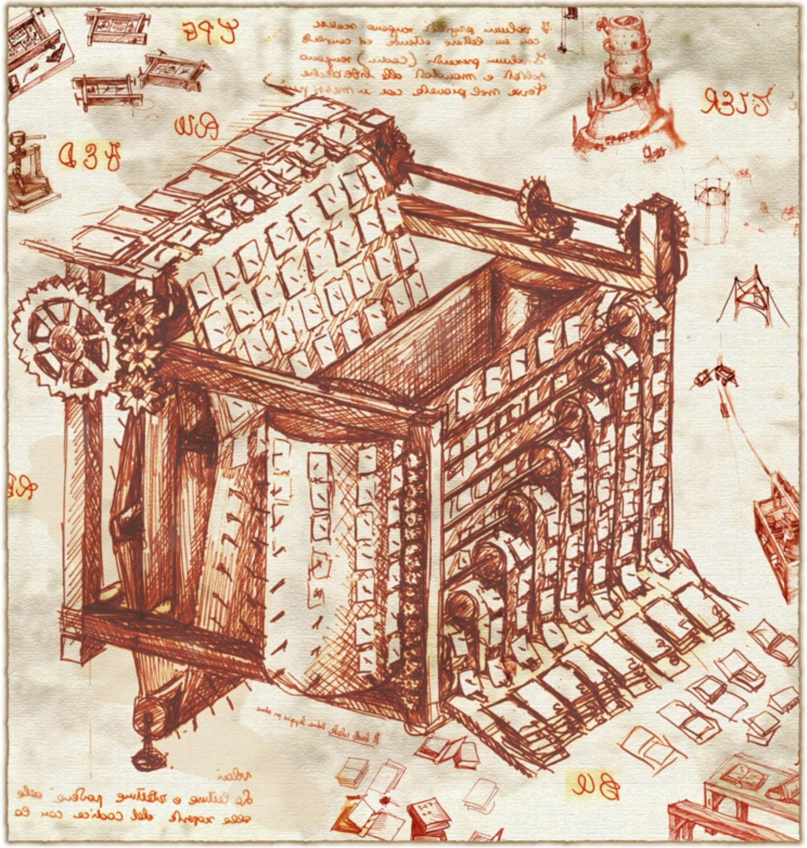
Introduction to Networking for Data AcQuisition (DAQ) systems

Petr Žejdl

CERN CMS-DAQ group

19-28 June 2024
Hefei, China



Sergio Cittolin © CERN

ISOTDAQ

International School of Trigger and Data Acquisition

Outline

- Examples of Computer Networks
- Computer Network Basics
  - ISO/OSI Network Model
  - Important protocols of the Internet Protocol Suite (TCP/IP)
  - Performance Considerations
- Real DAQ System of CMS Experiment

Examples of Computer Networks

- Remote meeting, conference, lecture?

Source: www.flaticon.com

- Straight line distance CERN – Hefei is over 8800 km
  - Many **inter**connected computer **net**works made the connection possible!
  - **Internet**

Examples of Computer Networks

Source: https://sekkeidigitalgroup.com/best-chinese-streaming-platforms/

Are you watching streaming services?

Or live television channels over IPTV?

Examples of Computer Networks

Source: https://sekkeidigitalgroup.com/best-chinese-streaming-platforms/

Are you watching streaming services?

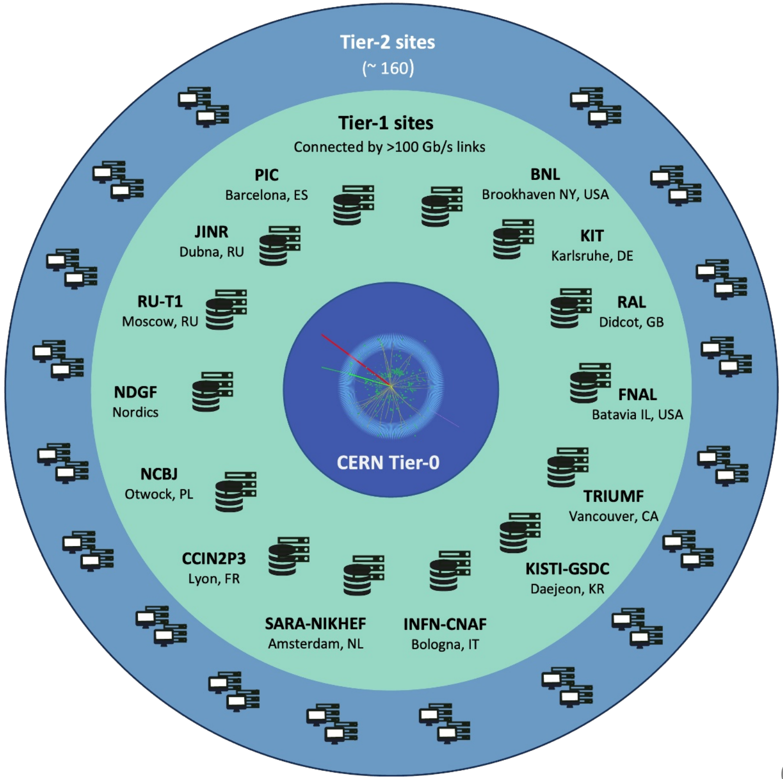
On a mobile phone over **wireless network**?

Or live television channels over IPTV?

Examples of Computer Networks

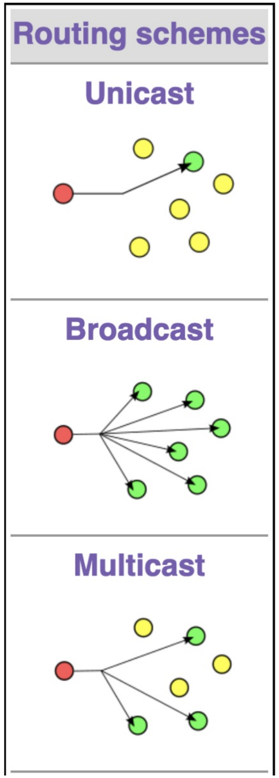
- **Worldwide LHC Computing Grid (WLCG)**
- Content is distributed and stored in 170 sites in 42 countries for processing in WLCG
- Global computing infrastructure: 1 milion computer cores, 2 exabytes of storage
- https://wlcg-public.web.cern.ch/

Servers racks of WLCG in CERN



Typical network communication patterns

- Unicast: one-to-one delivery
  - One stream per every destination/client
  - Dominant type of communication, often used by video streaming services
  - For two clients – two streams are used (double bandwidth)
- Broadcast: one-to-all delivery
  - Used by some network protocols (DHCP, ARP)
- Multicast: one-to-many delivery
  - One stream from the source, delivered to many destinations/clients
  - Used by IPTV in buildings



Network bandwidth considerations

- For broadcast or multicast delivery the network devices (routers, switches) automatically replicates packets as needed – saves bandwidth at the source/server

Source: https://en.wikipedia.org/wiki/Routing#Delivery_schemes

Commonalities, DAQ Systems

- **Streaming services**
  - Have content source stored on disks that need to be delivered asynchronously to many clients over **Content Delivery Networks (CDN)**
- **Worldwide LHC Computing Grid (WLCG)**
  - Example of WAN (Wide-area network) connecting 170 sites in 42 countries
  - Tiers of WLCG: https://wlcg-public.web.cern.ch/tiers
- **Data Acquisition Systems**
  - Solving the opposite problem compared to streaming services: “Many-to-one delivery”
  - Receive content (event) from detectors as fragments split over multiple links O(1000)
  - Event fragments have to be concatenated (event building) into events, filtered (online processing), and stored on disks/tapes in CERN tier-0 for further distribution and offline processing

DAQ Systems are based on commercially available technologies (COTS), similar to those used by the big players (Amazon, Google, Netflix, iQIYI, YooKu...)

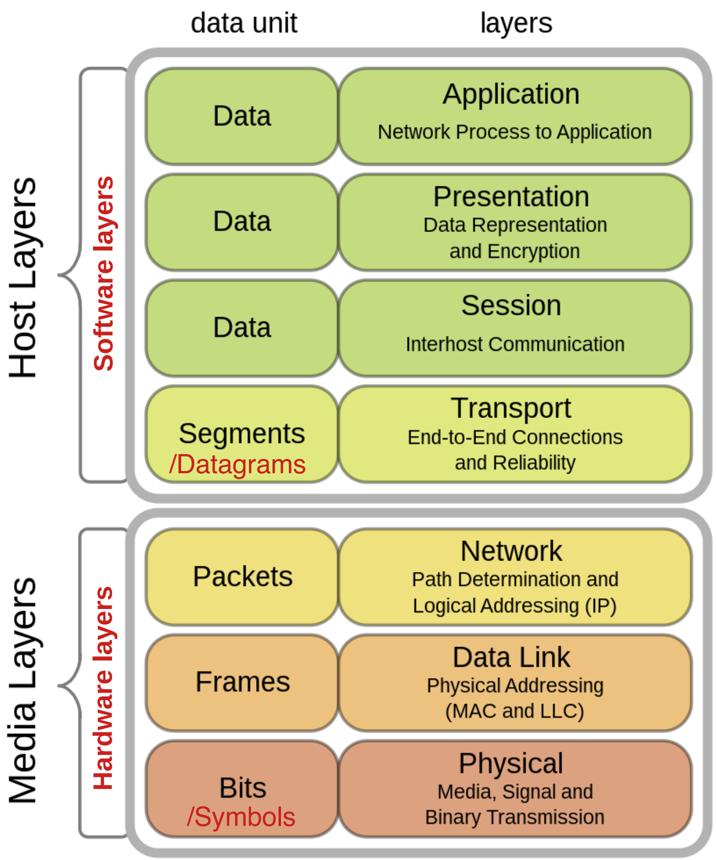
Computer Network Basics

ISO/OSI Model

Computer networks are highly complex system.

The **OSI (Open Systems Interconnection)** model, developed in the 1980s, breaks this complexity into layers.

- 7 layers, each with specific functions and protocols
- Layers interact with the ones directly above and below, adding headers or footers to data (data encapsulation)
- Published in 1984 as ISO/IEC 7498-1 standard
- It is a conceptual model, not a direct implementation*.
- Many network technologies adhere to this model with some modifications, e.g. Internet Protocol Suite (TCP/IP)

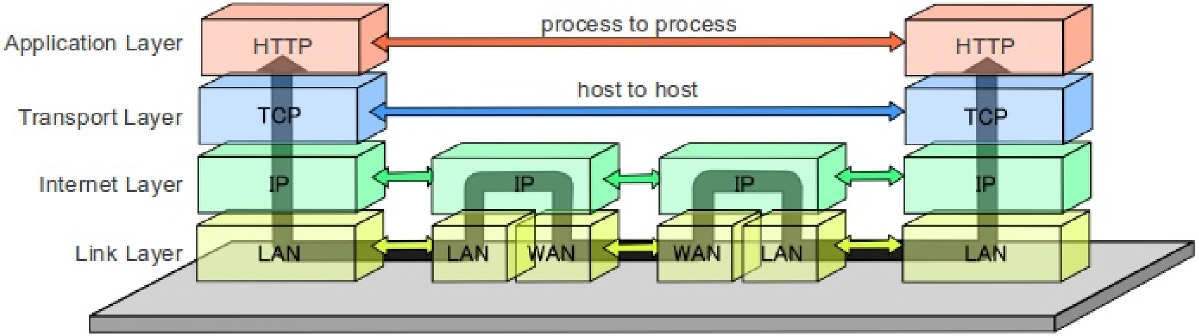


* OSI Protocols

Internet Protocol Suite (TCP/IP) in 4 Layers

- **Link:** communication within a single network segment (LAN)
  - Local Area Network (LAN) Protocols:
    - IEEE 802.3 Ethernet
    - IEEE 802.11 WiFi
  - Link Layer Protocols: Address Resolution Protocol (ARP)
- **Internet:** provides communication between independent networks
  - IPv4, IPv6
- **Transport:** host-to-host communication
  - TCP, UDP
- **Application**:
  - Process-to-process data exchange for applications
  - e.g. HTTP

Data Flow of the Internet Protocol Suite



https://upload.wikimedia.org/wikipedia/commons/b/b2/Data_Flow_of_the_Internet_Protocol_Suite.PNG

Data Link + Physical Layer

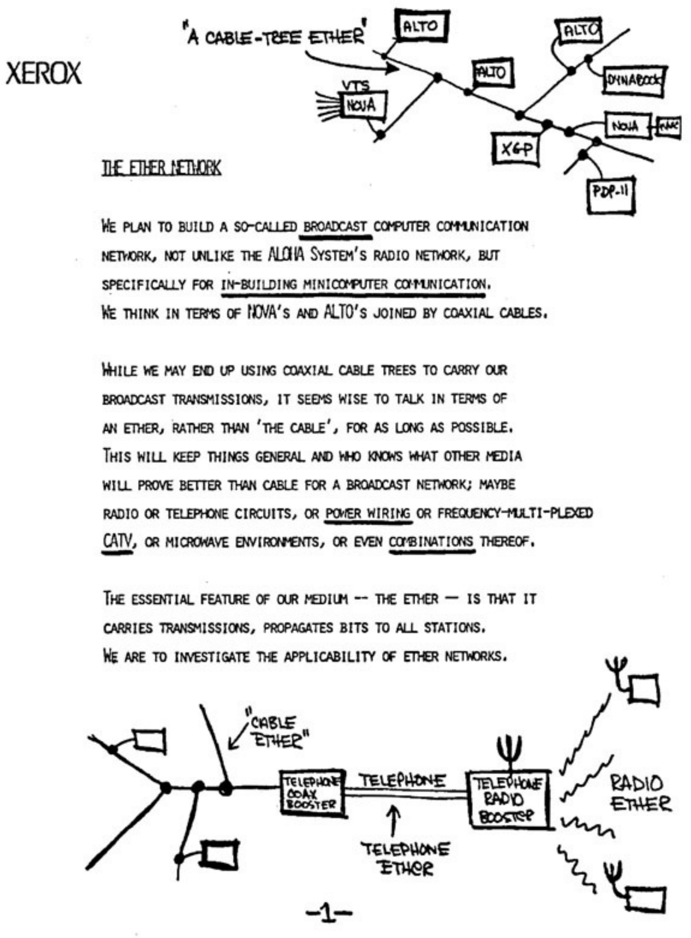
Link Layer in the Internet Protocol Suite

IEEE 802.3 Ethernet

Ethernet (1)

Ethernet is a set of networking technologies at data link and physical layers commonly used in computer networks

- First described on 22 May 1973 in a memo written by Robert Metcalfe
- Named after the luminiferous aether: “**The Ether** – That carries transmissions, propagates bits to all stations”
- “The Ether” is not limited to the cable – predicted wireless packet transmissions
- IEEE 802.3 global standard (since 1983)


XEROX

"A CABLE-TREE ETHER"

ALTO

ALTO

ALTO

DYNABOOK

VTS

NOVA

NOVA

XGP

PDP-11

THE ETHER NETWORK

WE PLAN TO BUILD A SO-CALLED BROADCAST COMPUTER COMMUNICATION NETWORK, NOT UNLIKE THE ALOHA SYSTEM'S RADIO NETWORK, BUT SPECIFICALLY FOR IN-BUILDING MINICOMPUTER COMMUNICATION. WE THINK IN TERMS OF NOVA'S AND ALTO'S JOINED BY COAXIAL CABLES.

WHILE WE MAY END UP USING COAXIAL CABLE TREES TO CARRY OUR BROADCAST TRANSMISSIONS, IT SEEMS WISE TO TALK IN TERMS OF AN ETHER, RATHER THAN 'THE CABLE', FOR AS LONG AS POSSIBLE. THIS WILL KEEP THINGS GENERAL AND WHO KNOWS WHAT OTHER MEDIA WILL PROVE BETTER THAN CABLE FOR A BROADCAST NETWORK; MAYBE RADIO OR TELEPHONE CIRCUITS, OR POWER WIRING OR FREQUENCY-MULTI-PLEXED CATV, OR MICROWAVE ENVIRONMENTS, OR EVEN COMBINATIONS THEREOF.

THE ESSENTIAL FEATURE OF OUR MEDIUM -- THE ETHER -- IS THAT IT CARRIES TRANSMISSIONS, PROPAGATES BITS TO ALL STATIONS. WE ARE TO INVESTIGATE THE APPLICABILITY OF ETHER NETWORKS.

"CABLE ETHER"

TELEPHONE COAX BOOSTER

TELEPHONE

TELEPHONE RADIO BOOSTER

RADIO ETHER

TELEPHONE ETHER

-1-

Ethernet memo:
https://broadbandlibrary.com/bob-metcalfe-lays-down-the-law/

Ethernet (2)

- Originally 10 Mb/s, today 400 Gb/s, very soon 800 Gb/s
- Local Area Network (LAN) forms a Network segment
  - All network devices connected over network switches
  - Can directly talk to every other device
  - Also called a Broadcast Domain
    - Broadcast frame reaches every device
- Addressing: 48-bit MAC (Media Access Control) address
  - Also called physical address, unique for every device
- Ethernet flow control mechanism avoiding packet loss
  - A pause frame is sent by NIC or switch in case of full buffers, temporarily stopping the transmitting device

ConnectX-7 **400Gb/s** NIC from NVIDIA (previously Mellanox)

Juniper QFX5130 switch with 32x **400Gb/s** ports

Juniper EX4100-F-12T switch with 12x Gigabit Ethernet, 6x 10 Gigabit Ethernet

Ethernet Frame

- Data are transmitted in **Ethernet frames**

Layer 2 Ethernet II Frame (Destination MAC through CRC)

Preamble	SFD	Destination MAC	Source MAC	VLAN Tag (Optional)	EtherType	Payload	CRC	IPG
7 bytes	1 byte	6 bytes	6 bytes	4 bytes	2 bytes	0-1500 bytes	4 bytes	12 bytes

Layer 1 Ethernet II Frame

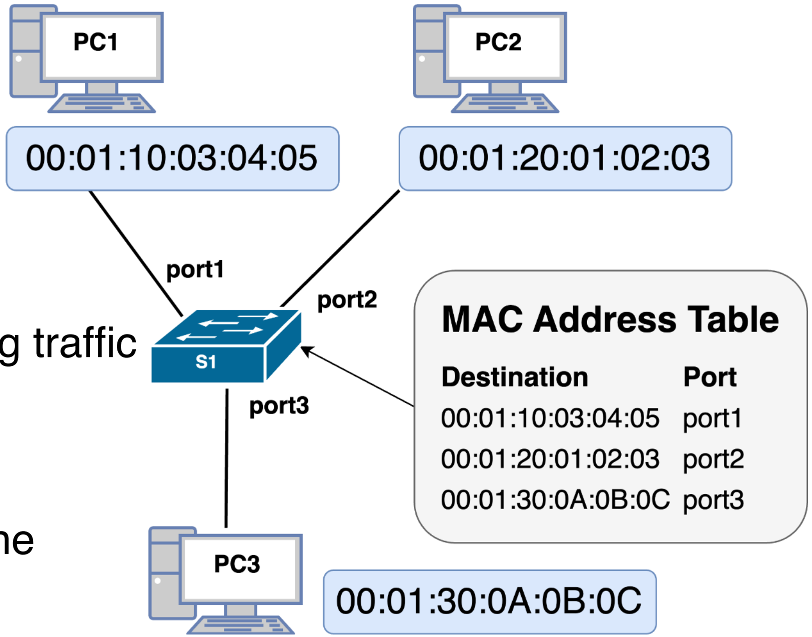
- **MAC address** to identify the Network Interface Controller (NIC) of the device
- **EtherType** identifies encapsulated protocol (0x0800 for IPv4, 0x0806 for ARP)
- Max payload length is 1500 for standard or 9000 for jumbo Ethernet frame
  - Also known as MTU (Maximum Transmission Unit)

- Preamble: Alternating 1 and 0 bit pattern for synchronization (b10101010)
- SFD: Start frame delimiter (b10101011)
- IPG: Inter-packet gap of 96 bit intervals

Frame Forwarding

- Example: PC1 sends a frame:

Destination MAC	Source MAC	…
00:01:30:0A:0B:0C	00:01:10:03:04:05	



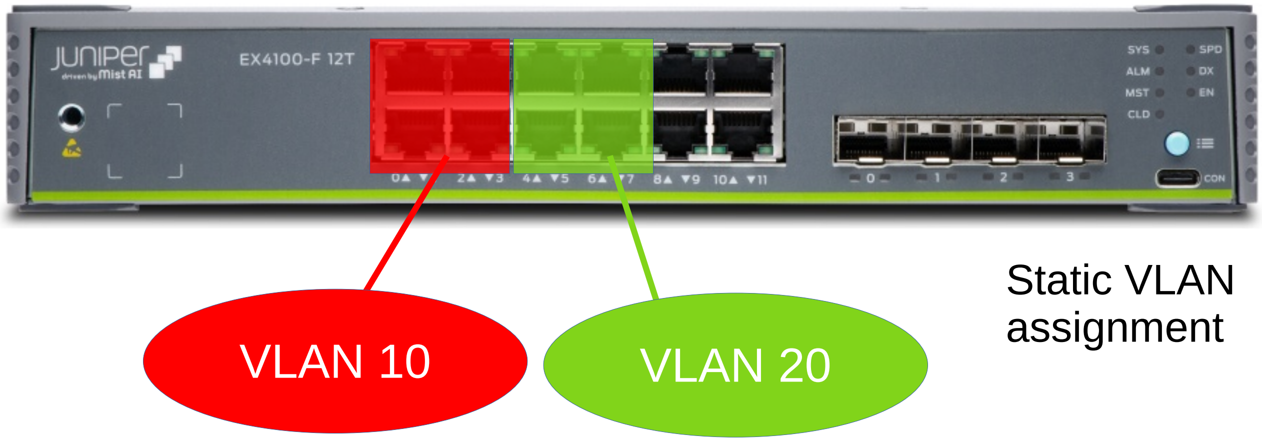
Destination	Port
00:01:10:03:04:05	port1
00:01:20:01:02:03	port2
00:01:30:0A:0B:0C	port3

Network Switch is layer 2 device

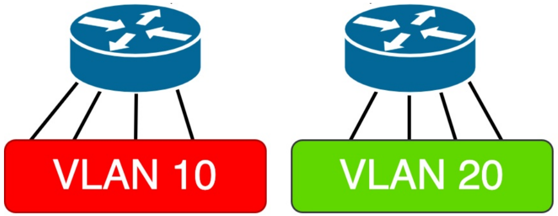
- Switch builds (learns) the MAC address table by observing traffic
- Forwards Ethernet frame to the output port based on the destination MAC address
- Frame for unknown destination is broadcasted to all but the original port
- The same applies for broadcast frames (destination MAC address FF:FF:FF:FF:FF:FF)
- When reply arrives, switch updates the MAC address table (learns by observing traffic)
- Items in MAC Address Table expires after a timeout (minutes)

Ethernet Virtual LAN

- Managed switch can be configured to create a virtual LAN (VLAN)
  - Separate isolated network with its own broadcast domain and separate IP address range (subnet)
  - VLANs can be static (fixed port assignment)
  - VLANs can be tagged (trunk)
    - TAG (4 bytes) added to Ethernet frame
    - Prioritization/Rate limiting can be applied between different VLAN tags

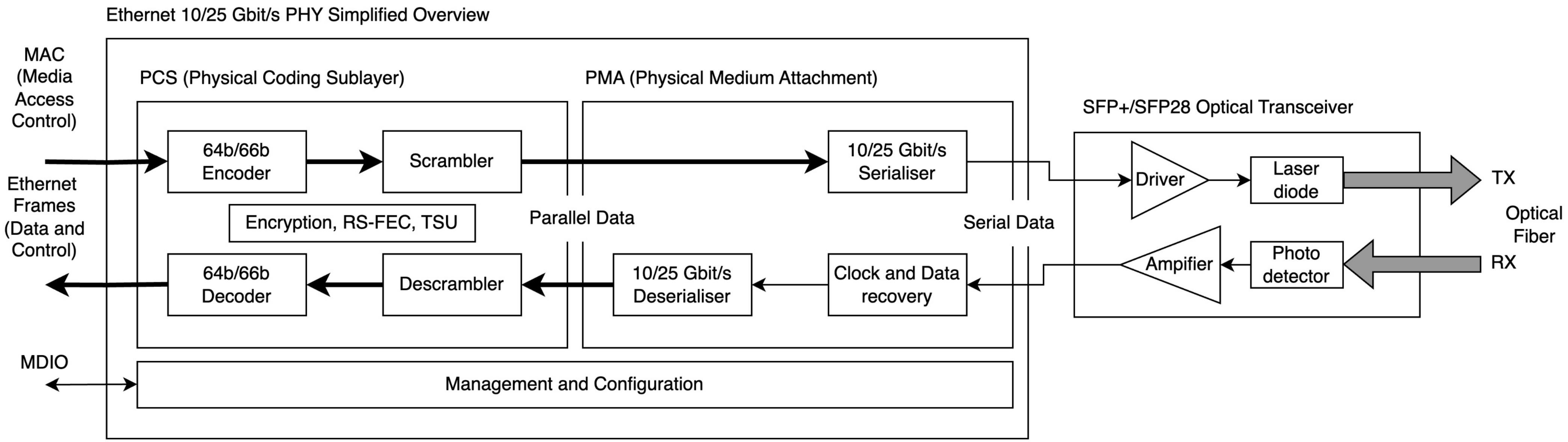


Static VLAN assignment



Standard Ethernet frames (no VLAN tag)

Ethernet Physical Layer 10/25 Gbit/s



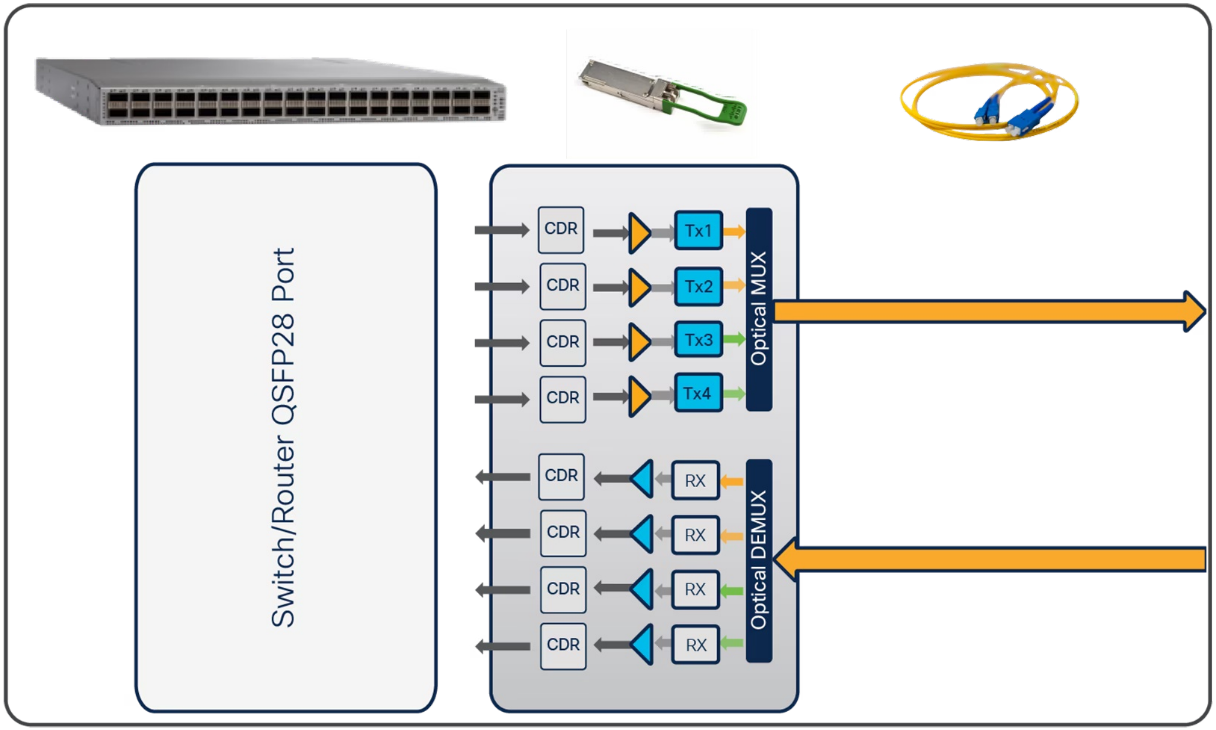
- 64/66b encoding: Adding 2 synchronization bits
  - Data word or control word (Idle, Start of Frame, End of Frame)
- Scrambler
  - Ensures even distribution of 1s and 0s in transmitted data for correct clock synchronization in the receiver

Going to higher speeds (100/200 Gbit/s)

- Higher speeds achieved through parallel data transmission over several electrical channels (lanes)

Common Optical Transceivers

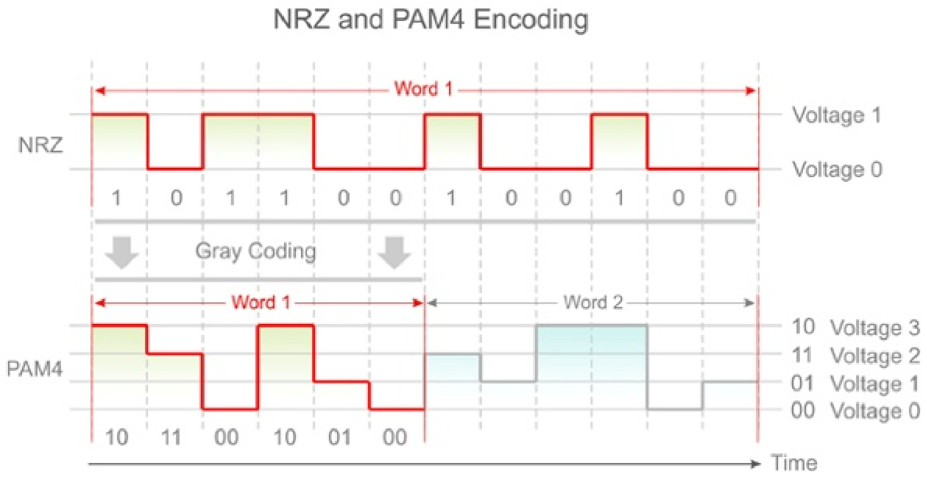
- SFP28: 25 Gbit/s
  - Small form-factor pluggable
  - Single lane device (25 Gbit/s)
- QSFP28: 100 Gbit/s
  - Quad SFP
  - 4-lane device (4x 25 Gbit/s)
- QSFP-DD: 200 Gbit/s
  - QSFP Double Density
  - 8-lane device (8x 25 Gbit/s)

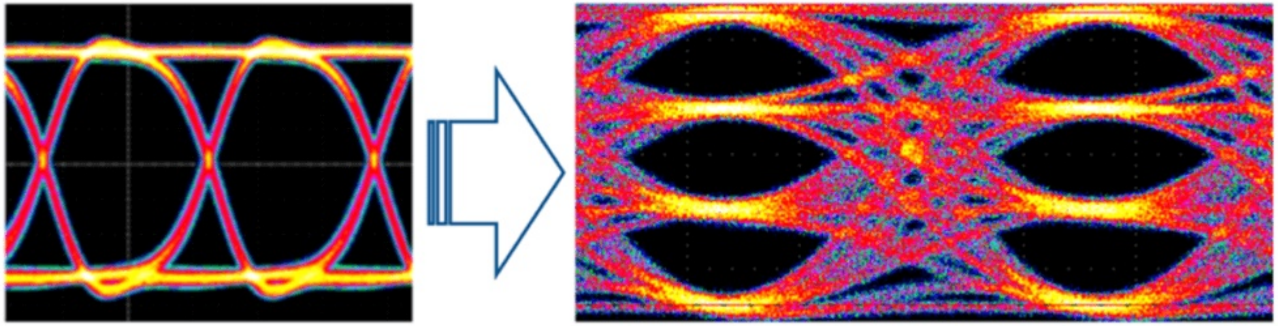


Source:
https://www.cisco.com/c/en/us/products/collateral/interfaces-modules/transceiver-modules/solution-overview-c22-743387.html

Going to even higher speeds (400 Gbit/s)

- NRZ (Non-Return-to-Zero) encoding
  - Binary code using low and high signal levels to represent single bit of information
- PAM4 (Pulse Amplitude Modulation 4-level) encoding
  - Multilevel (4-level) signal modulation format used to transmit signal.
  - Each signal level can represent 2 bits of information.
  - Transmitting 50 Gbit/s per lane
- QSFP56: 200 Gbit/s
- QSFP56-DD: 400 Gbit/s





NRZ Eye Diagram PAM4

Source:
https://blog.samtec.com/post/understanding-nrz-and-pam4-signaling/

Network Layer (3)

Internet Layer in the Internet Protocol Suite

Internet Protocol

Forms the **Internet**

- First described in 1974 by Vint Cerf and Bob Kahn
- IPv4 defined in RFC 791 (year 1981)
- IPv6 defined in RFC 2460 (year 1998) and RFC 8200 (year 2017)

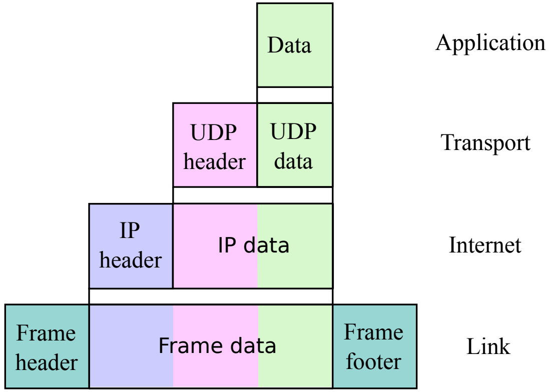
Properties

- Defines a logical addressing system: **IP Address**
- Provides communication between independent networks: performs **routing**
  - Forwarding packets (a hop) from one network to another based on the routing table
- Connectionless and stateless protocol
- Encapsulates and transports higher level protocols (TCP, UDP)
- Error and control messages signaled by ICMP (Internet Message Control Protocol) RFC 792
  - Also used by ping tool

IPv4 Header

- IPv4 header contains 20 bytes when used without options
  - Total length is the size of the entire packet including header and data payload
  - Time To Live (TTL) is decreased on every hop, packet is dropped when zero (prevents routing loops)
  - Protocol defines the (encapsulated) protocol in the data payload (6 for TCP, 17 for UDP)
  - Source and destination IP addresses are a 32-bit number

Protocol encapsulation



IPv4 header format

<table>
<tr><th>Offsets</th><th>Octet</th><th colspan="8">0</th><th colspan="8">1</th><th colspan="8">2</th><th colspan="8">3</th></tr>
<tr><th>Octet</th><th>Bit</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><th>0</th><th>0</th><td colspan="4">Version</td><td colspan="4">IHL</td><td colspan="6">DSCP</td><td colspan="2">ECN</td><td colspan="16">Total Length</td></tr>
<tr><th>4</th><th>32</th><td colspan="16">Identification</td><td colspan="3">Flags</td><td colspan="13">Fragment Offset</td></tr>
<tr><th>8</th><th>64</th><td colspan="8">Time To Live</td><td colspan="8">Protocol</td><td colspan="16">Header Checksum</td></tr>
<tr><th>12</th><th>96</th><td colspan="32">Source IP Address</td></tr>
<tr><th>16</th><th>128</th><td colspan="32">Destination IP Address</td></tr>
<tr><th>20</th><th>160</th><td colspan="32" rowspan="3">Options (if IHL > 5)</td></tr>
<tr><th>⋮</th><th>⋮</th></tr>
<tr><th>56</th><th>448</th></tr>
</table>

IPv4 Address Scheme (1)

- Every network node has IP address, assigned
  - Statically (e.g. by a configuration file in the OS)
  - Automatically through **DHCP** (Dynamic Host Configuration Protocol), RFC 2131
    - Network management protocol on top of UDP
    - Provides IP configuration (IP address, netmask, default gateway, DNS servers, time server, …)
    - IP address given based on a unique identifier, usually MAC address, but other options possible

- For sending a packet over the Link layer (Ethernet), a physical address is needed (MAC)
  - ARP (Address Resolution Protocol) is used to translate IP addresses to physical address
    - Broadcast message: `Who has 10.0.30.10?`
    - Receive reply: `10.0.30.10 is-at 08:00:30:11:22:33`
    - Replies are cached in ARP cache on the host, for limited amount of time

IPv4 Address Scheme (2)

- IP address is divided in network address and host address
- The size of the network (in bits) is determined by the network mask

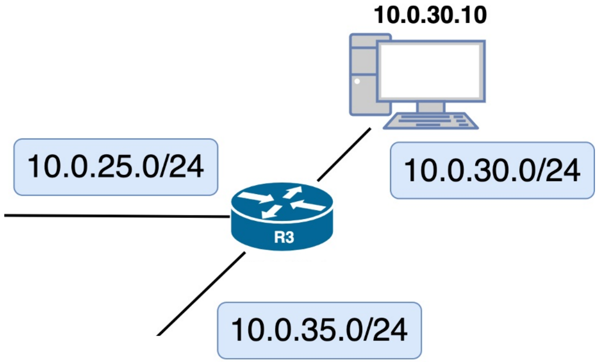
```
IP: 10.0.30.10/24   Mask:255.255.255.0

IP:     00001010.00000000.00011110.00001010   10.0.30.10
MASK:   11111111.11111111.11111111.00000000   255.255.255.0
---------------------------------------------------------
Bitwise AND gives the network address:

 =      00001010.00000000.00011110.00000000   10.0.30.0 <— Network (address)
```

Network address	Host address
10.0.30.	10

- Network address determines target network in the routing
- IP address 255.255.255.255 is the broadcast address for the local network, it is not routed.

IP Routing (1)

- Networks are interconnected via routers (gateways)
- Example: Router R3 receives IP packet with the **destination IP address: `10.0.30.10`**
  - Reminder:
    - Network address: `10.0.30.0/24`
    - Network mask: `255.255.255.0`
- Router decides where to forward the packet (a hop) based on the destination IP network
  - Once the routing decision is made, MAC address is obtained through ARP
    - `Who has 10.0.30.10?`
    - `Reply: 10.0.30.10 is-at 08:00:30:11:22:3`
  - Then packet is routed and forwarded with the obtained destination MAC address

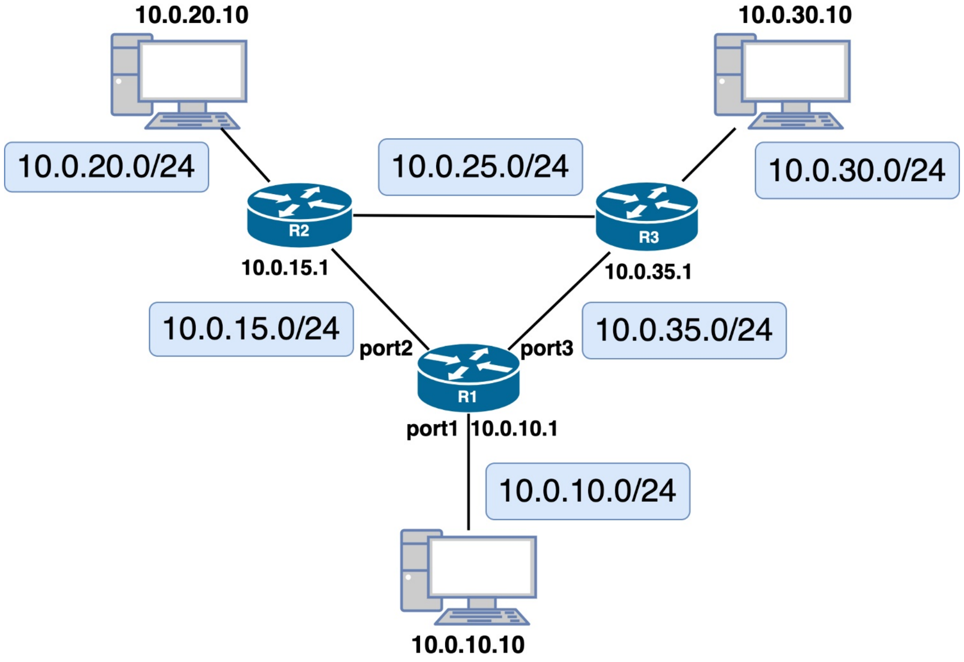


IP Routing (2)

- Example:

`IP 10.0.10.10 > 10.0.30.10`

  – Reminder:
    - Network address: `10.0.30.0/24`
    - Network mask: `255.255.255.0`



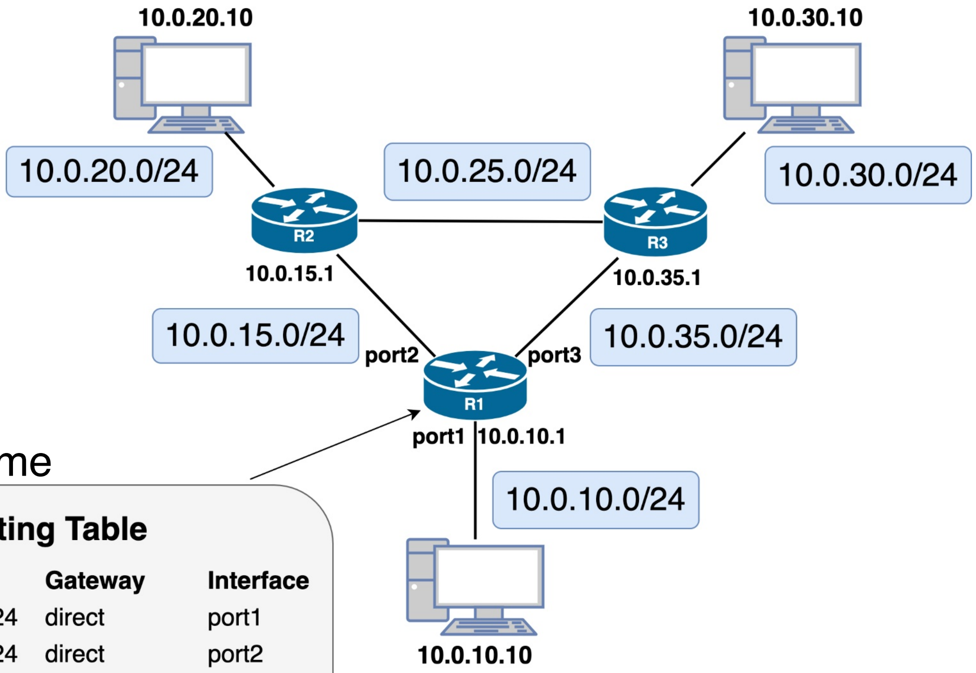
IP Routing (3)

- Example:

`IP 10.0.10.10 > 10.0.30.10`

  - Reminder:
    - Network address: `10.0.30.0/24`
    - Network mask: `255.255.255.0`

- Routing table
  - Defines the next hop for the IP packet
  - It is possible to have multiple paths to the same destination network, for redundancy
  - TTL field is updated, prevents infinite loops



R1 Routing Table

Network	Gateway	Interface
10.0.10.0/24	direct	port1
10.0.15.0/24	direct	port2
10.0.35.0/24	direct	port3
10.0.20.0/24	10.0.15.1	port2
10.0.25.0/24	10.0.15.1	port2
	10.0.35.1	port3
10.0.30.0/24	10.0.35.1	port3

Note:
- Diagram is simplified, some gateway address were omitted

Routing Summary

- Router decides where to forward the packet (a hop) based on the destination IP network and rules in the routing table
  - To a local network if directly attached to the router
  - To one of the next gateways
  - To the default gateway
  - Otherwise ICMP error message is sent back (e.g. “Destination network unreachable”)
- Routing tables are filled
  - Statically
  - Dynamically via a routing exchange protocol
    - OSPF, BGP

Transport Layer (4)

Internet Protocol Suite (TCP and UDP)

Transport Layer (Internet Protocol Suite)

- Provides host-to-host (end-to-end) transport service for applications
- Connection endpoints are identified through network address (IP) and port number (16 bit)
- Two main services are provided:
  - **TCP (Transmission Control Protocol)**
  - **UDP (User Datagram Protocol)**
- Common API is provided by the operating system
  - Modeled according to the Berkeley **socket interface** from 4.2BSD (1983)
  - Network socket types (Datagram, Stream, Raw)

TCP (Transmission Control Protocol)

Connection-oriented service providing reliable transport for TCP streams

- First described in 1974 by Vint Cerf and Bob Kahn
- Defined in RFC 675 and RFC 793 (year 1981)
- With full-duplex communication
- Header contains 20 bytes with no options

Reliable protocol

- Data are split and transmitted in segments
- Every sent segment has a sequence number
- Every received segment is acknowledged
- Peers know when data was delivered*
- Any lost segment(s) are re-transmitted (after a timeout)
- Data are delivered in-order

TCP segment header

<table>
<tr><th colspan="2">Offsets</th><th colspan="10">0</th><th colspan="10">1</th><th colspan="10">2</th><th colspan="2">3</th></tr>
<tr><th>Octet</th><th>Bit</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>0</th><th>1</th></tr>
<tr><th>0</th><th>0</th><td colspan="16">Source port</td><td colspan="16">Destination port</td></tr>
<tr><th>4</th><th>32</th><td colspan="32">Sequence number</td></tr>
<tr><th>8</th><th>64</th><td colspan="32">Acknowledgment number (if ACK set)</td></tr>
<tr><th>12</th><th>96</th><td colspan="4">Data offset</td><td colspan="4">Reserved
0 0 0 0</td><td>C
W
R</td><td>E
C
E</td><td>U
R
G</td><td>A
C
K</td><td>P
S
H</td><td>R
S
T</td><td>S
Y
N</td><td>F
I
N</td><td colspan="16">Window Size</td></tr>
<tr><th>16</th><th>128</th><td colspan="16">Checksum</td><td colspan="16">Urgent pointer (if URG set)</td></tr>
<tr><th>20</th><th>160</th><td colspan="32" rowspan="3">Options (if data offset > 5. Padded at the end with "0" bits if necessary.)</td></tr>
<tr><th>⋮</th><th>⋮</th></tr>
<tr><th>56</th><th>448</th></tr>
</table>

https://en.wikipedia.org/wiki/Transmission_Control_Protocol

***Delivered to the TCP/IP stack of the peer, not necessarily to the user space application**

Network Congestion and TCP Congestion Control

Network congestion causes low throughput due to packet drop and increased end-to-end delay

- Congestion is a network (design) problem, e.g. not enough bandwidth, slow links, ...
- Example: Two senders (2x 10Gbit/s) each sending a stream to single receiver (1x 10 Gbit/s)
  - Creates network congestion in the network switch
  - Network switch drops packet(s) that cannot be forwarded to the destination
  - If not controlled will cause a **congestion collapse** dues to the re-transmissions taking significant part of the network bandwidth severely limiting the useful throughput

TCP Congestion control prevents packet drop and congestion collapse by limiting the amount of data sent to the network

- Limitation happens at the sender
- https://en.wikipedia.org/wiki/TCP_congestion_control

TCP Flow Control

Flow control is end-to-end **protection mechanism** allowing the sender to send only the number of segments that the receiver can safely handle

- Is a **back-pressure mechanism**, protecting the receiver
- The receiver advertises the size of available **receiver buffer** through TCP **Window Size** field in the TCP header (in the ACK packet)

TCP Socket Buffer Size

- Variable that defines the size of the TCP buffer in the operating system
  - Two values, one for the sender and one for the receiver
- Puts the limit on the TCP window size
- Sent data are stored in this buffer and waiting for ACK; If not coming, then the whole window is re-transmitted after a timeout

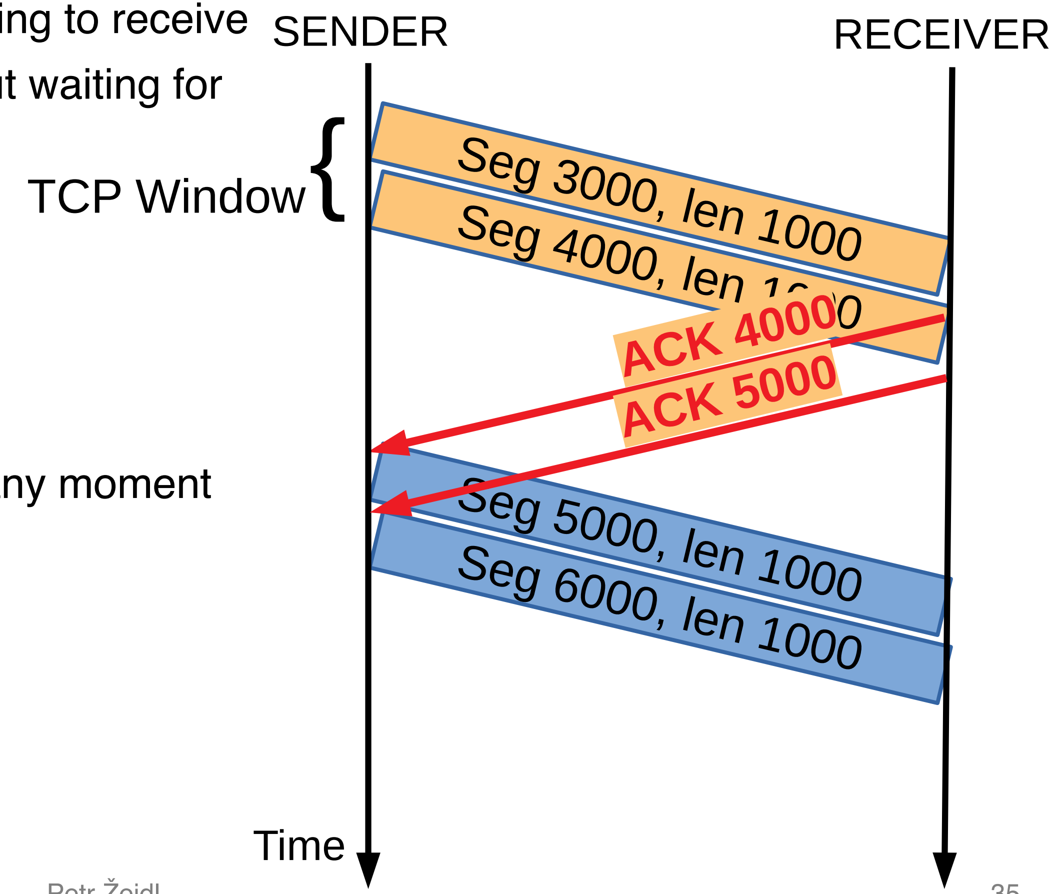
TCP Window

TCP Window

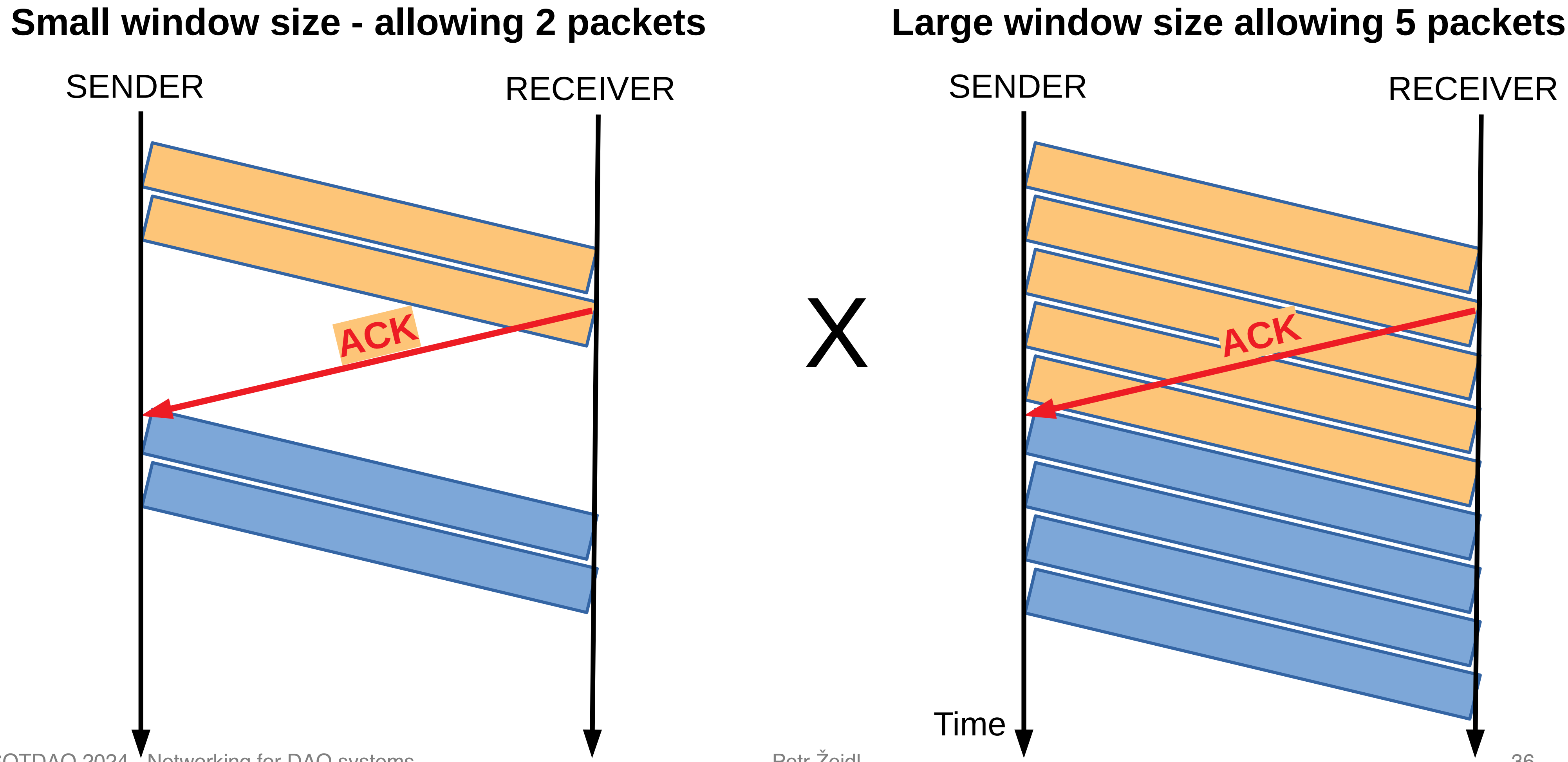
- Number of bytes the receiver is currently willing to receive
- Number of bytes the sender can send without waiting for ACK

Example

- Max allowed segment size is 1000 bytes
- Window Size is 2000 bytes
- **Two segments/packets** will be **in flight** at any moment
  - 2 segments, each contains 1000 bytes

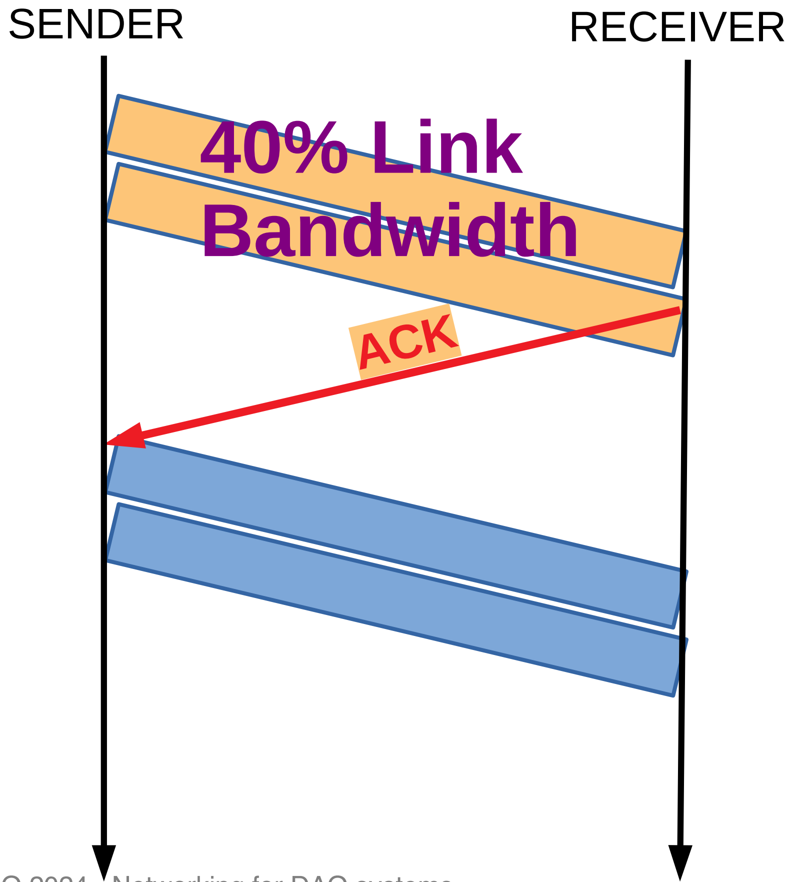


Good TCP throughput ↔ keep sending data



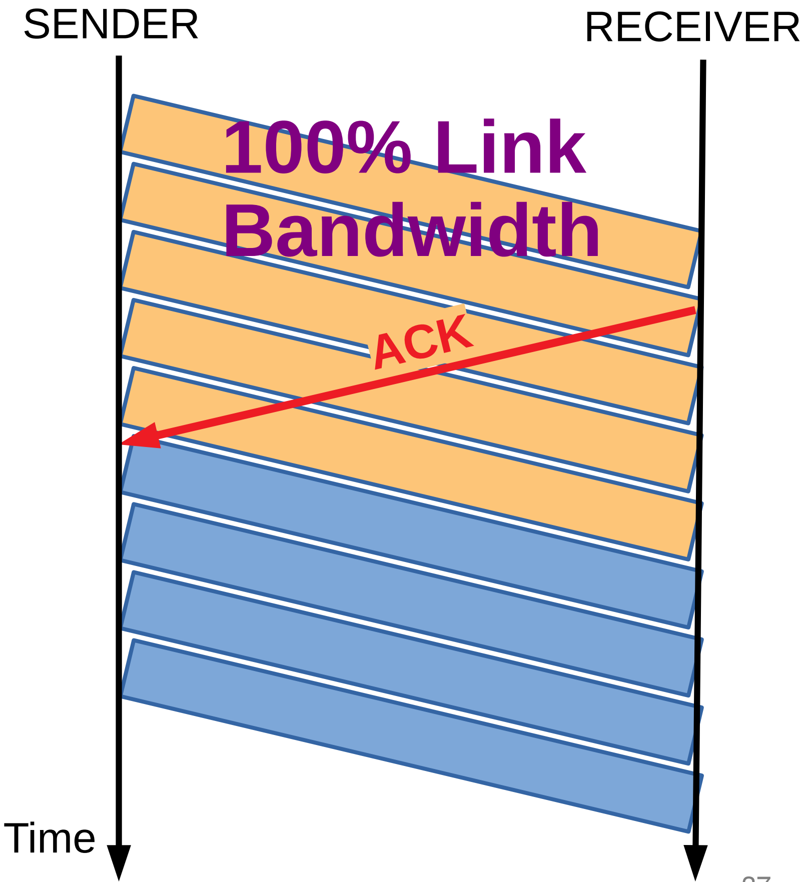
Good TCP throughput ↔ keep sending data

Small window size - allowing 2 packets



X

Large window size allowing 5 packets

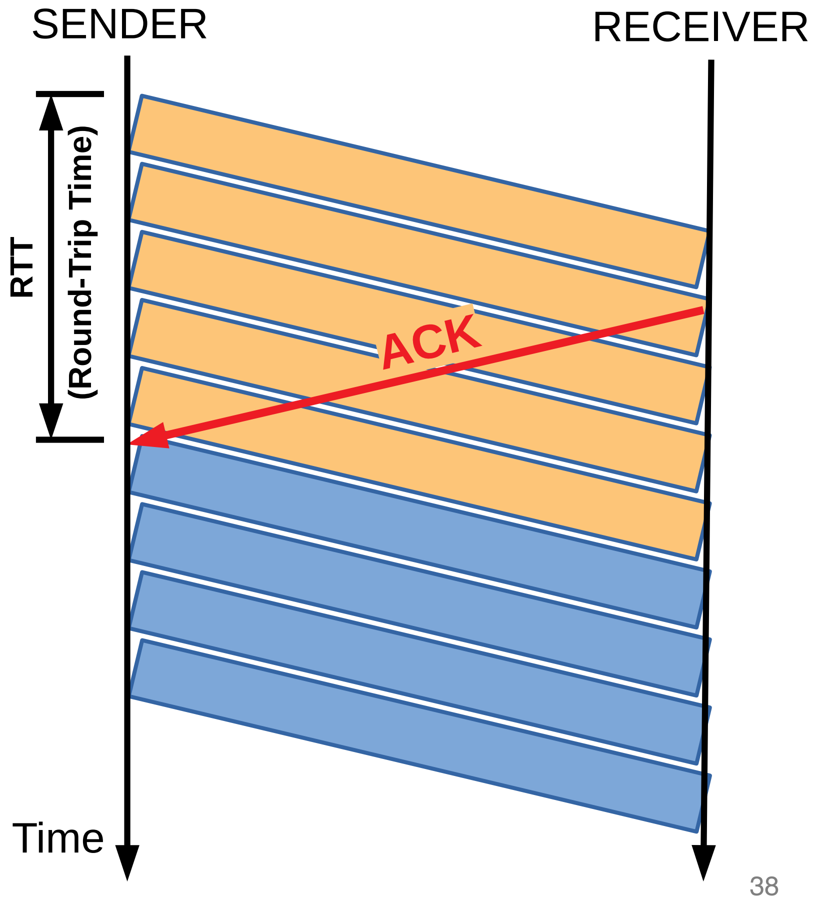


Good TCP throughput ↔ keep sending data

- Why my TCP throughput is low?
  - The receiver is slow
  - **TCP buffers are too small**
- What should be the size of the buffer?
  - Bandwidth-delay product (BDP):

BDP [bits] = Round Trip Time [s] * Link Bandwidth [bit/s]

- Use ping tool to estimate RTT of your network



Good TCP throughput ↔ keep sending data

Fast network, short distance

- Link bandwidth: 100 Gbit/s
- E.g. DAQ Network

RTT [us]	BDP [bytes]
40	50 000
100	1 250 000
1000	12 500 000

- **Slower network, long distance**
  - Link bandwidth: 10 Gbit/s
  - E.g. storage network

RTT [msec]	BDP [bytes]
1	1 250 000
10	12 500 000
40	50 000 000

- The default TCP socket buffer settings in CentOS 8 / RHEL 8:
  - Receive buffer (net.ipv4.tcp_rmem) = 6 MB
  - Send buffer (net.ipv4.tcp_wmem) = 4 MB

! Need to tune for your network!

UDP (User Datagram Protocol)

Connection-less service providing unreliable transport (datagrams)

- Designed by David P. Reed in 1980
- Defined in RFC 768
- Stateless
- Header contains 8 bytes

Unreliable protocol

- Provides no guarantees for message delivery, in-order delivery or duplicate protection

Advantages

- Simple protocol with minimal protocol handling in the OS – Low processing latency
- Supports broadcast and multicast, i.e. to send information to many destinations, e.g. in IPTV
- Used in DNS, DHCP, SNMP protocols and for network tunneling

https://en.wikipedia.org/wiki/User_Datagram_Protocol

UDP datagram header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Length																Checksum															

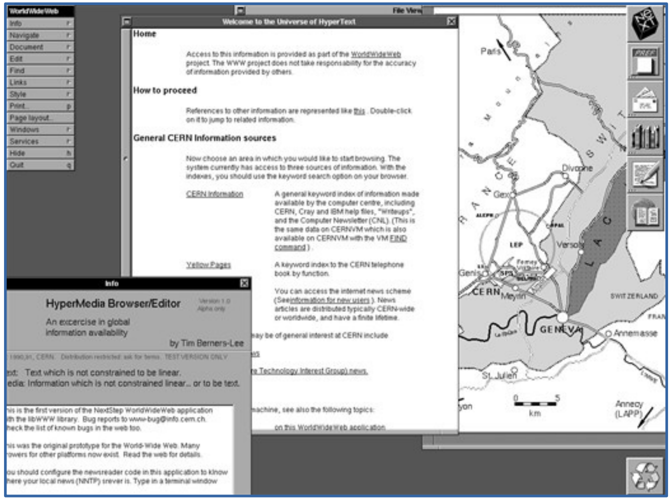
Application Layer

Application Layer

Many protocols at the application layer: DHCP, DNS, SSH, ...

One example is **HyperText Transfer Protocol (HTTP)**

- Client / server protocol (HTTP Request / HTTP Response)
- Heart of **World Wide Web** (WWW)
- First described by Tim Berners-Lee at CERN (year 1989)
- **The first website: http://info.cern.ch/**



A screenshot showing the WorldWideWeb browser created by Tim Berners-Lee

Source:
https://home.cern/science/computing/birth-web/short-history-web

Performance Considerations

Performance Considerations (1)

- Frame length & protocol overhead/data efficiency (useful payload vs total frame size)

Frame type	MTU	Layer 1 overhead Preamble+IPG	Layer 2 overhead Ethernet+FCS	Layer 3 overhead IPv4	Layer 4 overhead TCP	Payload (MSS)	Total	Efficiency
Standard	1500	20	18	20	20	1460	1538	93.93%
Jumbo	9000	20	18	20	20	8960	9038	99.14%

$$efficiency = \frac{useful\ payload}{useful\ payload + overhead}$$

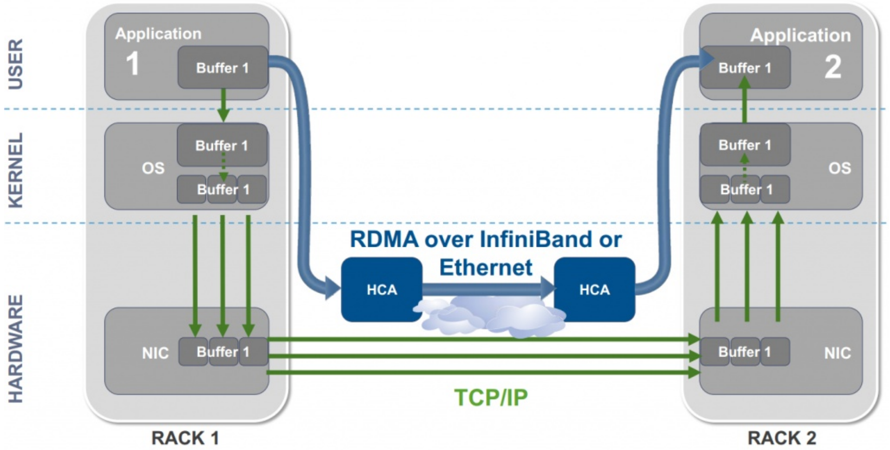
- TCP block of 64K bytes is split into
  - 45 segments of 1460 bytes
  - 8 segments of 8960 bytes
- Small segments
  - Increases frequency of interrupt requests (IRQs) from NIC
  - Substantial CPU cycles spent on TCP protocol handling
- Max throughput obtained with TCP offload (hardware acceleration on NIC)
  - TCP segmentation offload (TSO)
  - Large receive offload (LRO)

Performance Considerations (2)

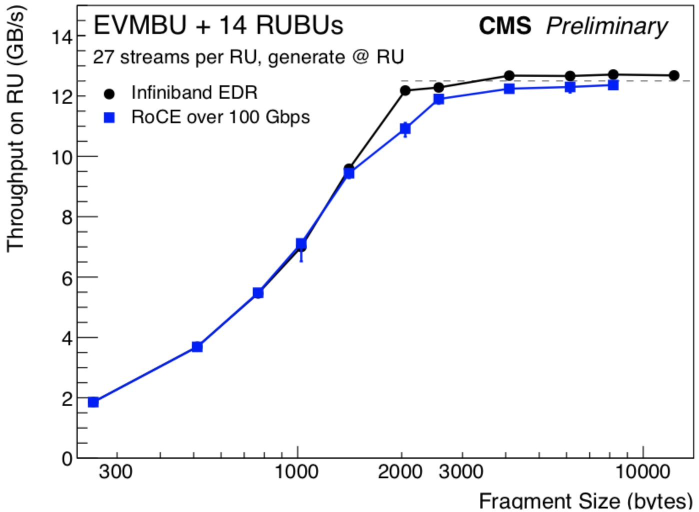
- Year 2020 server node with 3 GHz CPU (32 cores, 128MB cache) and 100GE NIC
  - 1 CPU core at 100% for receiving one stream at ~50 Gbit/s
  - 1 CPU core at 30% for IRQ processing
  - More than 2 cores used only for the protocol processing at 100 Gbit/s!
- Advantages of TCP/IP and Ethernet
  - Very well known protocols
  - Easy to debug (e.g. tcpdump)
- Disadvantages
  - Large CPU consumption on high speed networks
  - Achieving high throughput or low-latency requires some OS and NIC tuning
- Going beyond required HPC (High-Performance Computing) network
  - Using RDMA (Remote DMA) over Infiniband or RoCE network
    - CPU is idle during the RDMA transfer

RoCE (RDMA over Converged Ethernet)

- **Exploring RoCE v2**
  - Enables remote direct memory access (RDMA) between servers without involving CPU
  - Infiniband protocol encapsulated in UDP/IP packet
  - **Fully accelerated by the network adapter (HCA),** transparent to the OS
  - **The software API (IB verbs) is the same for Infiniband and RoCE**
- Comparing RoCE with native Infiniband
  - Gives similar performance
- RoCE has strong network requirements
  - **Requires loss-less non-blocking Ethernet network**
  - Relies on Ethernet (priority) flow control
  - May require Explicit Congestion Notification (ECN)
    - Similar to TCP congestion control but with network switch support



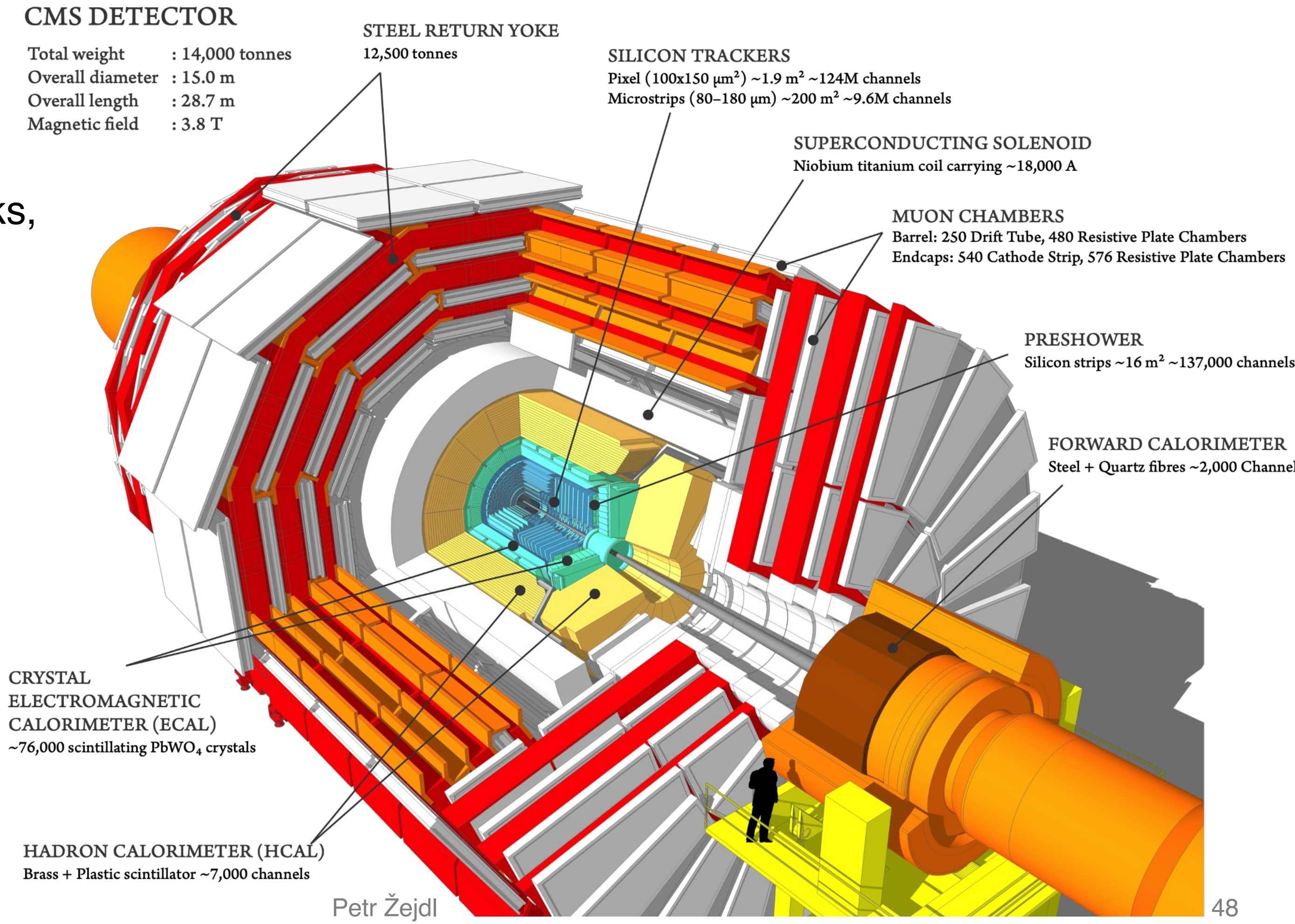
Source: Mellanox - Advanced Network and Storage Interconnect Technologies



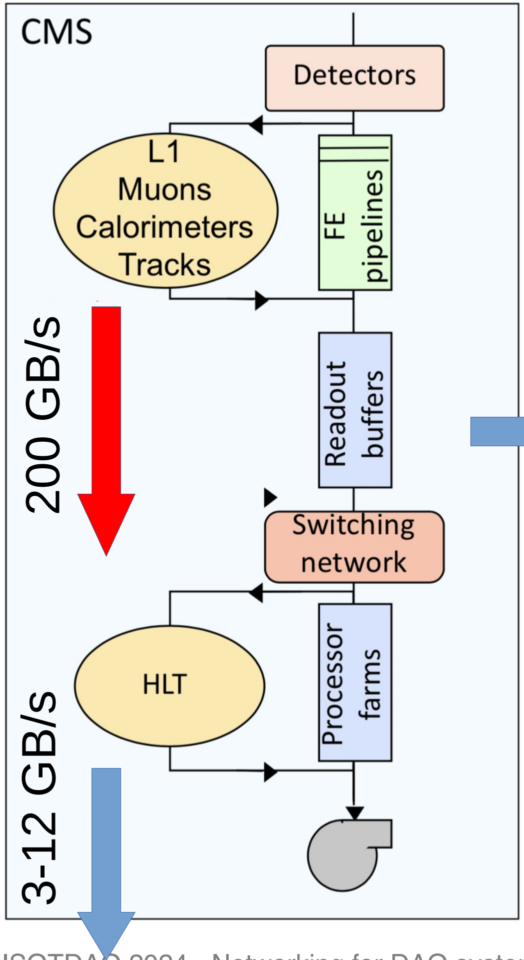
Real Data Acquisition System
CMS DAQ

CMS (Compact Muon Solenoid)

- 40 MHz collision rate
- 100 kHz L1 trigger rate
- 134 M channels
- DAQ interfaced with 660 links, each over 10 Gb/s Ethernet
- DAQ throughput 1.6 Tb/s

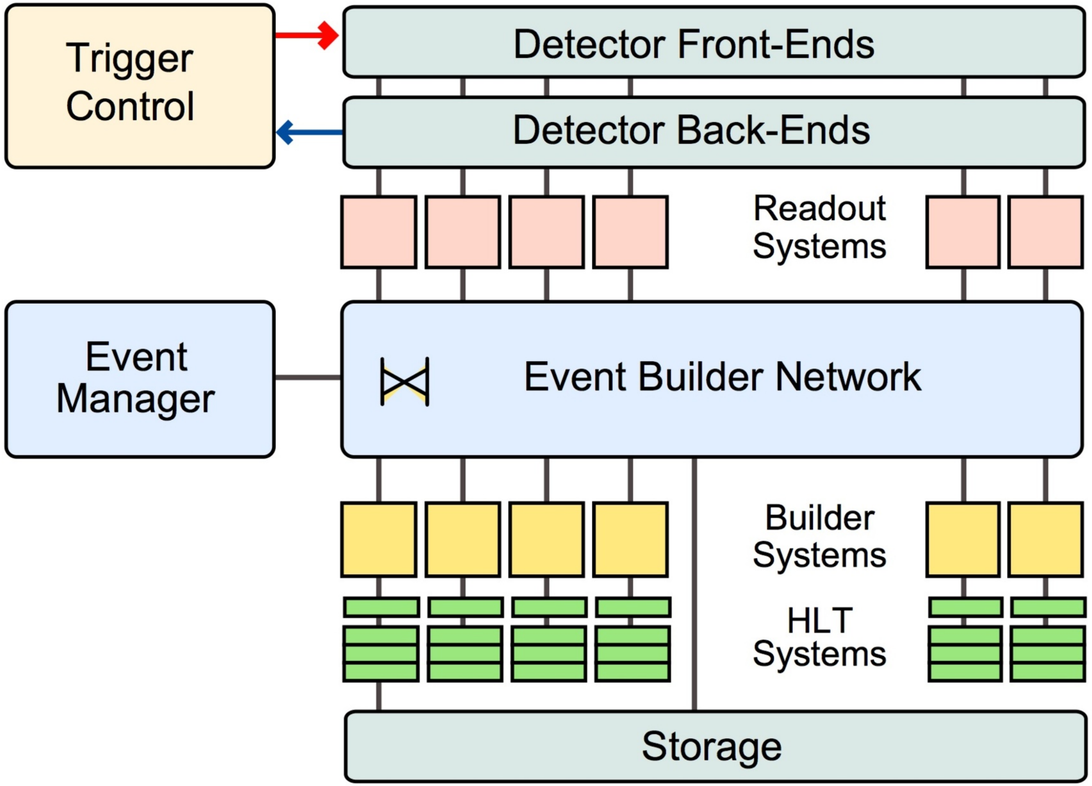


DAQ to 1st order



Parallel readout, linear event building

Run-2:



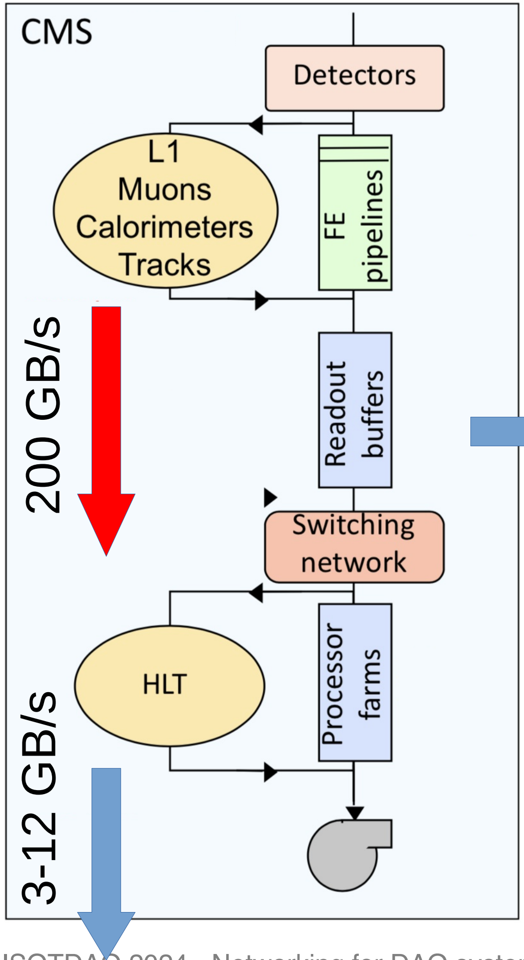
~90 PCs
Readout Unit (RU)

~70 PCs
Builder Unit (BU)

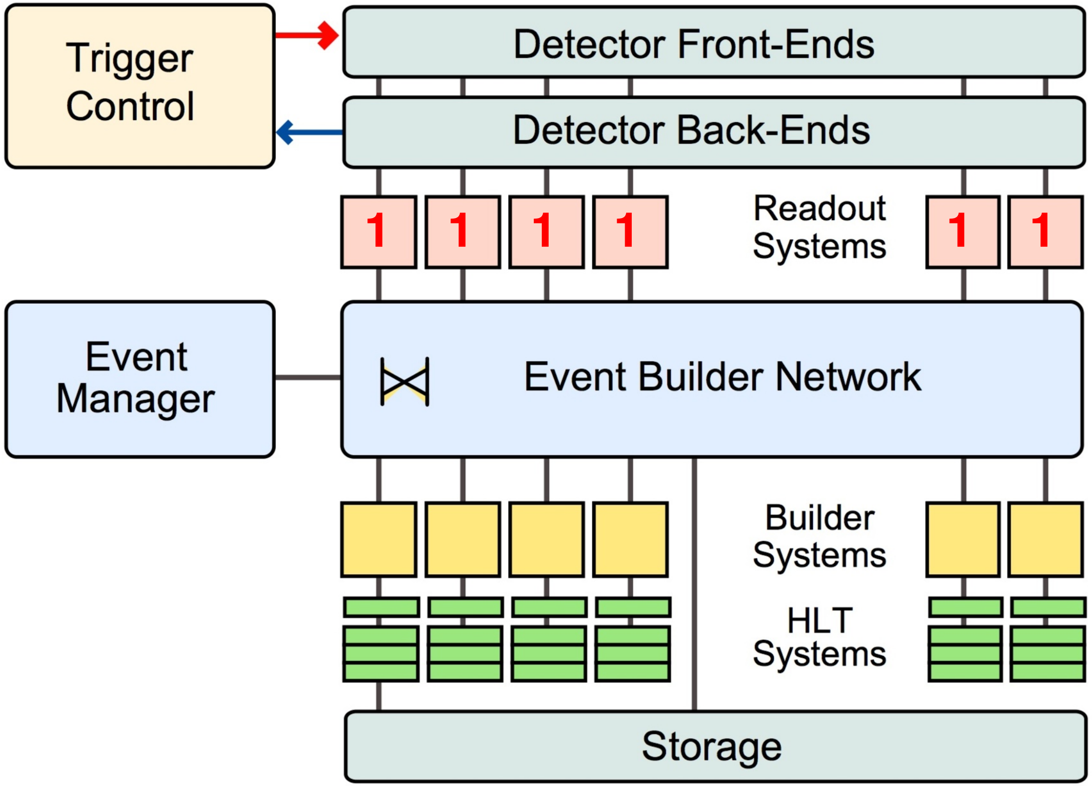
~1100 PCs
(~32k CPU cores)

Event building: Fragments from all RUs have to be read in order to assemble a single event in a single BU (the process is happening in parallel for all BUs)

DAQ to 1st order, Event Building



Parallel readout, linear event building



Run-2:

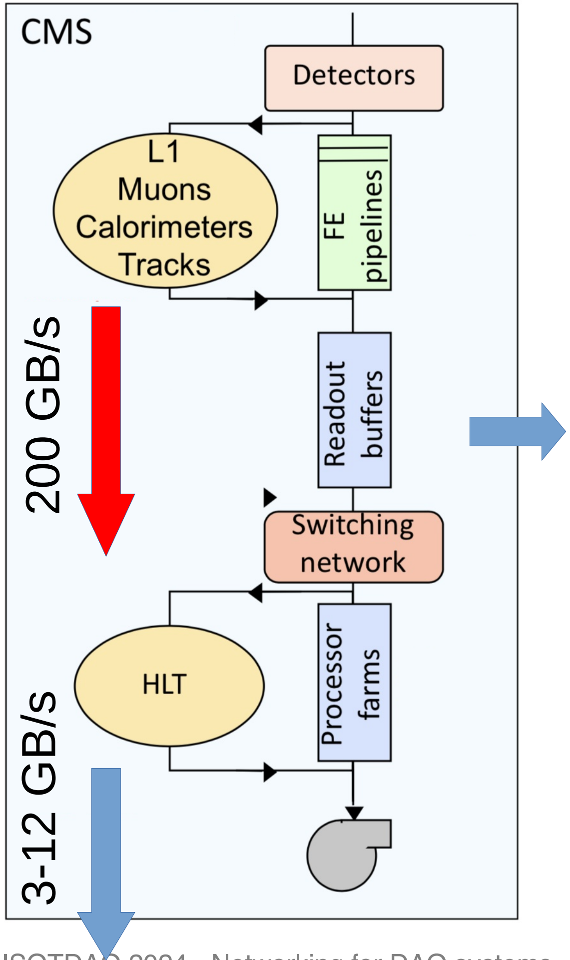
~90 PCs
Readout Unit (RU)

~70 PCs
Builder Unit (BU)

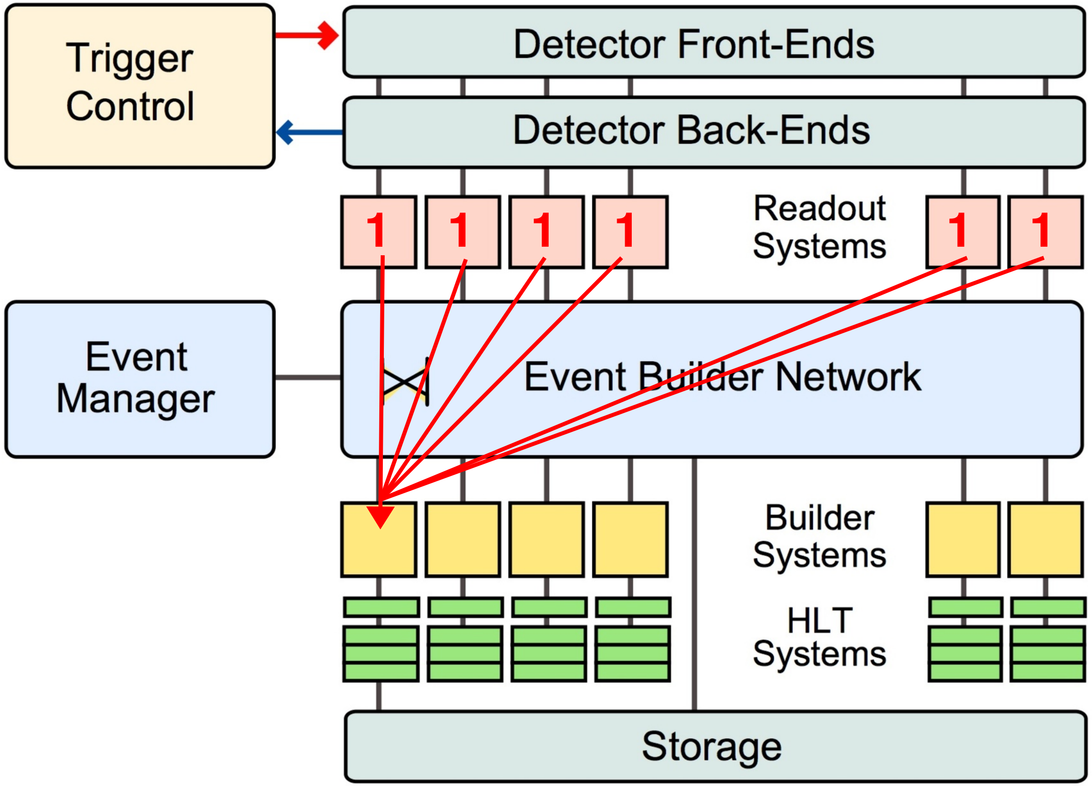
~1100 PCs
(~32k CPU cores)

Event building: Fragments from all RUs have to be read in order to assemble a single event in a single BU (the process is happening in parallel for all BUs)

DAQ to 1st order, Event Building



Parallel readout, linear event building



Run-2:

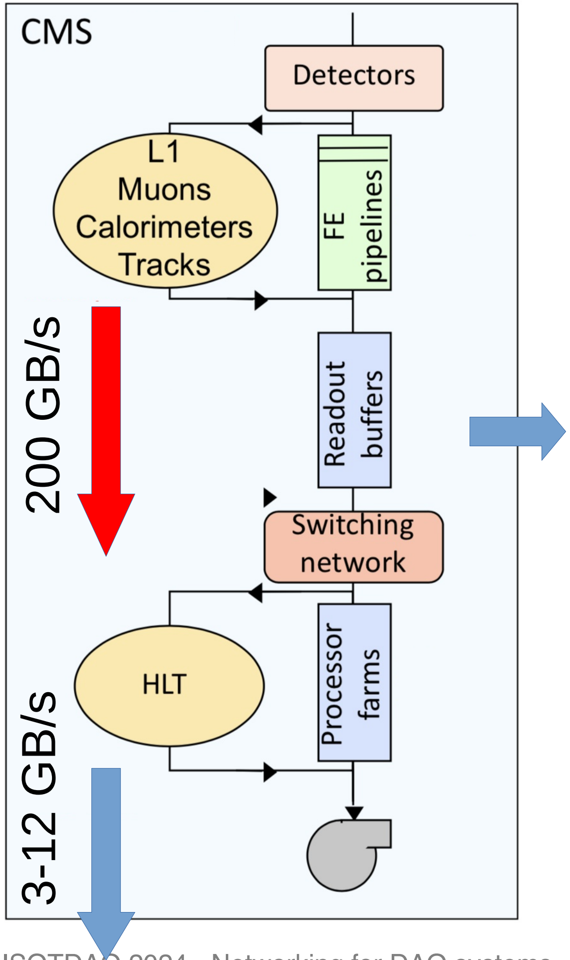
~90 PCs
Readout Unit (RU)

~70 PCs
Builder Unit (BU)

~1100 PCs
(~32k CPU cores)

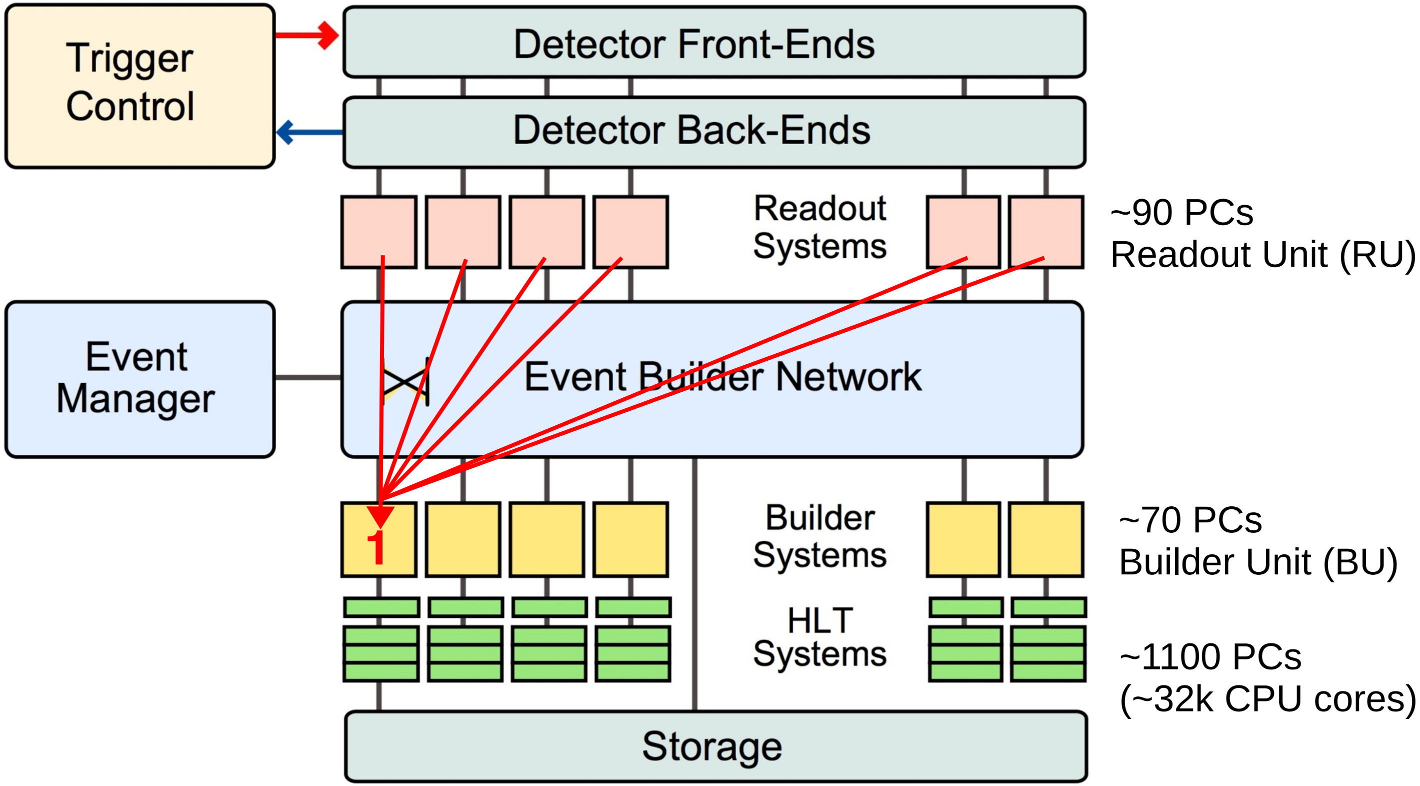
Event building: Fragments from all RUs have to be read in order to assemble a single event in a single BU (the process is happening in parallel for all BUs)

DAQ to 1st order, Event Building



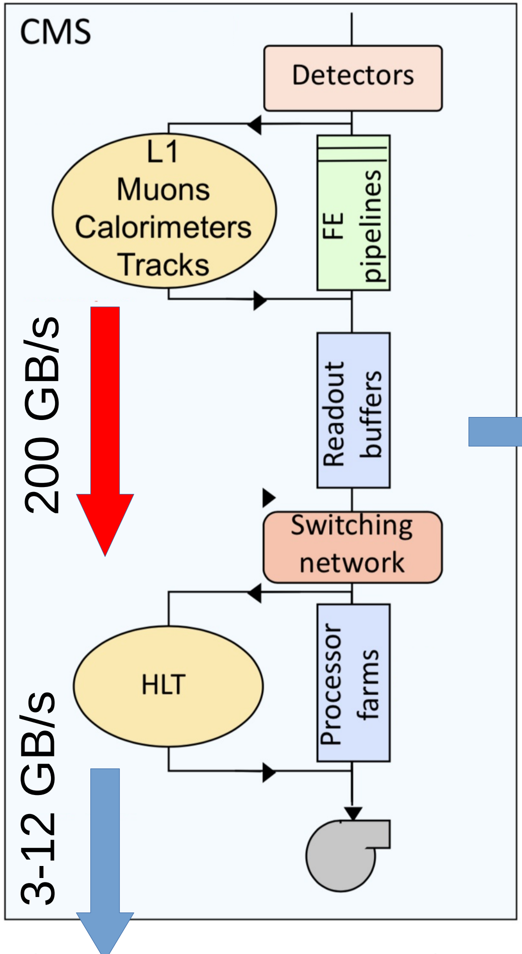
Parallel readout, linear event building

Run-2:

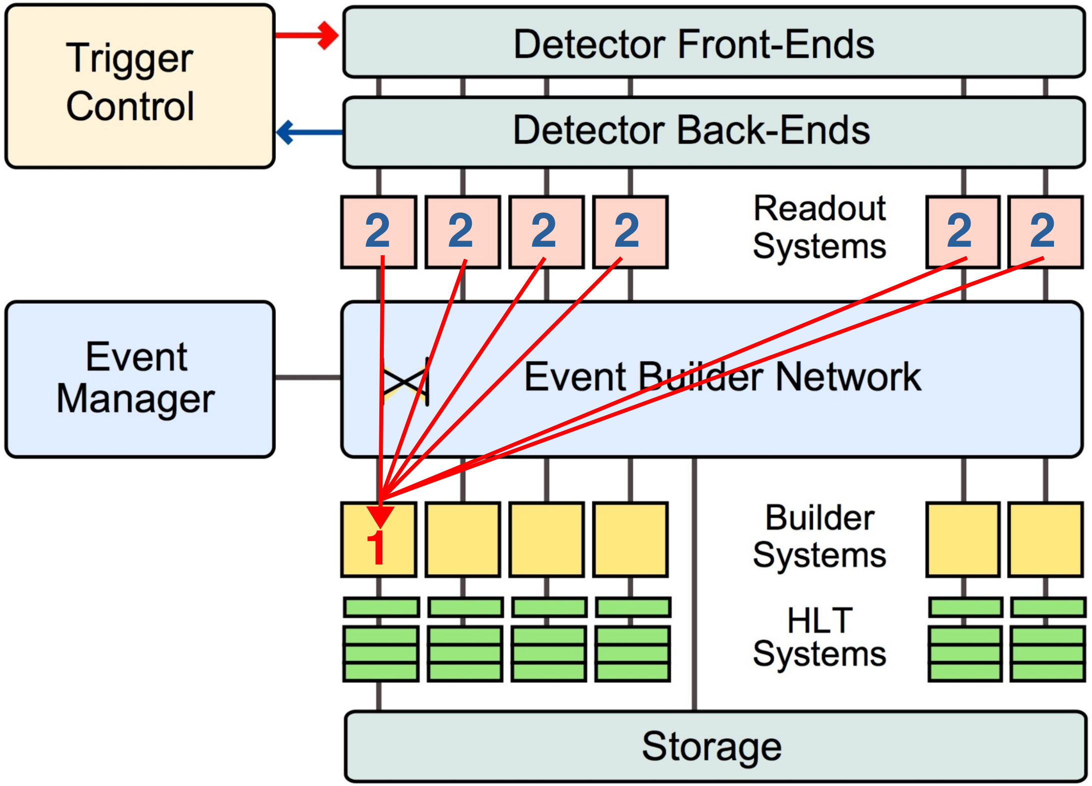


Event building: Fragments from all RUs have to be read in order to assemble a single event in a single BU (the process is happening in parallel for all BUs)

DAQ to 1st order, Event Building



Parallel readout, linear event building



Run-2:

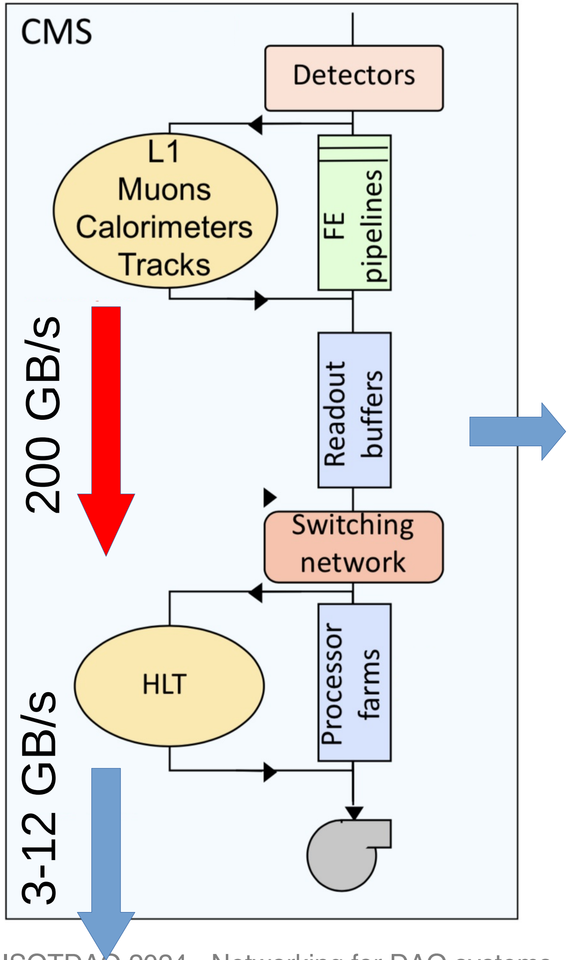
~90 PCs
Readout Unit (RU)

~70 PCs
Builder Unit (BU)

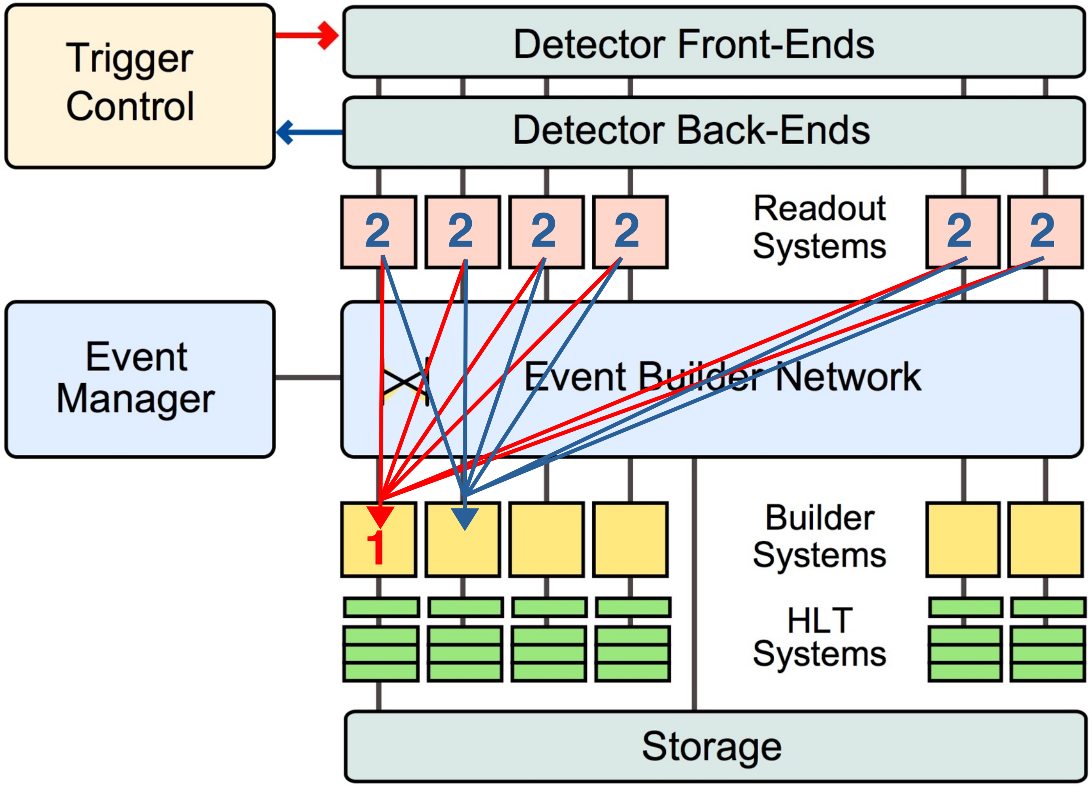
~1100 PCs
(~32k CPU cores)

Event building: Fragments from all RUs have to be read in order to assemble a single event in a single BU (the process is happening in parallel for all BUs)

DAQ to 1st order, Event Building



Parallel readout, linear event building



Run-2:

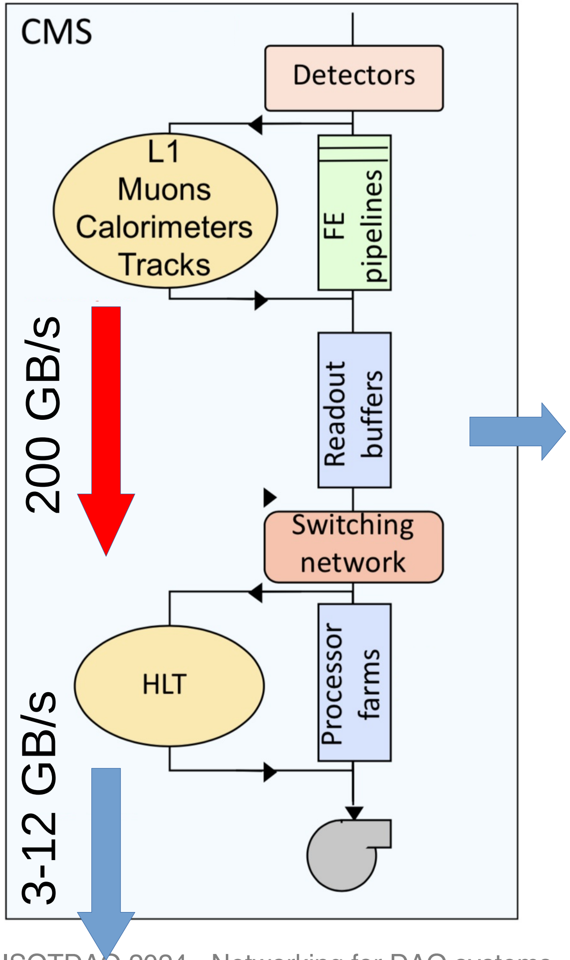
~90 PCs
Readout Unit (RU)

~70 PCs
Builder Unit (BU)

~1100 PCs
(~32k CPU cores)

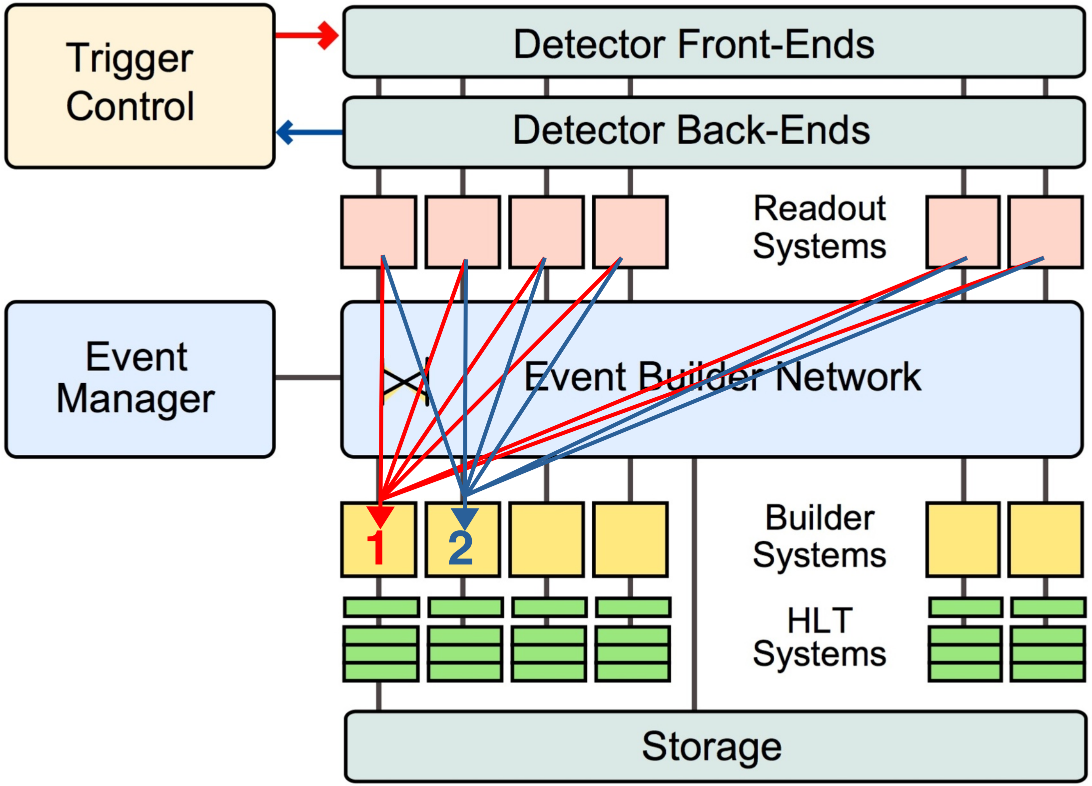
Event building: Fragments from all RUs have to be read in order to assemble a single event in a single BU (the process is happening in parallel for all BUs)

DAQ to 1st order, Event Building



Parallel readout, linear event building

Run-2:



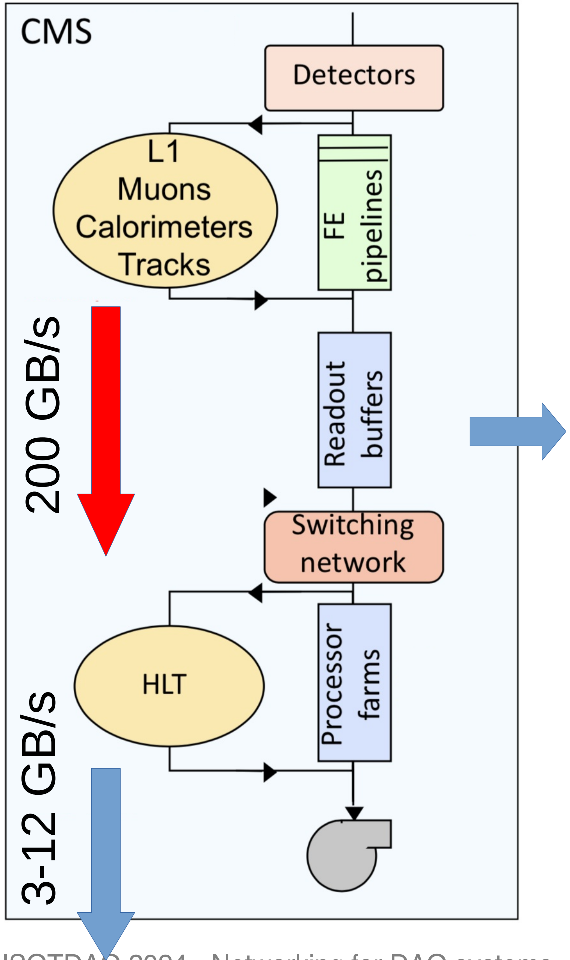
~90 PCs
Readout Unit (RU)

~70 PCs
Builder Unit (BU)

~1100 PCs
(~32k CPU cores)

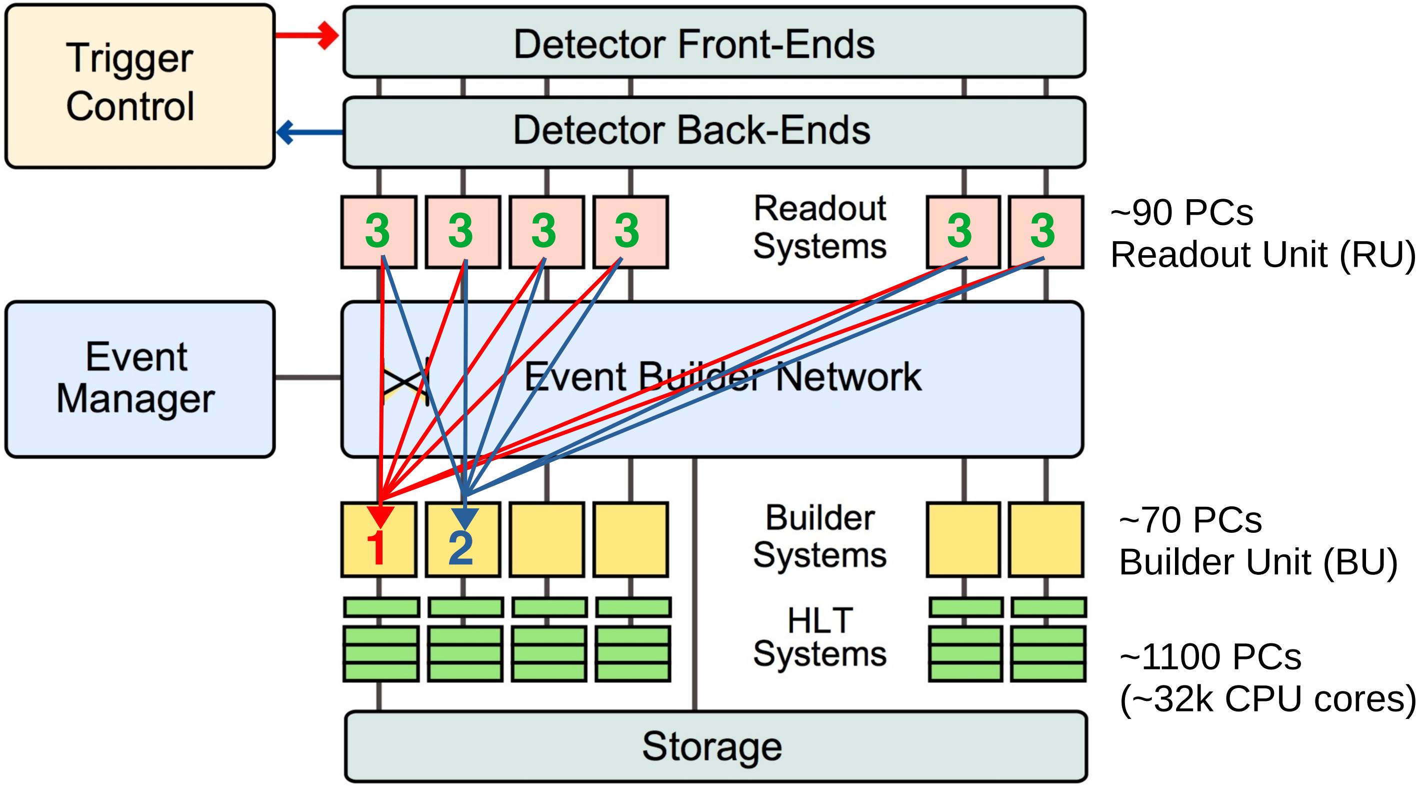
Event building: Fragments from all RUs have to be read in order to assemble a single event in a single BU (the process is happening in parallel for all BUs)

DAQ to 1st order, Event Building...



Parallel readout, linear event building

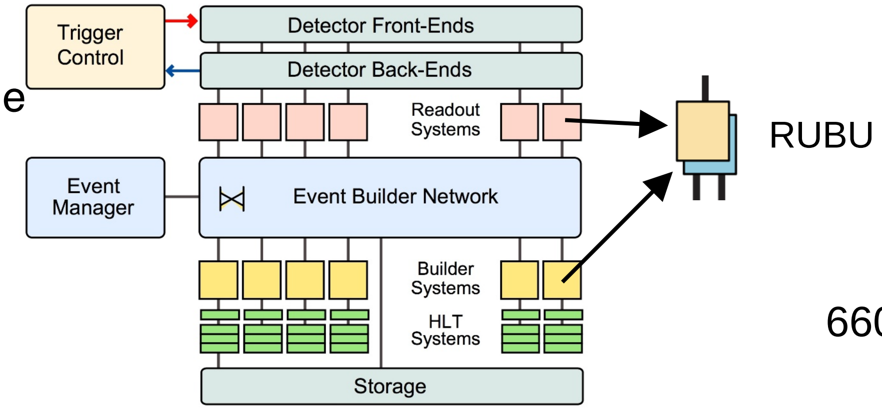
Run-2:



Event building: Fragments from all RUs have to be read in order to assemble a single event in a single BU (the process is happening in parallel for all BUs)

DAQ Performance Considerations

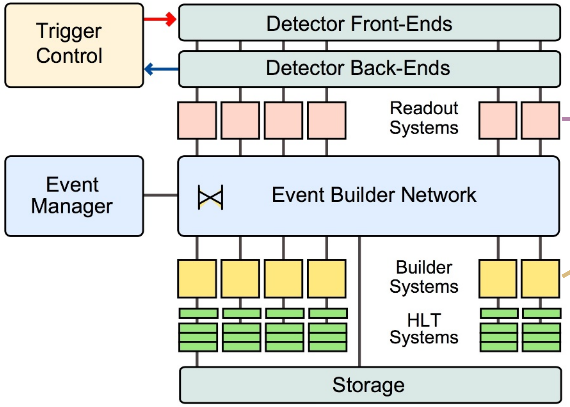
- Traffic from detector back-ends is synchronous
  - Packets with fragments sent at the sent time
  - Causing temporal buffer overflow (= congestion) in the switches
  - One solution is to use deep buffer switches
- Fragments have small size (1-2 KB)
  - Need to be sent in larger blocks for good network throughput/efficiency
  - TCP/IP fills the entire MTU (jumbo frames used)
- For Run-3 we built 100 Gb/s network with performant servers
  - Allowed to merge functionality of RU and BU into a single node RUBU
  - Folded event building

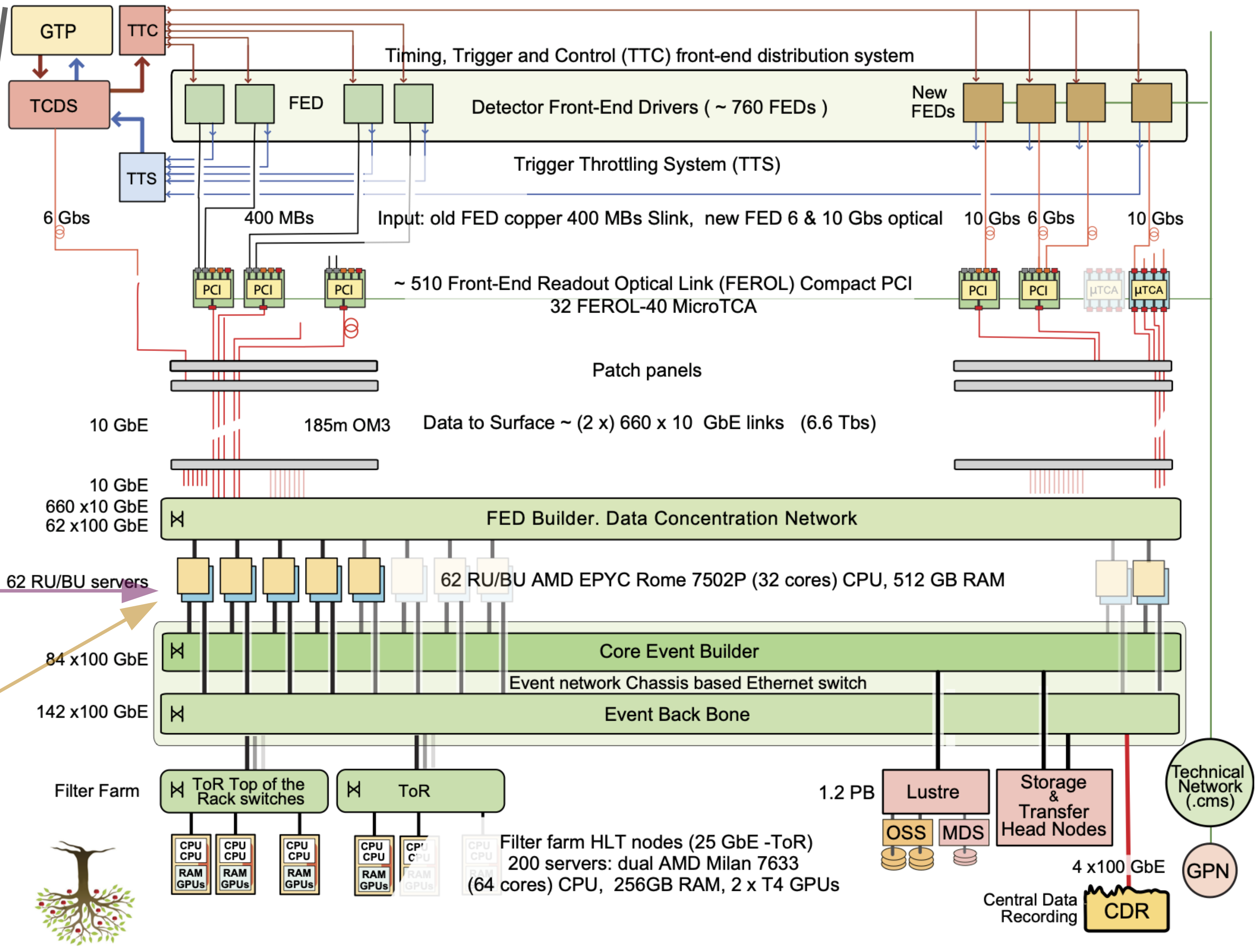


660x 10 Gb/s and 62x 100 Gb/s Data Concentrator Switch

DAQ for Run-3

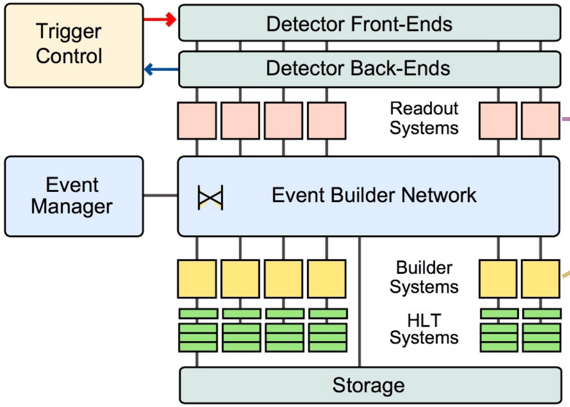
DAQ system in “more” realistic view

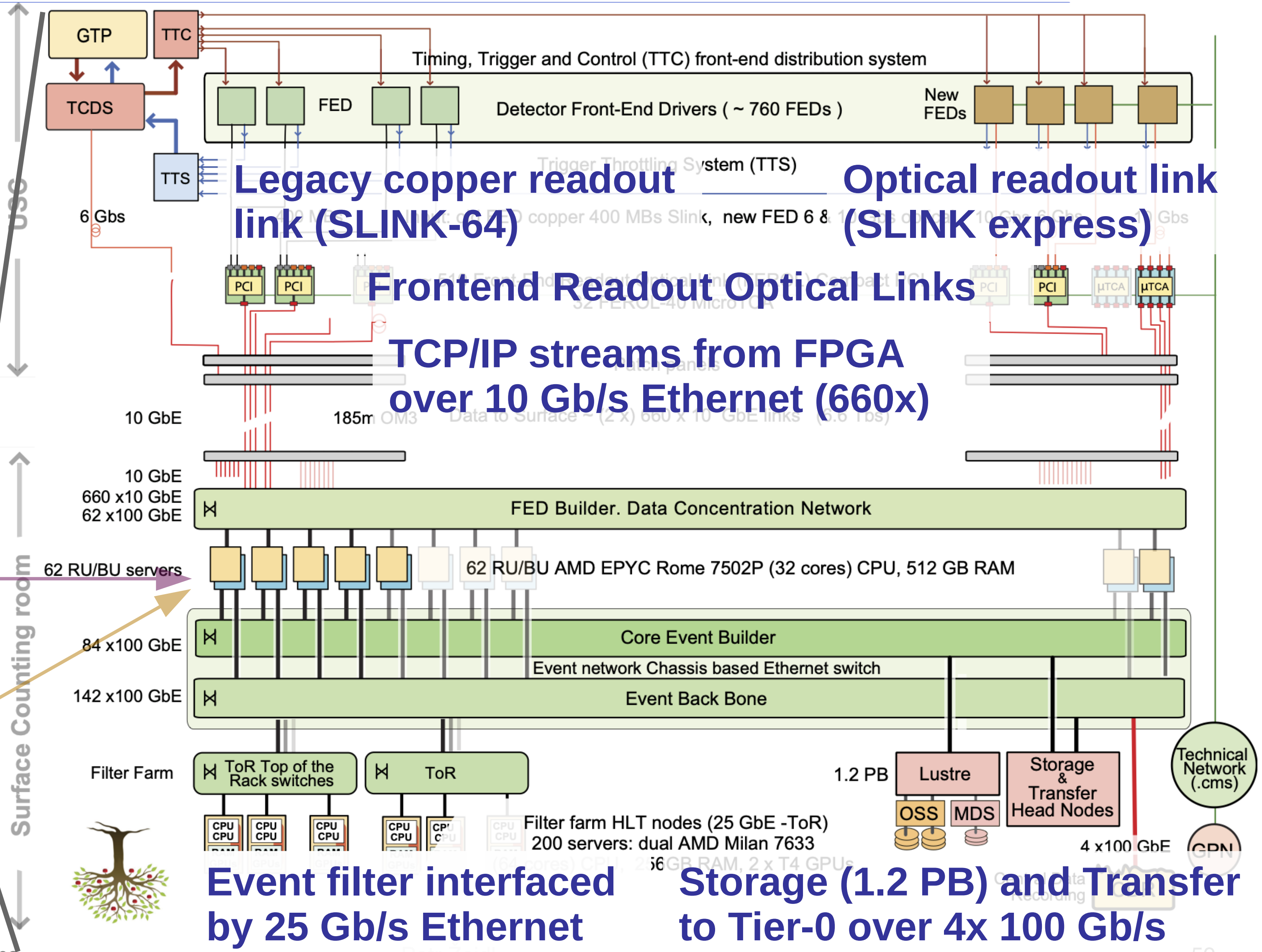




DAQ for Run-3

DAQ system in "more" realistic view





CMS DAQ

- Detector electronics is very specific and uses custom built protocols and links
  - DAQ is interfaced to detector read-out over a common interface (SLINK)
- Standard reliable protocol (TCP/IP) used to interface FPGA and transport fragments
  - Multiple 10GbE interfaces aggregated into 100GbE in the switches
  - Deep buffer switches used to absorb congestion caused by synchronous traffic pattern
- Building complete events requires fast reshuffling of memory buffers
  - Good job for RDMA (Infiniband or RoCE)
  - RoCE needs a dedicated loss-less network

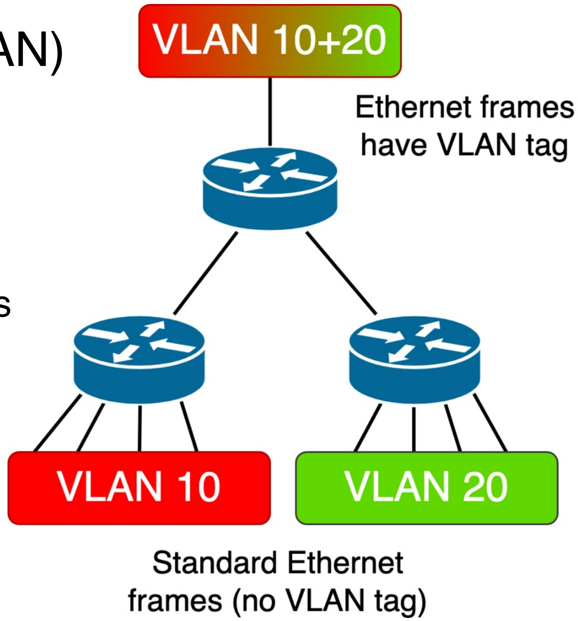
Summary

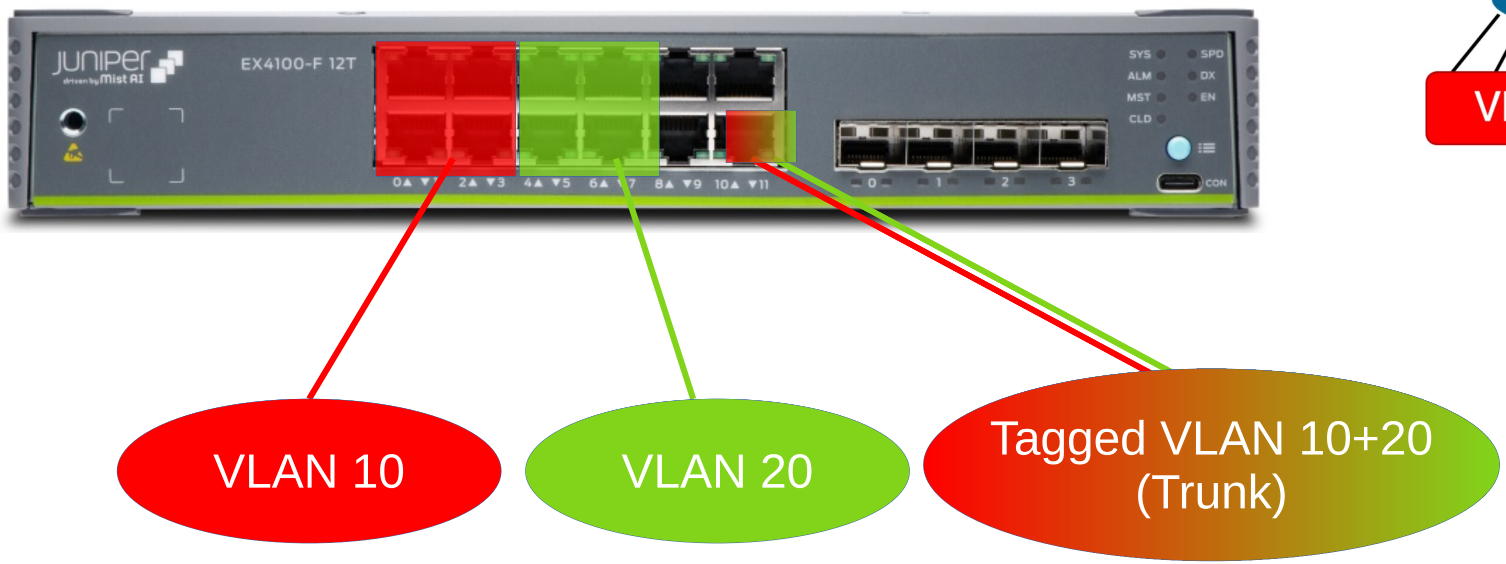
- Experiment data (events) are valuable!
- DAQ built from commodity network technologies and industry standards
  - Making development and maintenance more efficient compared to custom technologies
- But DAQ systems for physics have different requirements compared to e.g. campus networks
  - Long lived network streams without network congestion in DAQ vs many short lived streams in campus networks
  - Non-blocking and loss-less transport is often required (congestion free)
- DAQ shares some characteristics with HPC
- Performance and budget are important!
  - Need to know your protocols and networks to use them efficiently
  - Good monitoring tools help a lot
  - Good vendor support helps as well

BACKUP

Tagged Virtual LAN

- Managed switch can be configured to create a virtual LAN (VLAN)
  - Separate isolated network with its own broadcast domain
  - Using separate IP address map (subnet)
  - VLANs are static (fixed port assignment) or tagged (trunk)
    - Prioritization/Rate limiting can be applied between different VLAN tags





IPv4 Addressing Scheme

- IPv4 address is a 32-bit number – 4 294 967 296 addresses
  - Private networks ~18 million addresses
  - Multicast addresses ~270 million addresses
- Certain address ranges have special purpose

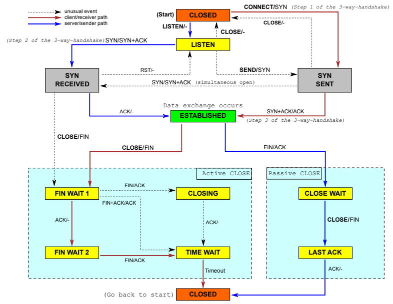
10.0.0.0/8	10.0.0.0–10.255.255.255	Private
127.0.0.0/8	127.0.0.0–127.255.255.255	Loopback address on the local host
172.16.0.0/12	172.16.0.0–172.31.255.255	Private
192.168.0.0/16	192.168.0.0–192.168.255.255	Private
224.0.0.0/4	240.0.0.0–255.255.255.254	Multicast
255.255.255.255/32	255.255.255.255	Broadcast

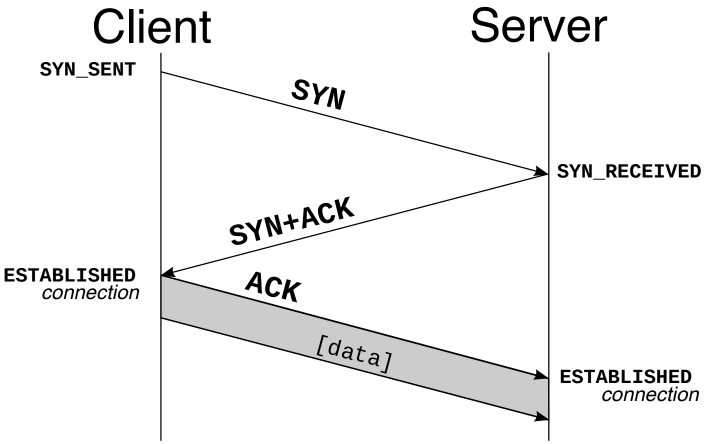
Sources:
- https://www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml
- https://www.rfc-editor.org/rfc/rfc1918.html

TCP provides connection oriented service/streams

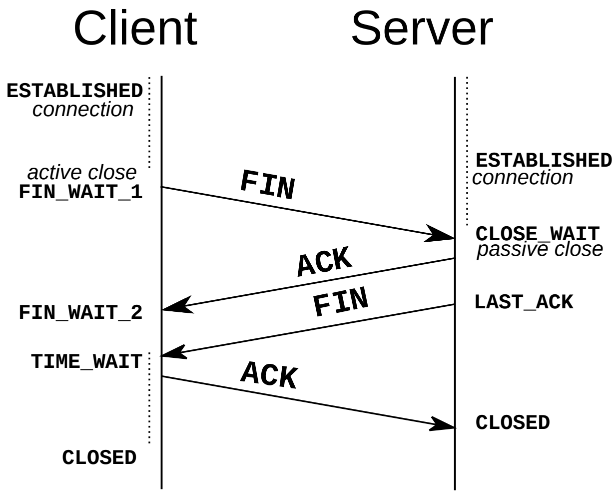
TCP protocol is stateful (11 states)

- Connection opening requires a 3-way handshake with the peer
- Connection closing requires a 4-way handshake with the peer
  - Each side of the connection terminating independently





Connection opening (3-way handshake)



Connection closing (4-way handshake)

Network Debugging and Monitoring

Network Debugging

- `ping` and `tcpdump` are our friends!

```
$ ping 10.177.128.56
PING 10.177.128.56 (10.177.128.56) 56(84) bytes of data.
64 bytes from 10.177.128.56: icmp_seq=1 ttl=64 time=0.118 ms
64 bytes from 10.177.128.56: icmp_seq=2 ttl=64 time=0.045 ms
64 bytes from 10.177.128.56: icmp_seq=3 ttl=64 time=0.037 ms
```

```
$ tcpdump -nn -i ens3f0
ARP, Request who-has 10.177.128.56 tell 10.177.128.57, length 46
ARP, Reply 10.177.128.56 is-at 0c:42:a1:79:86:e0, length 28
IP 10.177.128.57 > 10.177.128.56: ICMP echo request, id 7, seq 1, length 64
IP 10.177.128.56 > 10.177.128.57: ICMP echo reply, id 7, seq 1, length 64
IP 10.177.128.57 > 10.177.128.56: ICMP echo request, id 7, seq 2, length 64
IP 10.177.128.56 > 10.177.128.57: ICMP echo reply, id 7, seq 2, length 64
IP 10.177.128.57 > 10.177.128.56: ICMP echo request, id 7, seq 3, length 64
IP 10.177.128.56 > 10.177.128.57: ICMP echo reply, id 7, seq 3, length 64
```

- Timestamps are removed for better reading
- Tcpdump option `-e` also shows the MAC addresses

Network Monitoring

- **SNMP** (Simple Network Management Protocol)
  - Part of the Internet protocol suite since 1988, using UDP
  - Supported by all network devices
  - **Devices are actively polled** for variables (packet counters, errors, temperatures)

- **Telemetry Interface**
  - New **push model** based on Google protocol buffers
  - Network devices deliver data in periodic updates
  - Supports zero suppression
  - Eliminates polling

Monitoring variables in Grafana