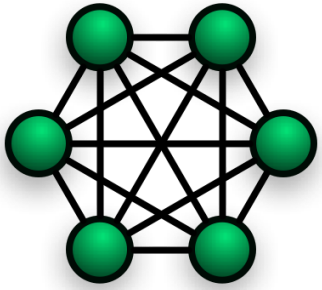
Hierarchical Topology



Ring Topology



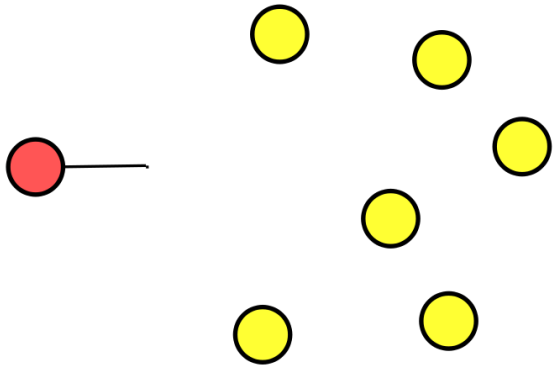
Partial Mesh Topology

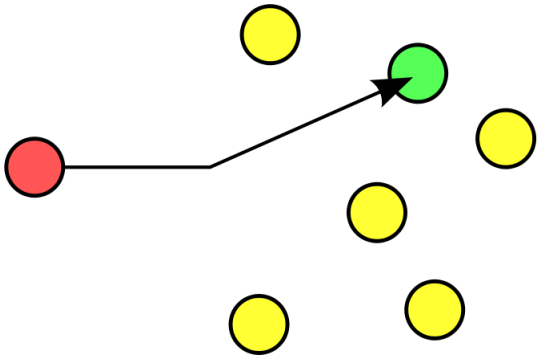


Fully Mesh Topology

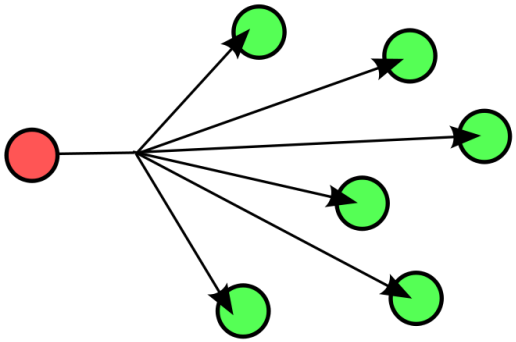
Network communication

- One-to-one
- One-to-all
- One-to-many

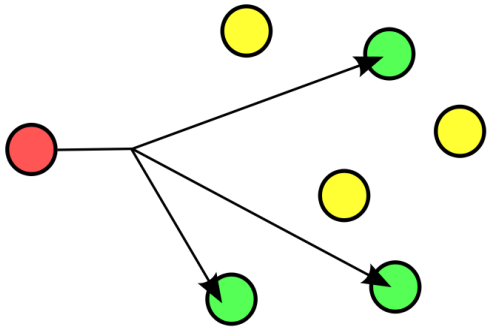




Unicast

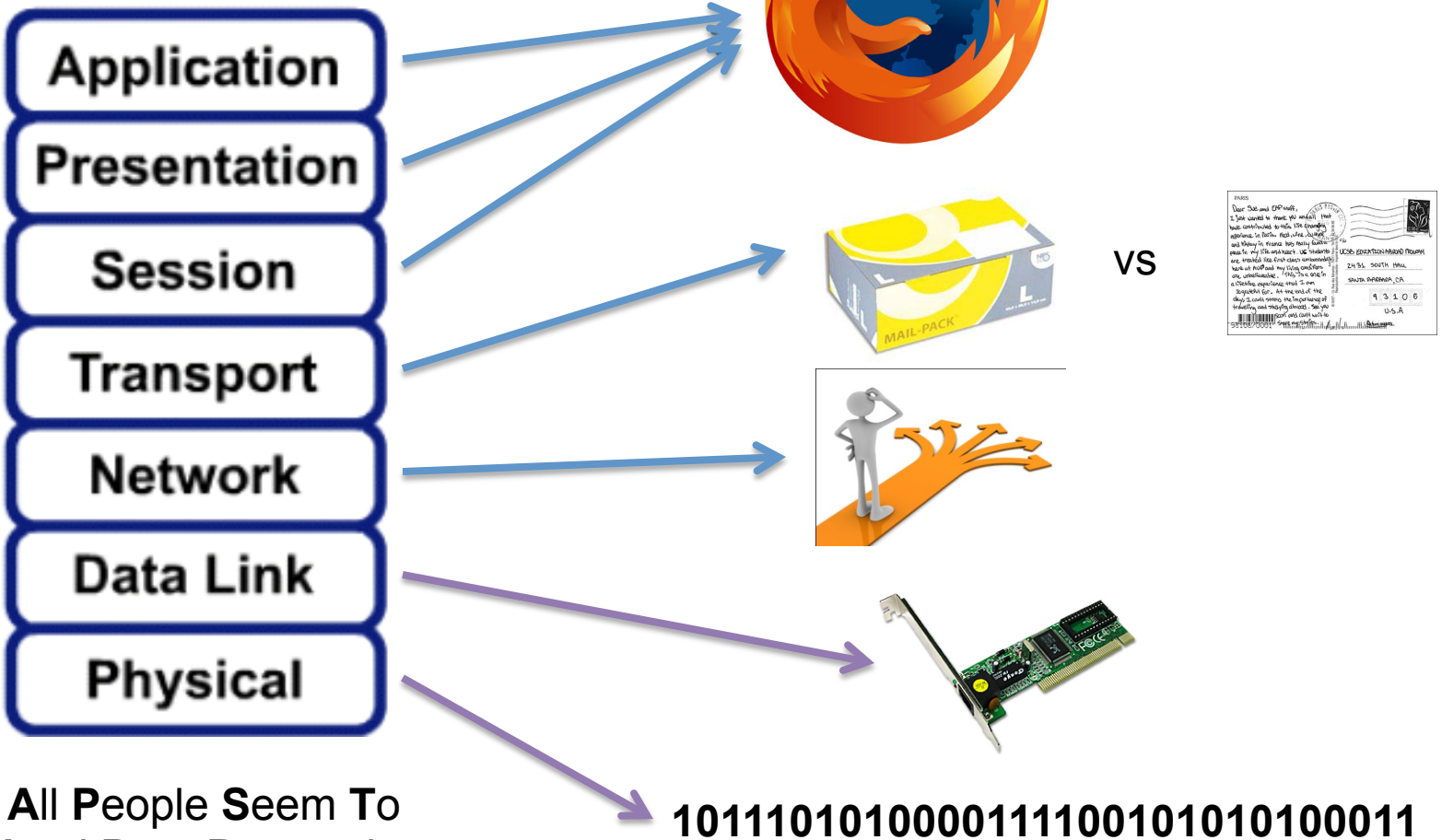


Broadcast



Multicast

OSI Model. Divide et impera.



OSI Model. Divide et impera.

All **P**eople **S**eem **T**o **N**eed **D**ata **P**rocessing

Why layers in OSI ?

- Simplifies understanding of networking
- Breaks networking tasks into smaller, manageable, chunks
- Allows for platform independence
- Provides a standard for networking manufactures
- Easier to determine the correct networking protocol required to connect
- Problem investigation is easier and debugging time is shortened

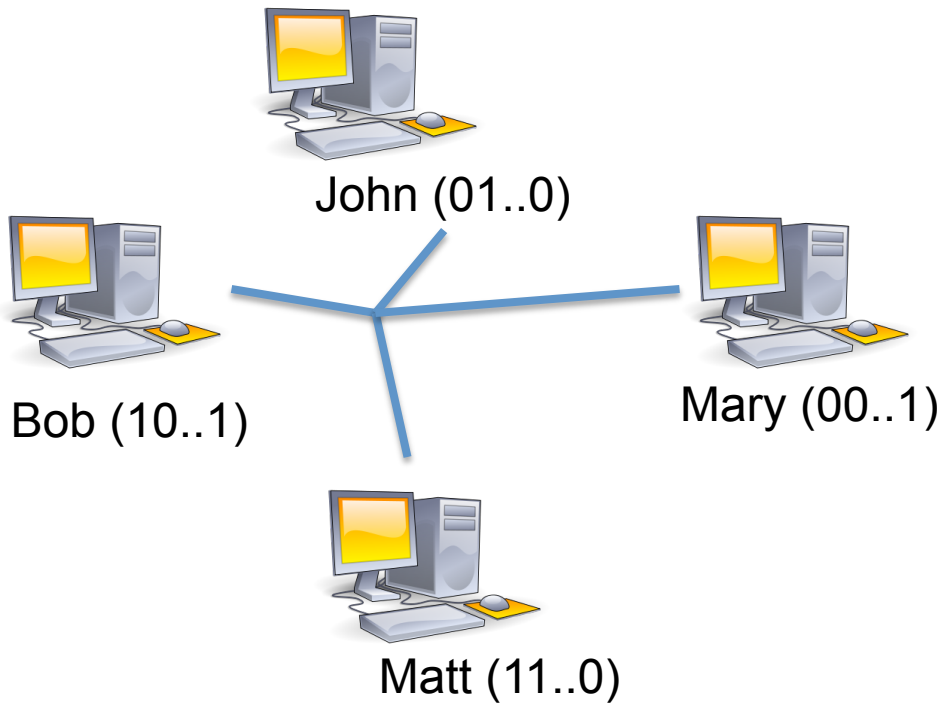
Outline

- Introduction
  - Networking basics
  - OSI Model
- Technologies and protocols
  - Ethernet
  - IP Protocol
  - Routers and routing
- Network monitoring
- Software defined networking

Ethernet. Reliable since 1973.

- Created at Xerox in 1973, released as an open standard in the early 80s
- Later modified to comply with the OSI model, ratified as IEEE 802.3 in 1985
- Ethernet has evolved significantly since then:
  - Proved flexible as a technology, able to upgrade to new media and faster data transmission speeds.
  - 10Gig Ethernet ratified as IEEE 802.3ae
  - Optical fiber has joined copper as media of choice for the IEEE 802.3 family
- Flexibility came through the simplicity of Ethernet's structure
- Ease of installation and maintenance

Ethernet

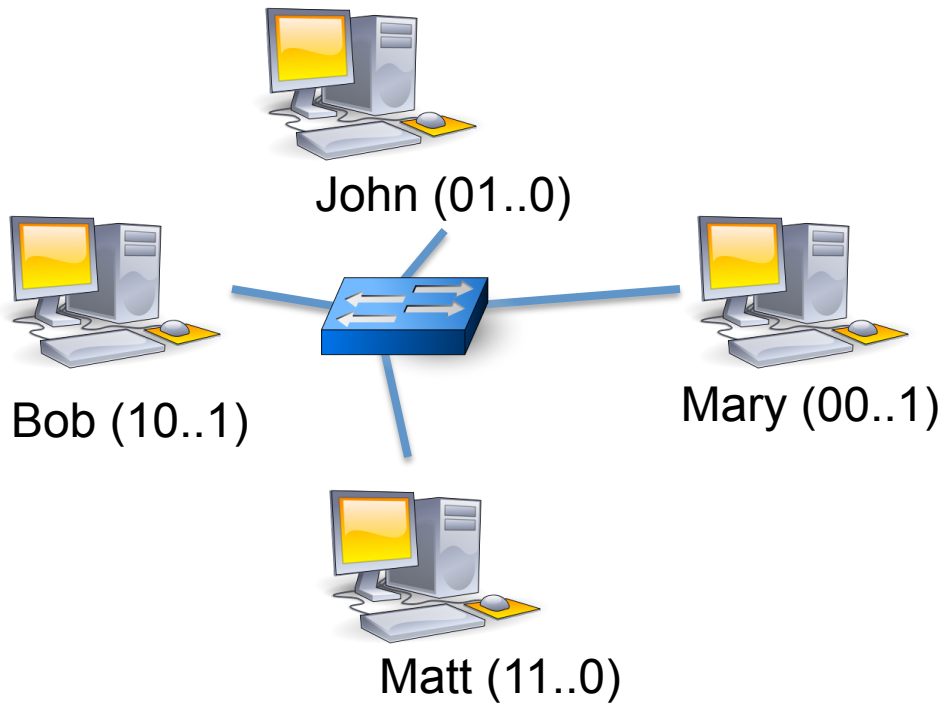


- Small to medium size group of computers
- Frame based technology
- Defines wiring and signaling standards (Layer1)
- Defines a flat addressing scheme with local visibility, called **MAC** (Layer 2)

8	6	6	2	46 ~ 1500 bytes	4
Pre-amble	Dest.	Source	Type/ Length	Data	Frame check

Basic Ethernet frame

Ethernet. Switch



- Analyses incoming frames and switches them to correct segment using MAC addresses (a process called switching)
- Simultaneous data transmissions without medium sharing
- Layer 2 device

8	6	6	2	46 ~ 1500 bytes	4
Pre-amble	Dest.	Source	Type/ Length	Data	Frame check

Basic Ethernet frame

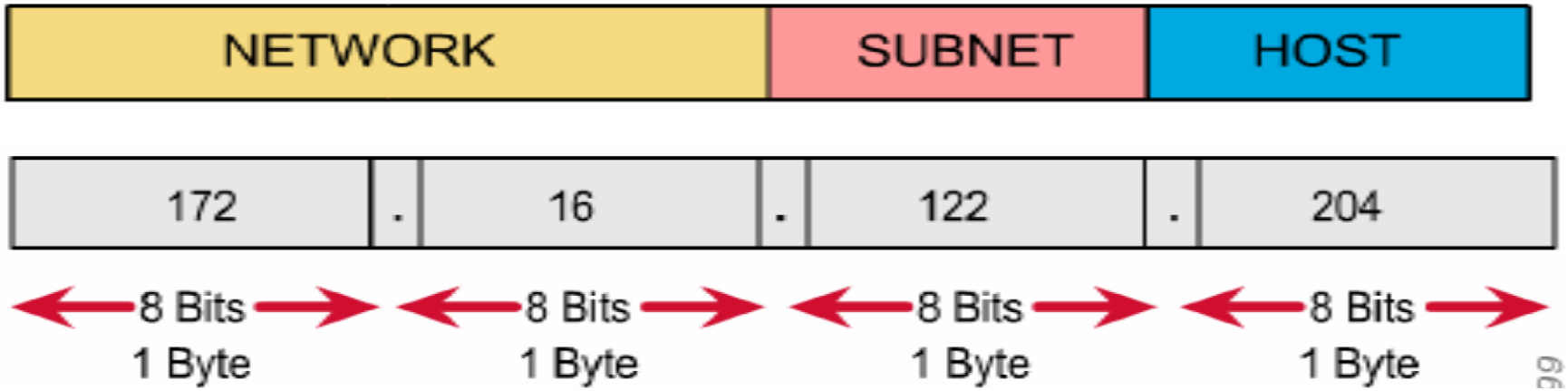
IP. (Un)reliable since 1974.

- Connectionless, best effort protocol
- Designed to be encapsulated into layer 2 protocols , such as Ethernet
- Initially created by Vint Cerf and Bob Kahn in 1974
- IPv4 described in RFC 791 (1981) hyperlink

- Defines a hierarchical (logical) addressing scheme capable of connecting all the hosts in the world (Layer 3)
- Routes packets towards destination using best available path, with the help of routing protocols (Layer 3)

IP (Addressing)

- 32bit address space (IPv4)
- Hierarchical addressing (similar to postal addressing)
- Global visibility
- ARP (Address Resolution Protocol) used to map an IP address with an Ethernet MAC address (layer 2, local visibility)



Routers

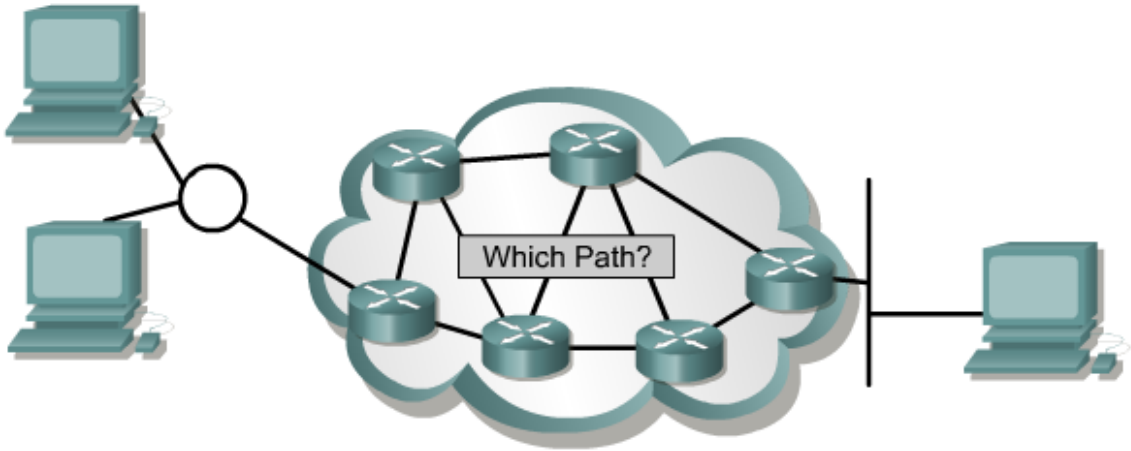
- **Connect** together **separate networks**, sometime of various networking technologies (ex: Ethernet and DSL)
- Make path determination decision based upon logical addresses (such as IP). The process is called **routing**.
- Layer 3 networking devices
- Routing and switching are similar concepts, but are in different layers:
  - Routing occurs in Layer 3, uses IP
    - Maintains routing tables (IP network addresses)
    - Maintains ARP tables (IP to MAC mappings)
  - Switching occurs in layer 2, uses MAC
    - Maintains switching tables (MAC addresses)

Routing

The **process of selecting paths** in a network along which to send network traffic, based upon logical addresses (such as IP).

A routing protocol allows one router to share information with other routers regarding known network paths as well as its proximity

- Static routing
- Dynamic routing
  - Distance Vector
  - Link State



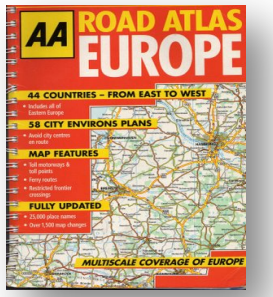
Routing. Dynamic routing

Distance Vector Protocols

- Each router tell its neighbors about its view over the network
- Routes are advertised as a vector of distance and direction.
- Routers do not have knowledge of the entire path to a destination

Link State Protocols

- Each router tells the world about its neighbors
- Routes are computed based on the network connectivity map (topological database)
- Routers have knowledge of the entire path to a destination

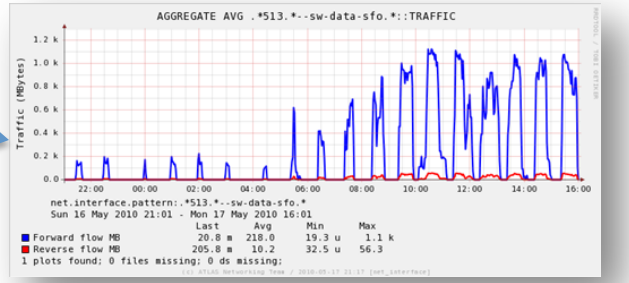


Outline

- Introduction
  - Networking basics
  - OSI Model
- Technologies and protocols
  - Ethernet
  - IP Protocol
  - Routers and routing
- Network monitoring
- Software defined networking

Network Monitoring. SNMP

- A standard protocol for managing devices on IP networks (switches, routers, computers etc);
- Exposes management data in the form of variables on the managed systems. These variables are then queried;
- Used to gather device-based or port-based statistics (traffic volume, errors, packets, discards, temperature etc);

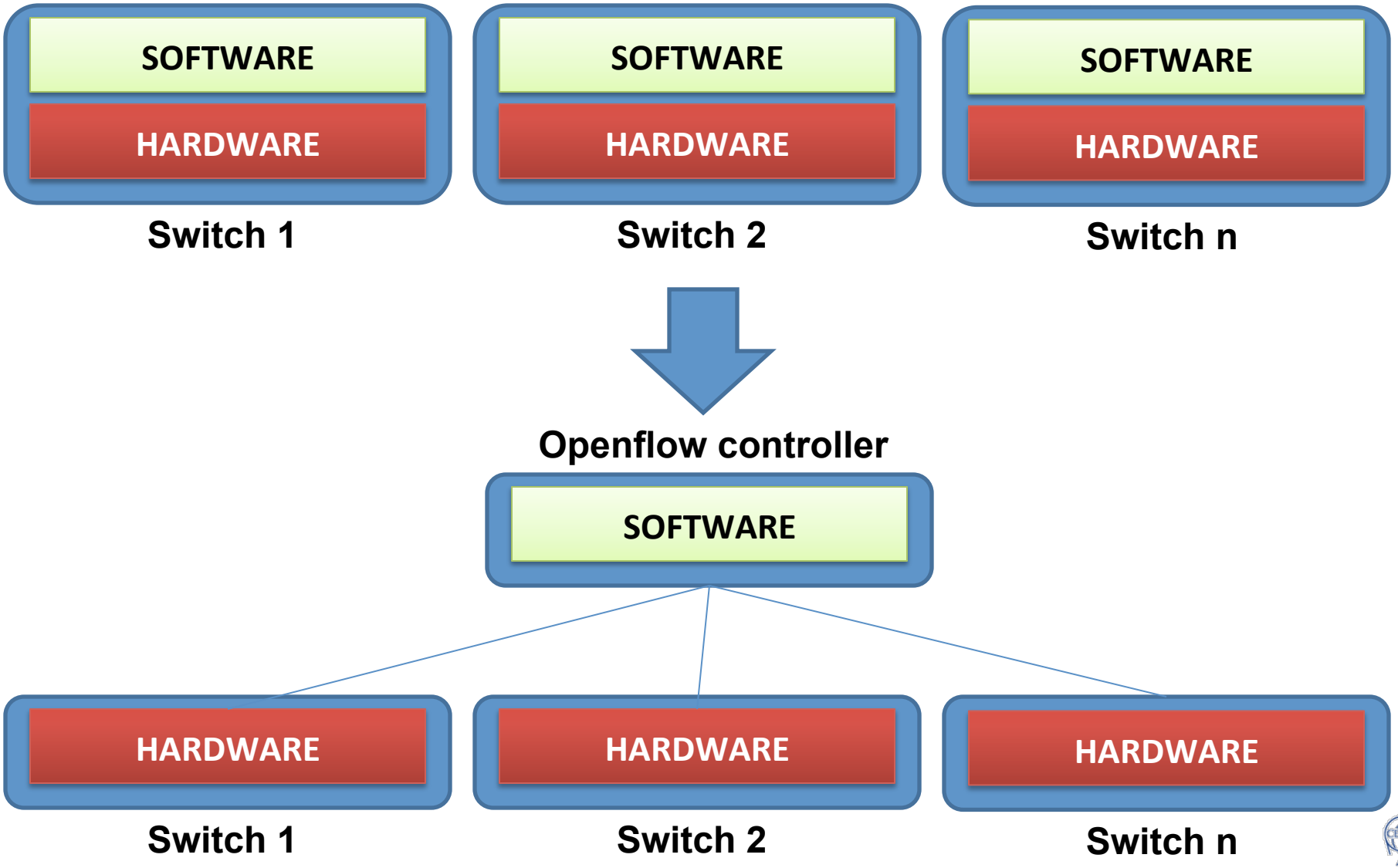


Network Monitoring. sFlow & NetFlow

- Network monitoring technology to gather flow-related statistics;
- Can track the source and destination for packets that passes through an interface;

- sFlow compute statistics based on a sampling mechanism;
- NetFlow keeps a record for every flow. If needed, it can also use sampling.

Software defined networking. OpenFlow



Software defined networking. OpenFlow

- An open protocol to remotely add/remove flow entries;
- Allows the path of network packets through a network to be determined by a software running on a separate server, called a **controller**
- The controller run a **NOS** (Network Operating System) that allows user applications to control network behavior.

- Allow researchers to run routing experiments in their network;
- Already supported on several routers (ex: HP Procurve)

The show must go on …

Data Networks

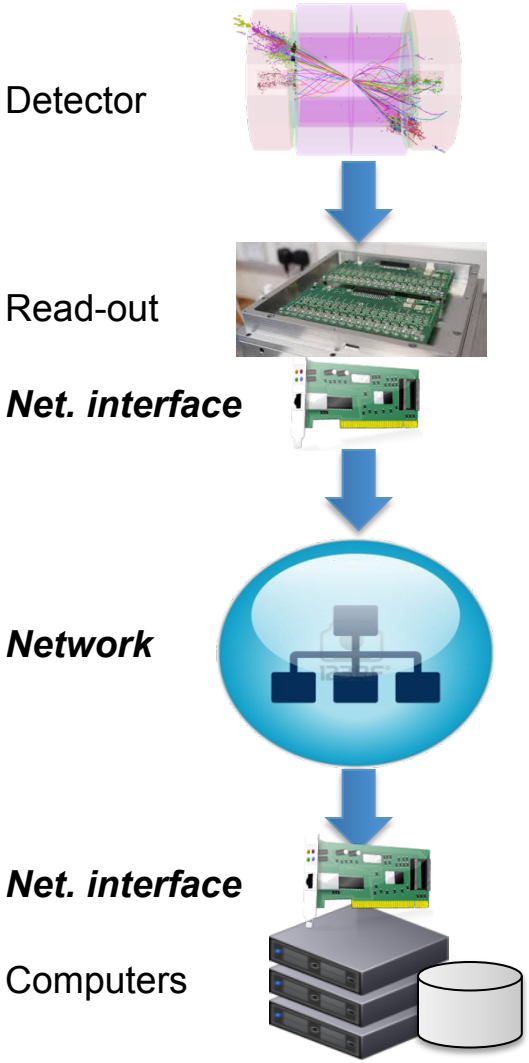
Networking for Data Acquisition

Dan Octavian Savu
&
Stefan Stancu

CERN

ISOTDAQ 2014, Budapest

Data Acquisition uses networks



- Detector
  - Measure physical phenomena
- Read-Out
  - Digitize and perform basic processing
  - Possibly data buffers
  - **Interface to network**
- **Network**
  - Connect all read-outs to analysis computers
  - Allows computers to collect data from all sources
- Computer(s)
  - **Interface to network**
  - Collect data from all sources
  - Analyze and filter data
  - Store data

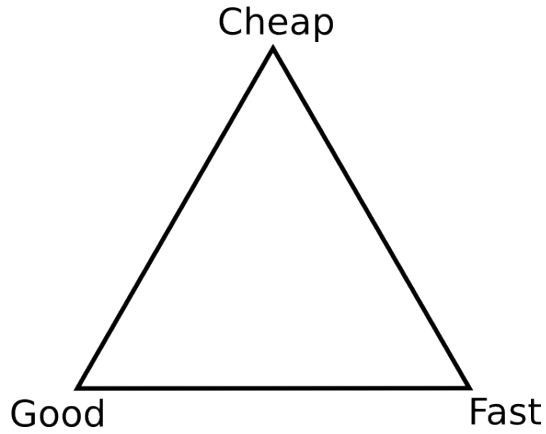
Networks for Data Acquisition

Outline

- Network technology choice
  - Ethernet
- Transport layer
  - UDP
  - TCP
- Linux networking optimizations
- DAQ Networks for large experiments
- DAQ application design: *push* vs. *pull*

Network Technology Choice

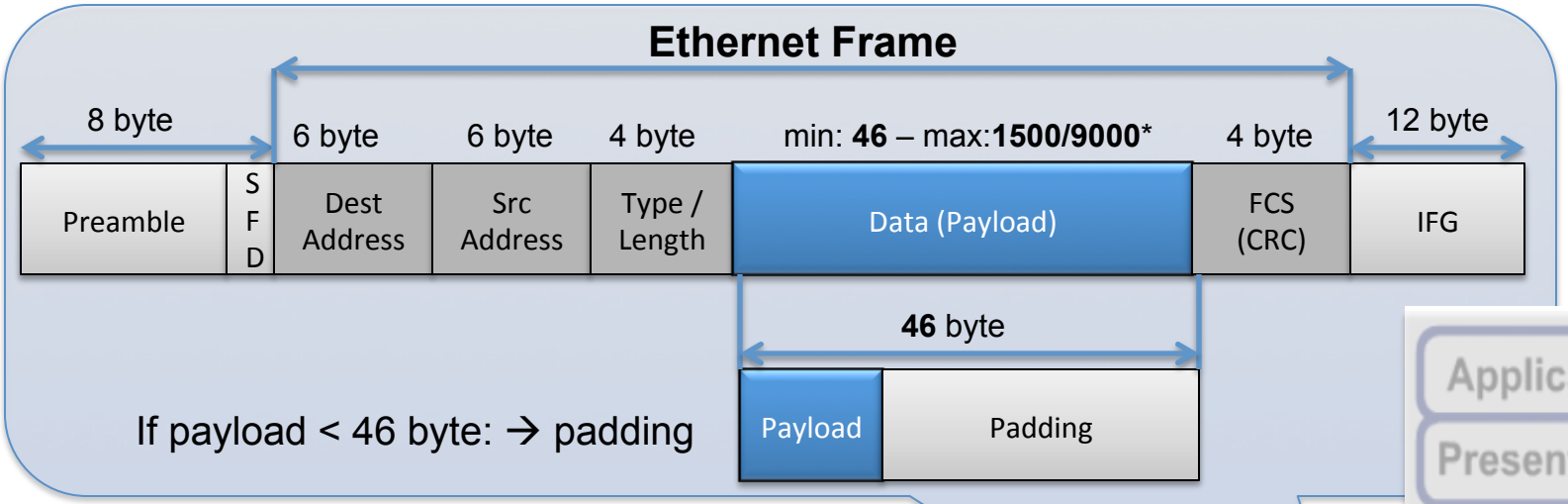
- Requirements
  - Scalability
  - Reliability
  - Low cost
  - Expected life-time
  - Provider's health

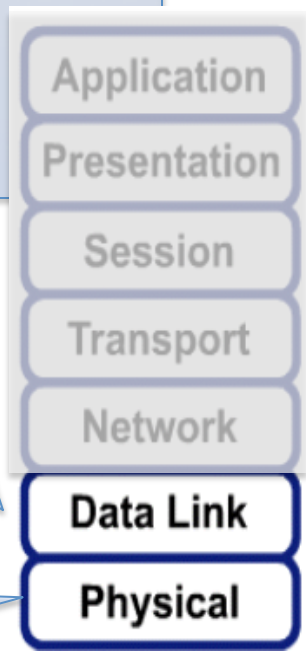


- **Ethernet** is a de-facto standard
  - OSI Layer-2
  - Largely used in the industry
  - Many providers
  - Ubiquitous
  - Layering allows to put anything on top of it

- Other technologies: Myrinet, InfiniBand

Ethernet





All flavors of media and speeds:

- … even slower but this is now history
- 100 Mbit/s: copper (UTP), fiber
- 1 Gbit/s: copper (UTP), fiber
- 10 Gbit/s: fiber, copper (twinax, UTP)
- 40 Gbit/s: fiber
- 100 Gbit/s: fiber

Networks for Data Acquisition

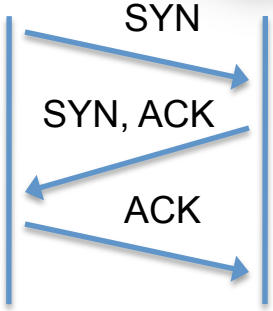
Outline

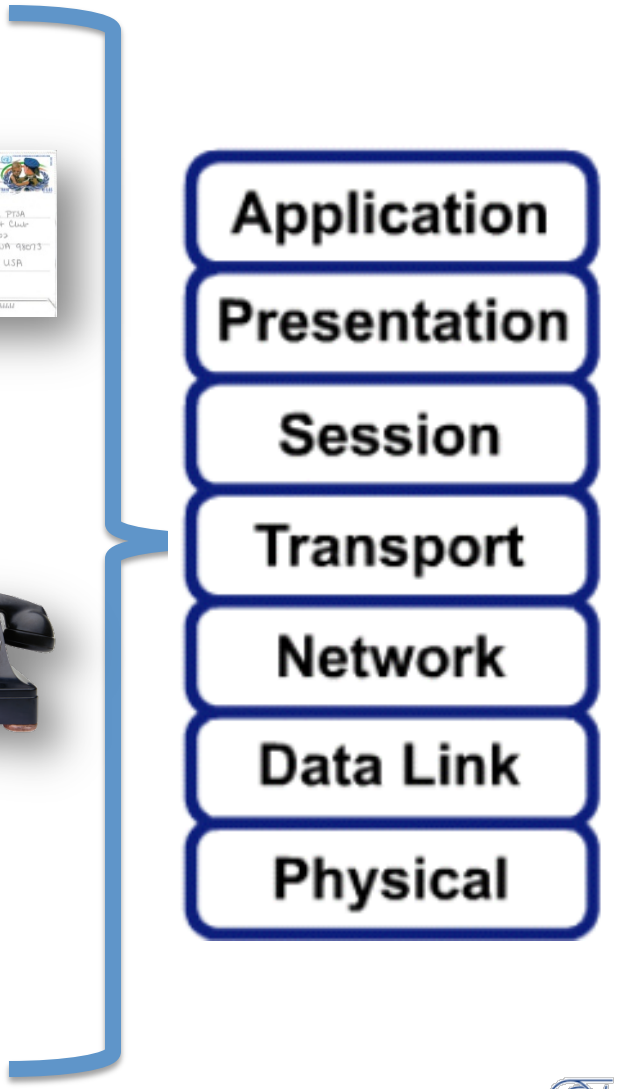
- Network technology choice
  - Ethernet
- Transport layer
  - UDP
  - TCP
- Linux networking optimizations
- DAQ Networks for large experiments
- DAQ application design: *push* vs. *pull*

Major Transport Protocols: TCP and UDP

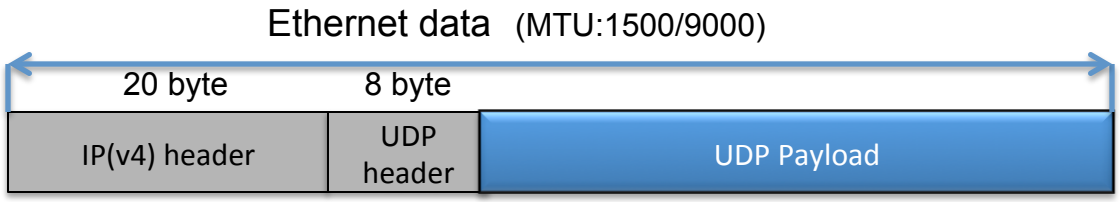
- Unreliable Datagram Protocol (UDP)
  - Unreliable but simple
  - Connectionless
  - RFC 768
    - http://tools.ietf.org/html/rfc768

- Transport Control Protocol (TCP)
  - Connection oriented protocol
  - Flow control
  - Lossless
  - RFC 793
    - http://tools.ietf.org/html/rfc793



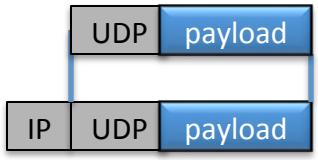


UDP/IP (over Ethernet)

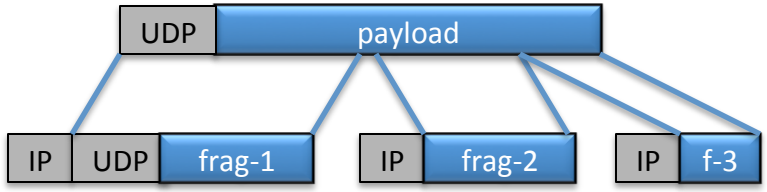


- UDP immediately sends messages
  - Message size decided by application
  - UDP is not MTU aware
  - IP will take care of putting data in MTUs

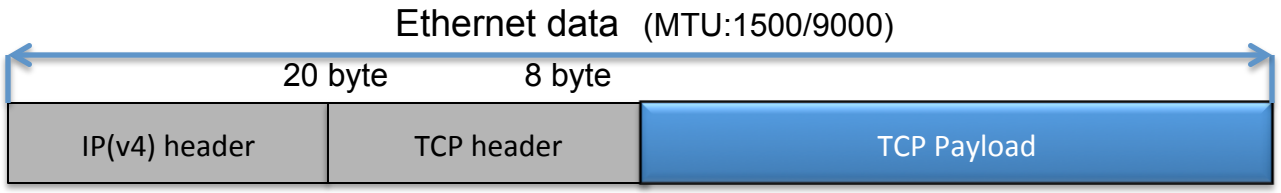
- Payload < MTU
  - IP: one frame



- Payload > MTU
  - **IP** will fragment: multiple frames

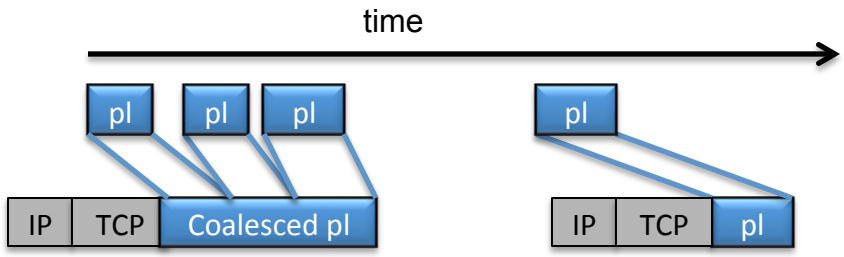


TCP/IP (over Ethernet)

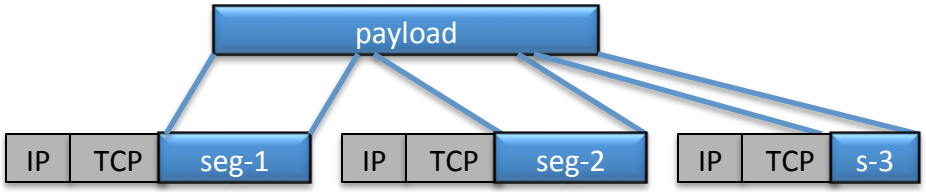


- TCP
  - Is MTU aware
  - Nagle Algorithm
    - Buffers data for a short while before sending it
    - Message size optimized by TCP

- Payload < MTU
  - Nagle may coalesce data for you



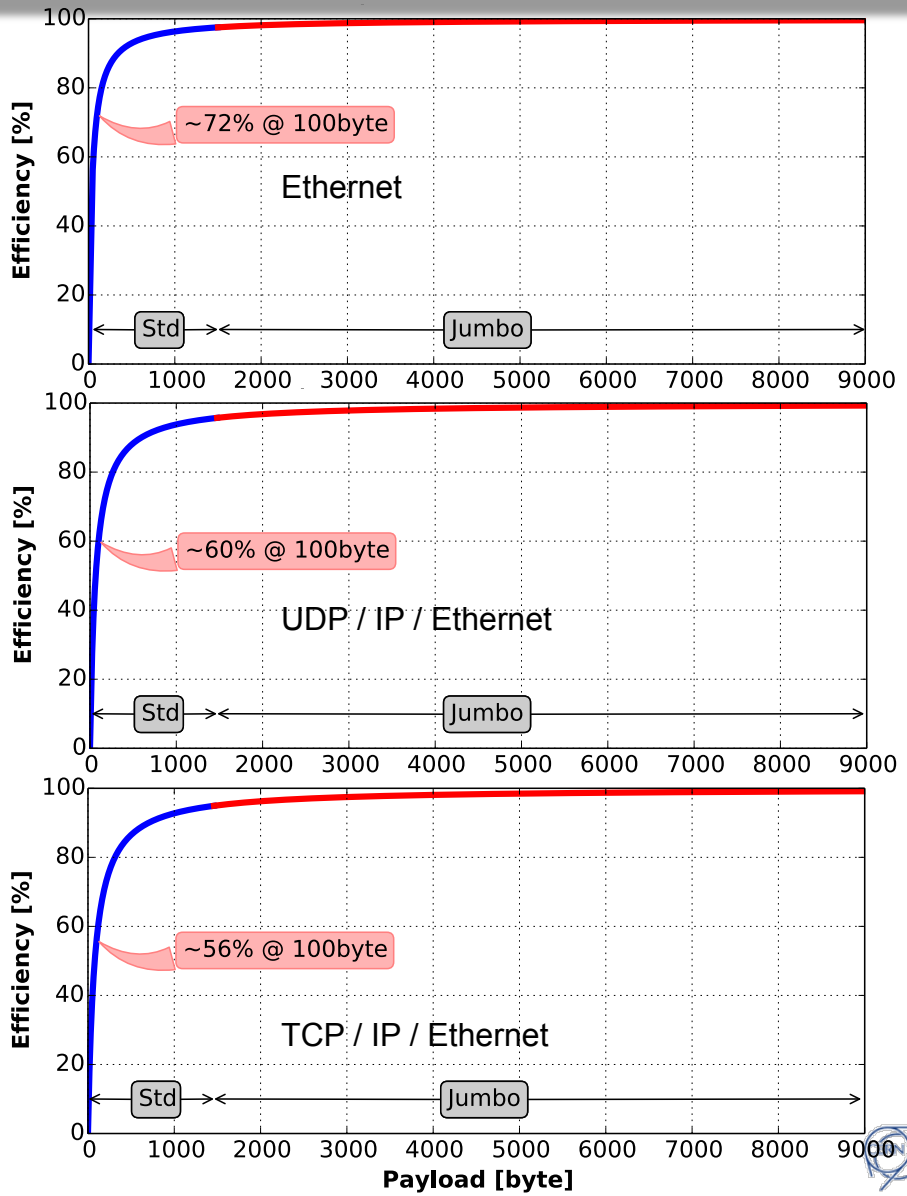
- Payload > MTU
  - **TCP** will segment: multiple frames
  - No IP fragmentation



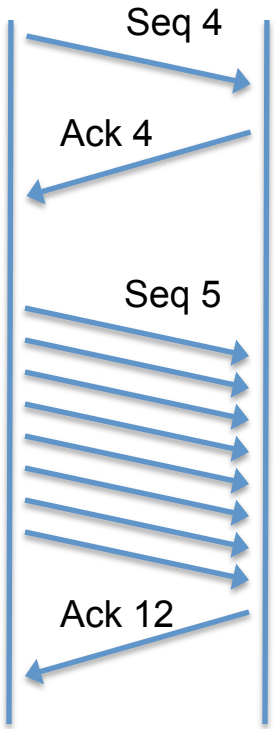
Encapsulation – Efficiency

$$Efficiency = \frac{Payload}{Payload + Overhead}$$

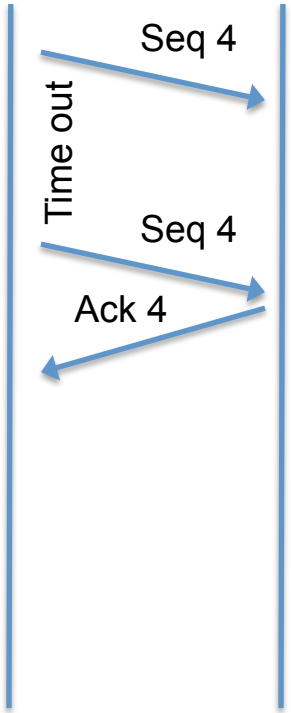
Encapsulation	Efficiency (100 byte)	Efficiency (1 byte)
Ethernet	72%	1.2%
UDP/IP/Eth	60%	1.2%
TCP/IP/Eth (Nagle algo)	> 56%	> 1.2%



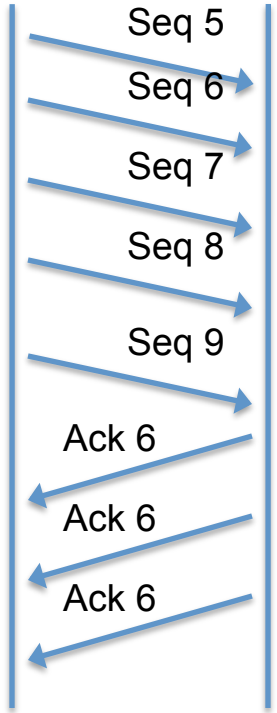
TCP: How it works



Normal transmission



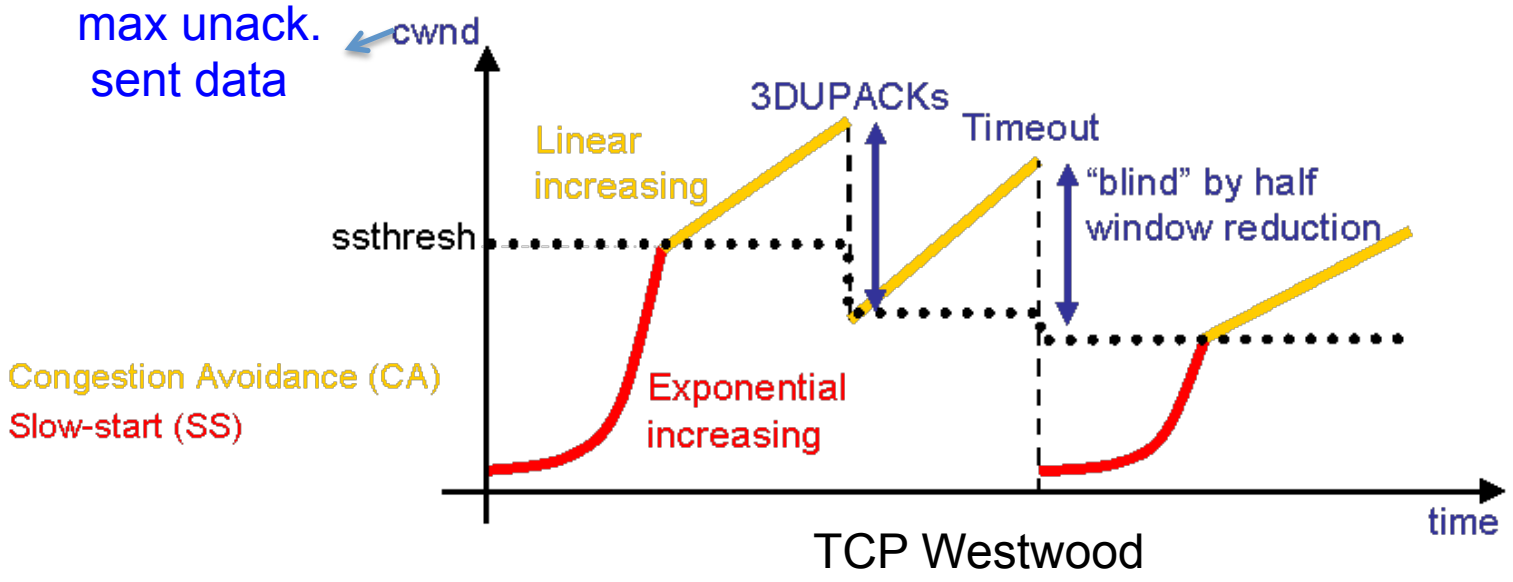
Packet never acknowledged
Time out



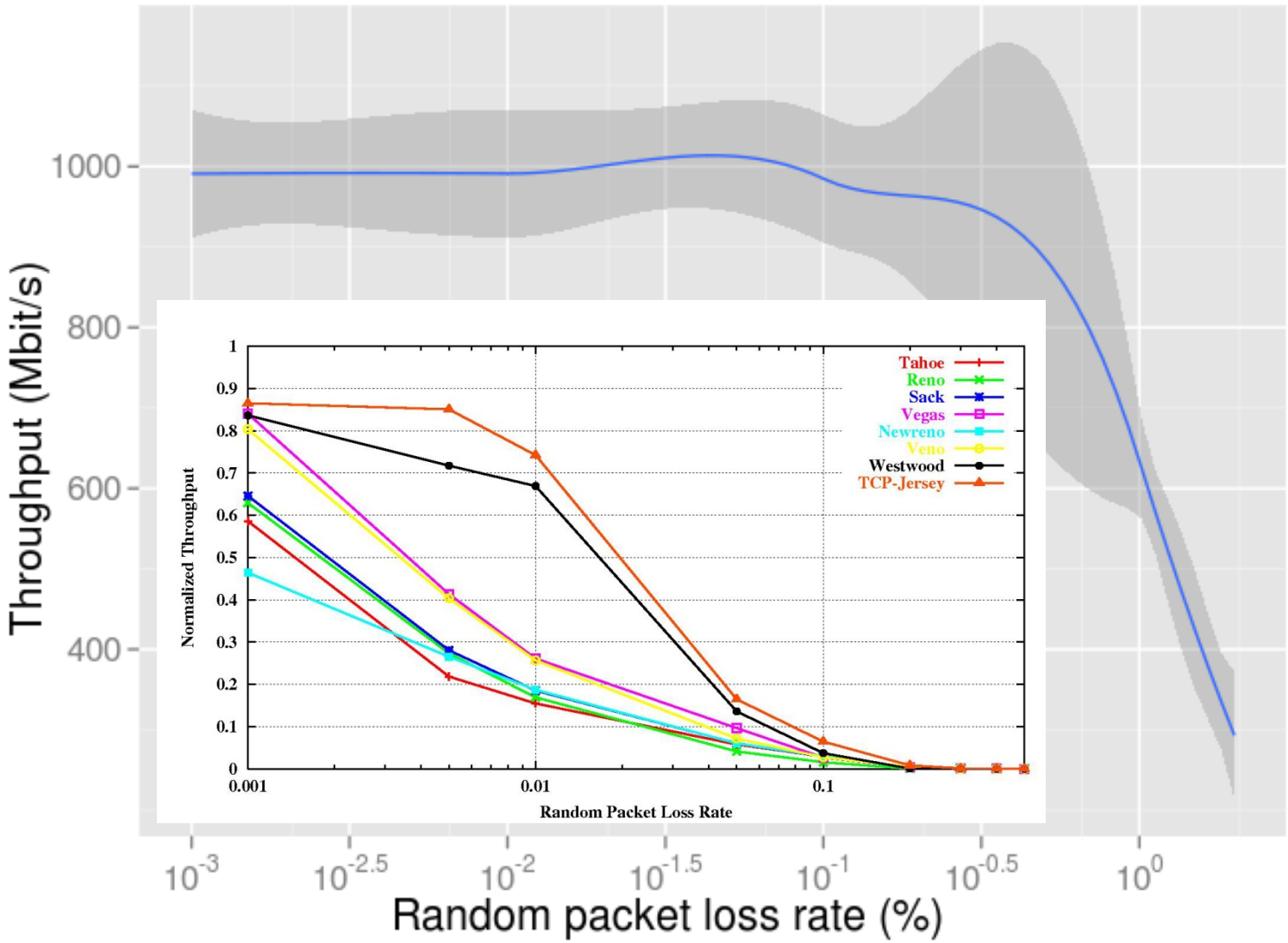
Packet lost
in a sequence
3 duplicated ACKS

TCP: Reliability Vs Performance

- Cause of packet loss:
  - Congestion
  - Discards and errors
  - Faulty hardware or drivers
- Congestion is detected
  - If time out happens while no ACK received
  - If multiple duplicate ACKs received
- Congestion avoidance: adapt data rate to the traffic conditions



TCP Performance with Packet loss



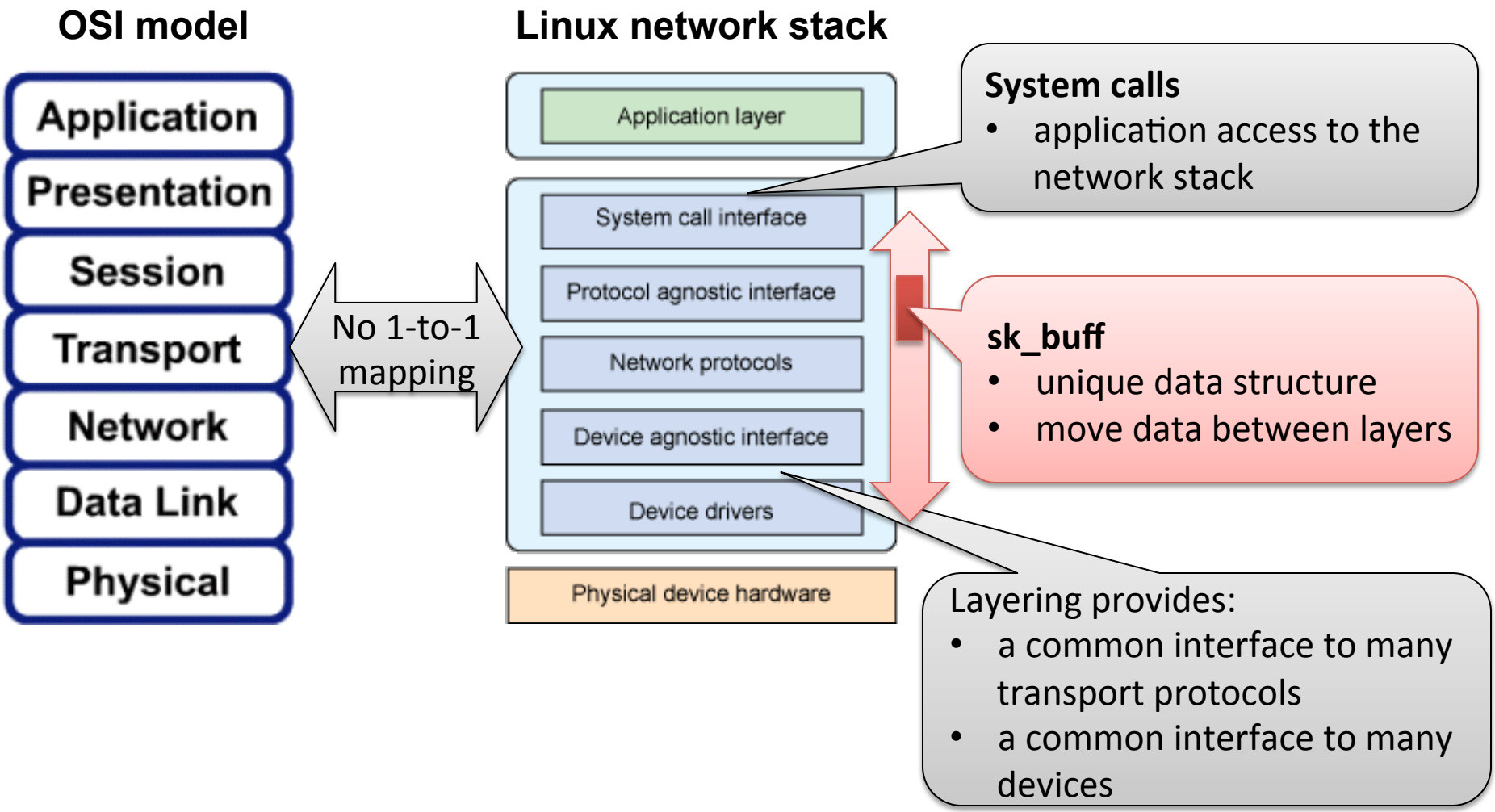
MTU = 9000 bytes
Tools: netem and iperf
1 GbE link
Binary Increase Congestion control algorithm

Networks for Data Acquisition

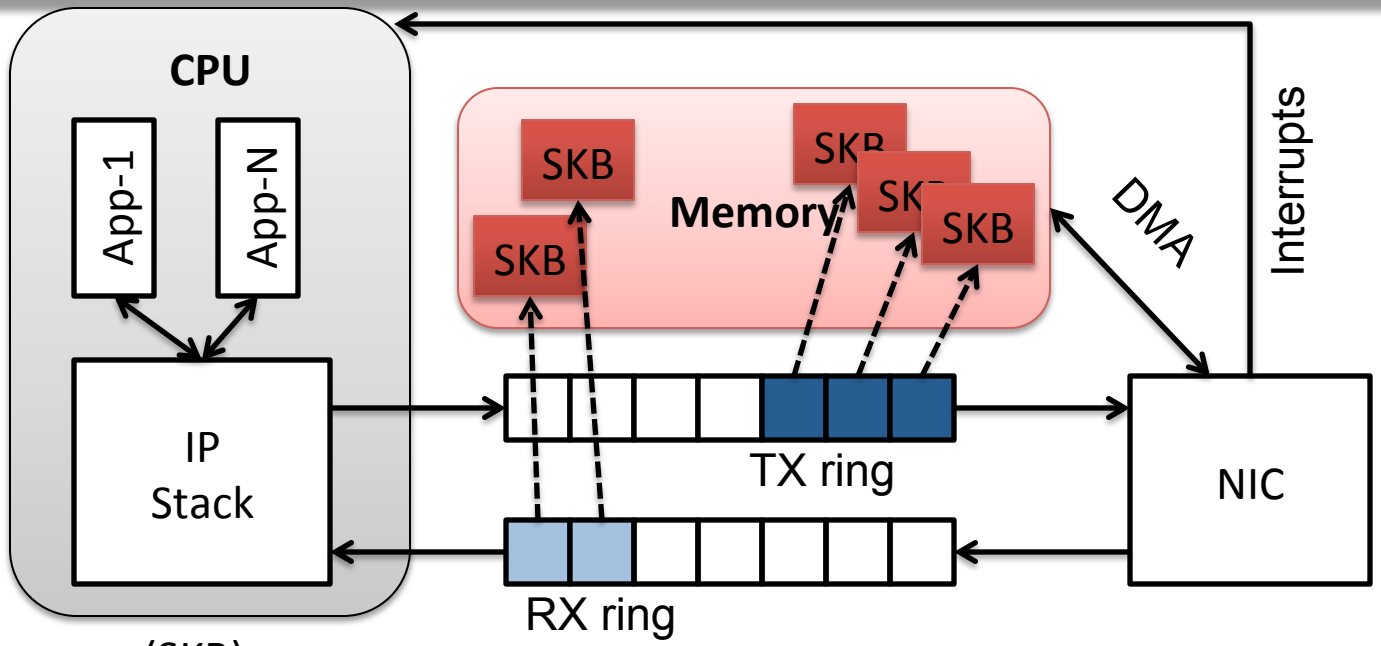
Outline

- Network technology choice
  - Ethernet
- Transport layer
  - UDP
  - TCP
- Linux network stack & optimizations
- DAQ Networks for large experiments
- DAQ application design: *push* vs. *pull*

From theory to practice



Kernel – NIC interaction



Send

- Data in memory (SKB)
- Descriptor to TX ring
- NIC fetches data via DMA
- NIC **interrupts** when finished sending

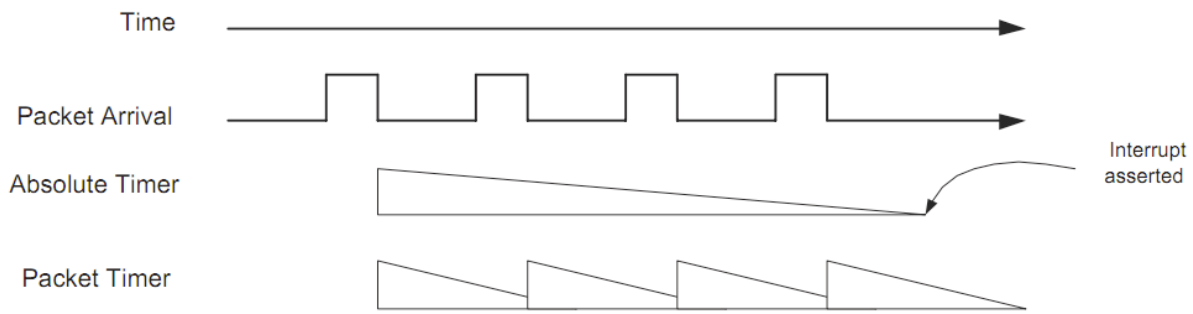
Receive

- NIC puts data in memory (SKB) via DMA
- NIC puts descriptor in RX ring
- NIC **interrupts**
- CPU fetches the SKB and frees up the RX ring descriptor

Interrupt coalescing

- Hardware interrupt has a cost
  - Context switch of a CPU
    - Saving and loading registers and memory maps, updating various tables and list
  - Happens every time an Ethernet frame is received ?
    - 1538 bytes -> 12304 bits -> 1 frame every 1.23 µs @ 10 GbE

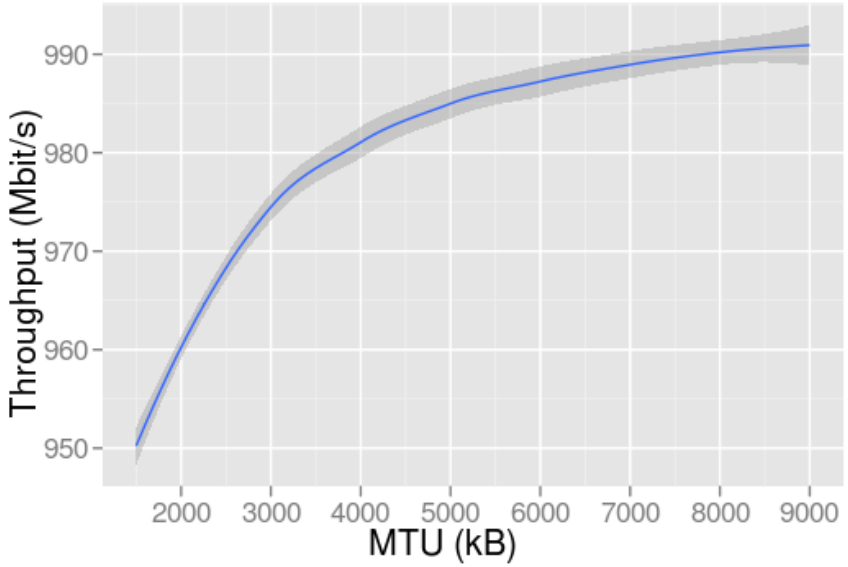
- Lower the rate with interrupt coalescing
  - 1 interrupt for several frames
  - Do not add too much latency in case of low traffic

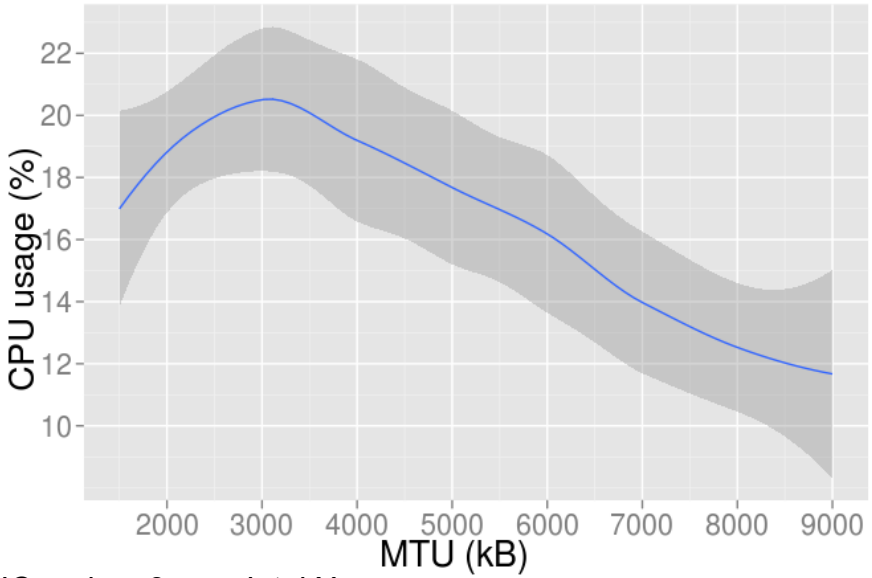


  - Careful with the ring buffer size
    - Packets are discarded if the buffer is full

Jumbo Frames

- Improve max throughput (encapsulation overhead)
  - 94% @ 1500 MTU
  - 99% @ 9000 MTU
- **Reduce the frame rate**
  - Lower interrupt rate
  - Less data dis/re – assembling for the CPU



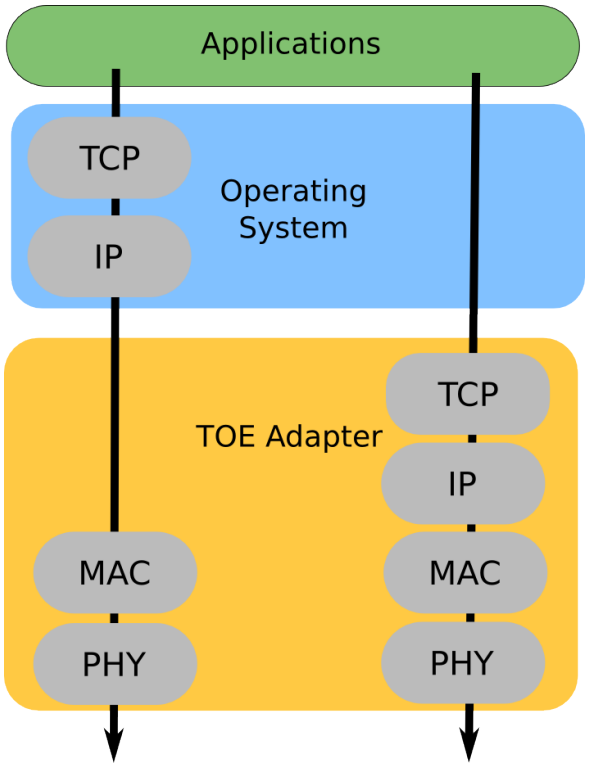


Tests performed on a Broadcom NIC and an 8 core Intel Xeon processor

Considering efficiency at all stages

- Higher Layer Processing consists of numerous tasks:
  - Checksum calculations
  - Data copies
  - Header stripping or adding
  - Protocol logic implementation
  - Reordering
  - Reassembling
  - ….
- High bandwidth impact on CPU consumption
  - 100% of one 2.33 GHz core for a bandwidth of about 5 Gb/s (without any tuning ☺)
- Trying to outsource these to hardware controllers

NIC Offloading



TOE: export processing to hardware controllers

- Purpose: free host CPU cycles
- TCP Offload Engine: TCP/IP stack processed by the network device
  - Checksum computing
  - Transport protocol segmentation
- A TOE capable device will offer the OS a much larger MTU (SKB size)
  - TSO = TCP Segmentation Offload (send)
    - the NIC takes care of segmenting the large SKB
  - LRO = Large Receive Offload (receive)
    - the NIC assembles data from multiple frames/ segments into a large SKB

Basic optimizations for DAQ

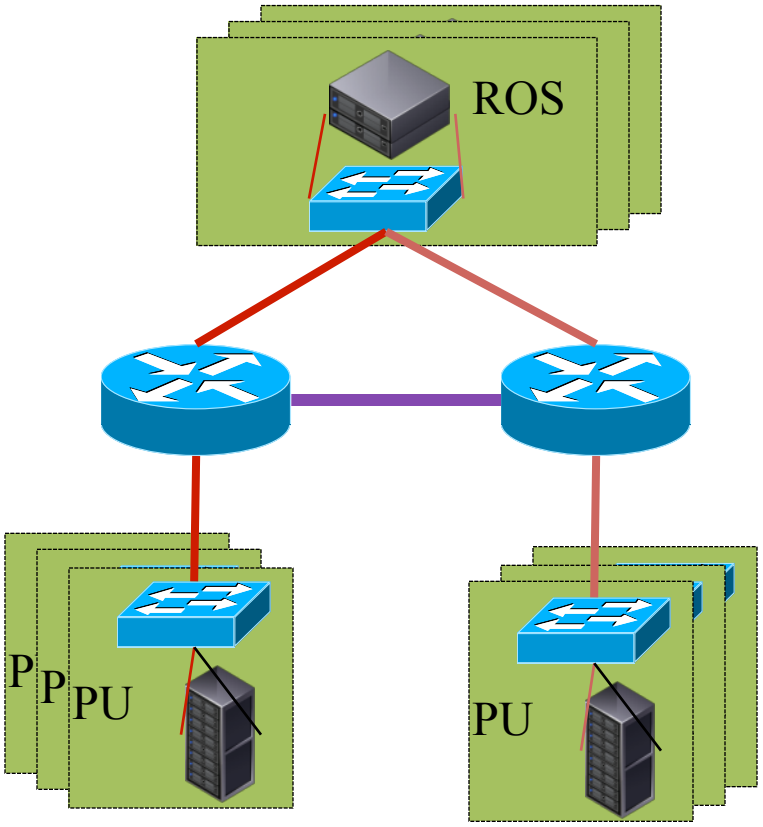
- Hosts
  - Mainly for reception side
    - Surprisingly at first side, *reception* is the more resource consuming side
  - Provide large kernel buffers and large socket buffer for the application
    - Machines can easily cope
  - Tune IRQ moderation
- Network devices
  - Usually clean network with an unidirectional dataflow
  - Enable jumbo frames on all port to improve bandwidth
  - Maximize buffers
    - Packet loss has a big impact on performance (see previous TCP slide)
    - Latency is a 2nd order concern for DAQ

Networks for Data Acquisition

Outline

- Network technology choice
  - Ethernet
- Transport layer
  - UDP
  - TCP
- Linux network stack & optimizations
- DAQ Networks for large experiments
- DAQ application design: *push* vs. *pull*

DAQ Network for a large experiment



ATLAS DAQ Network

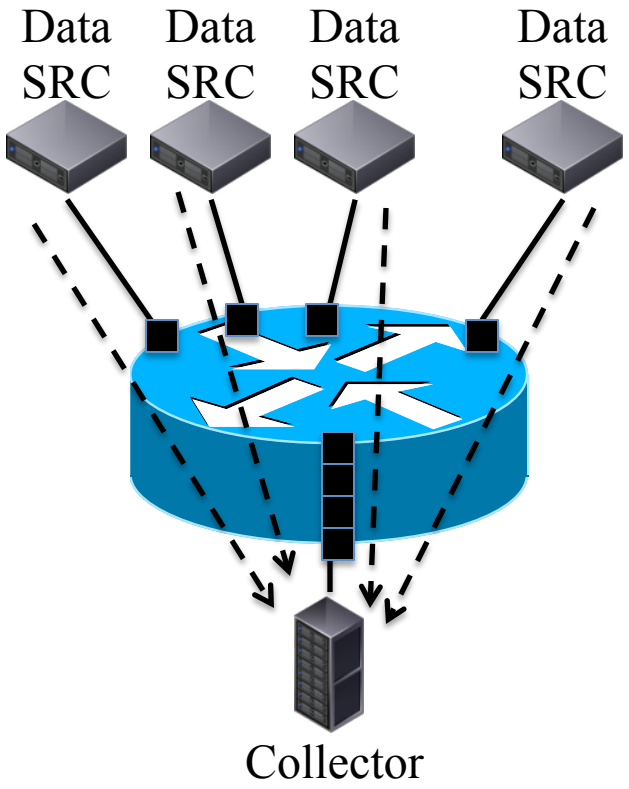
- LHC DAQ systems use O(1000) nodes
  - too large for a single device
- Typical multi-layer architecture
  - Aggregation layer
  - Core layer
  - De-aggregation / fan-out / edge
- Simple, reliable and fast
  - static routing
  - layer 2 switching
  - QoS (high priority for important traffic)

Networks for Data Acquisition

Outline

- Network technology choice
  - Ethernet
- Transport layer
  - UDP
  - TCP
- Linux network stack & optimizations
- DAQ Networks for large experiments
- DAQ application design: *push* vs. *pull*

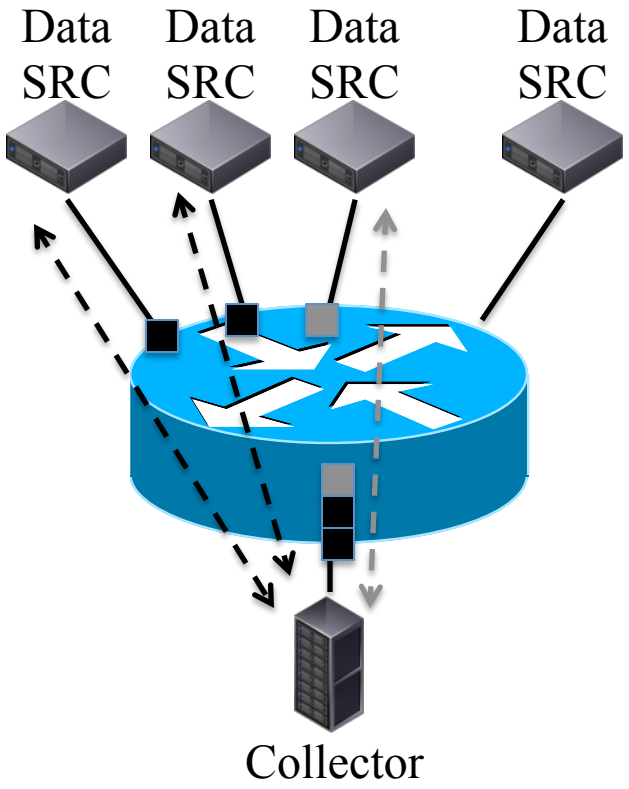
DAQ – push design



Push

- Data SRCs simultaneously send data to a collector
- Funnel-in effect on the switch
  - Packets need to be buffered before being sent to the Collector
  - The more sources, the worse
- Advantages:
  - Simple design of the data sources
- Disadvantages
  - Rely on network buffers for not losing data
  - Collector must cope with the rate

DAQ – pull design



Pull

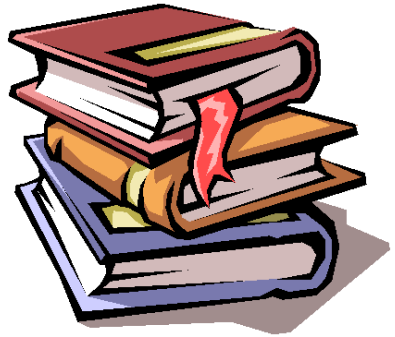
- Data SRC buffer data and provide it on request
- Controlled funnel-in effect on the switch
  - Collector can limit the number of outstanding requests
  - Not affected by the number of sources
- Advantages:
  - Full control of network traffic
  - Collector asks as much as it can handle
  - Collector can slow down in case of loss detection
- Disadvantages
  - Data sources complexity:
    - Buffering
    - Request-reply protocol implementation

Conclusion

- Networking basics
  - OSI Model is your friend !
  - Ethernet is used to connect hosts together
  - IP is used to connect networks together
  - SNMP and sFlow/NetFlow are used to monitor the network
- DAQ networks
  - Only scratched the surface of a few standard or new technologies
  - Many other protocols exist:
    - less known than TCP and UDP
    - very interesting for DAQ
  - Numerous other optimizations
- New investigation
  - SDN: you can build your custom DAQ network
    - Any topology (don't care about loops anymore, like with standard Ethernet)
    - Engineer your flows as you want
    - Main limitation: OpenFlow rules table size in HW devices
  - 40/100 GbE
  - InfiniBand

References

- **Wikipedia**
- **IETF RFCs**
- **« man » pages**
- **Conference proceedings and journals**



A few noticeable:

1. INTEL Corp. Interrupt Moderation Using Intel® GbE Controllers. http://download.intel.com/design/network/applnots/ap450.pdf, 2007.

2. Wenji Wu. The Performance Analysis of Linux Networking Packet Receiving http://lss.fnal.gov/archive/2006/pub/fermilab-pub-06-406-cd.pdf

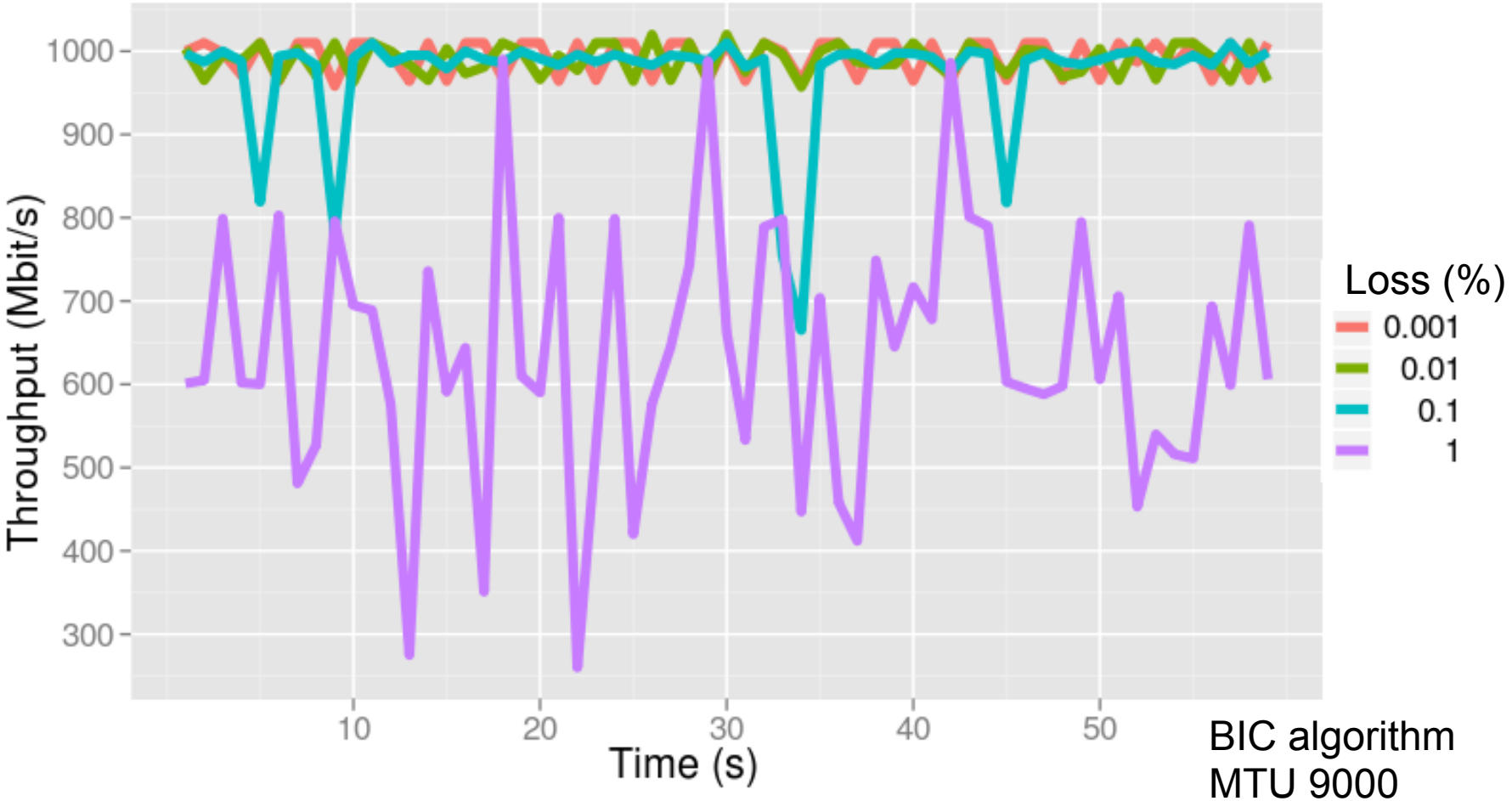
3. Sequence diagrams for TCP/IP stack and many protocols http://www.eventhelix.com/RealtimeMantra/Networking/

4. 10 Gigabit Ethernet Association http://www.10gea.org/tcp-ip-offload-engine-toe.htm

5. Binary Increase Congestion Control for Fast, Long Distance Networks http://netsrv.csc.ncsu.edu/export/bitcp.pdf

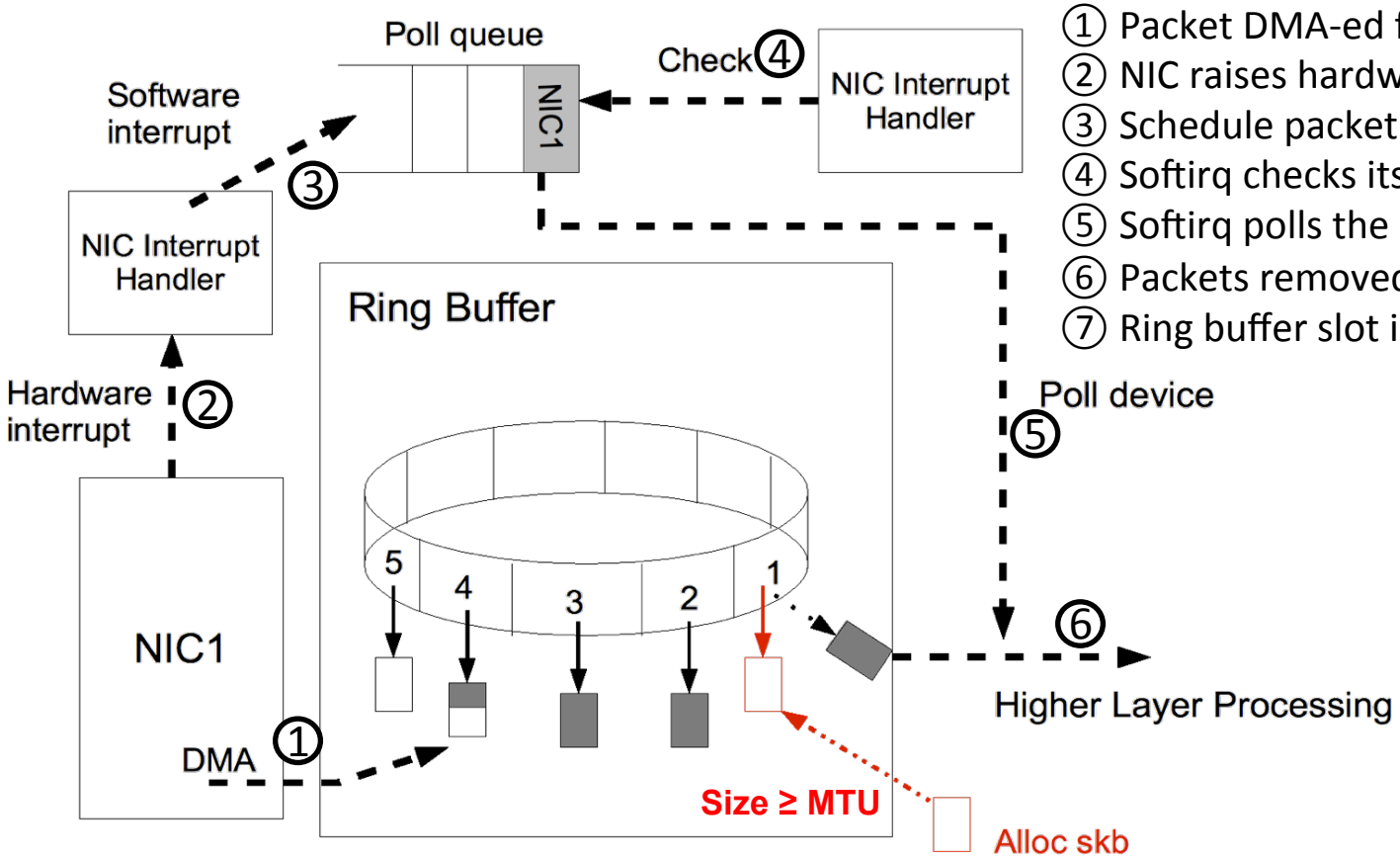
Bck

TCP Throughput over time



BIC algorithm
MTU 9000
1 GbE link
Netem and iperf

Eff-host: From the device to the Kernel



① Packet DMA-ed from NIC to Ring Buffer
② NIC raises hardware interrupt
③ Schedule packet receiving Softirq
④ Softirq checks its CPU's device poll-queue
⑤ Softirq polls the NIC's ring buffer
⑥ Packets removed from ring buffer
⑦ Ring buffer slot is reinitialized

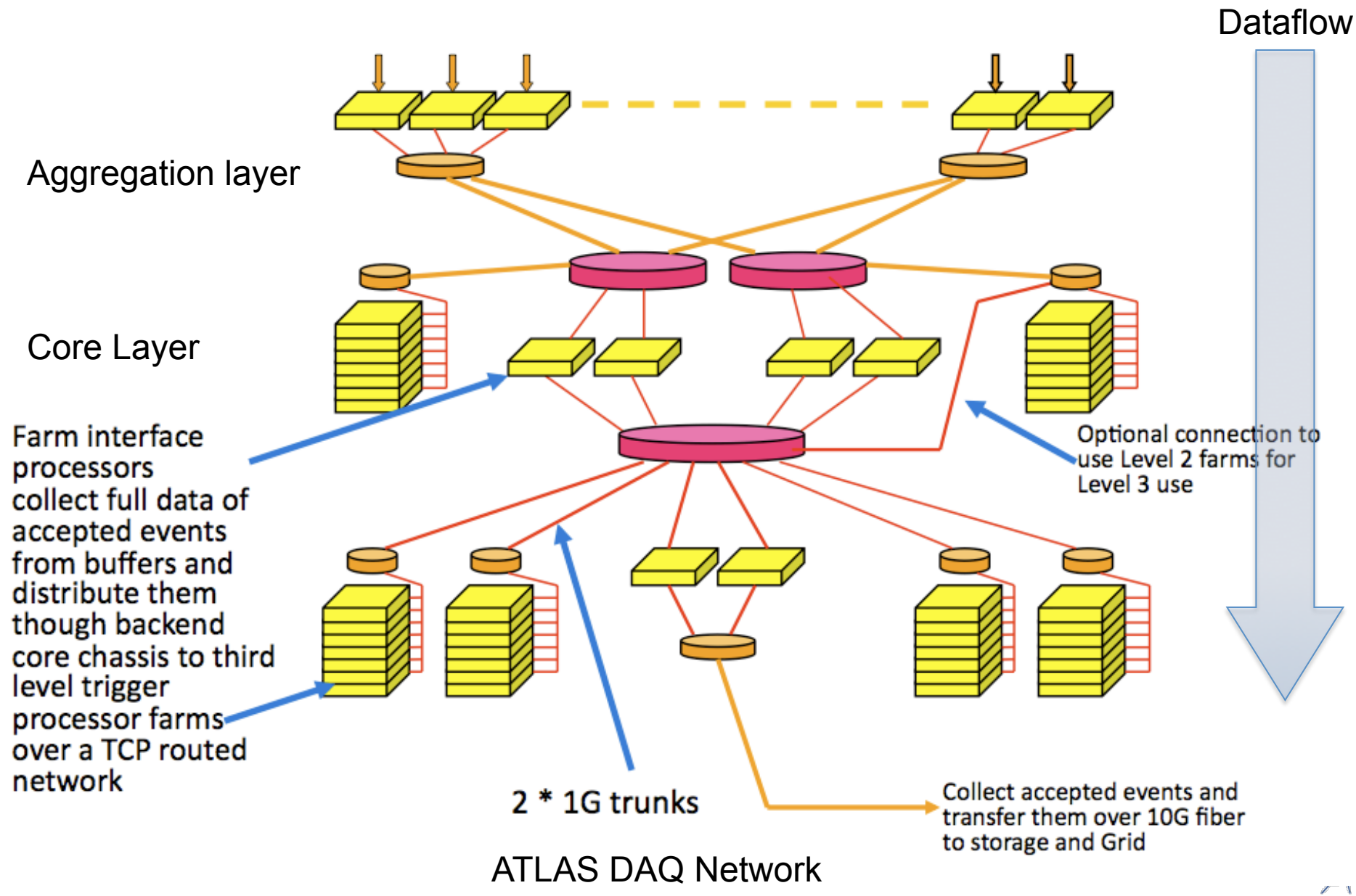
Network devices for DAQ

Core router

Edge switch

- LHC data acquisition systems uses O(1000) ports -> too large for a single device
- Typical architecture
  - Aggregation layer
  - Core layer
  - De-aggregation / fan-out / edge
- Top-down dataflow relies on static routing and layer 2 switching

Overview


Dataflow
Aggregation layer
Core Layer
Farm interface processors collect full data of accepted events from buffers and distribute them though backend core chassis to third level trigger processor farms over a TCP routed network
Optional connection to use Level 2 farms for Level 3 use
2 * 1G trunks
Collect accepted events and transfer them over 10G fiber to storage and Grid
ATLAS DAQ Network