Introduction to Networks in DAQ

Niko Neufeld
niko.neufeld@cern.ch

CERN

ISOTDAQ 2010, Ankara

Acknowledgments & Disclaimer

- Thanks to B. Martin for material on the ATLAS network
- Thanks to G. Liu and J.C. Garnier for comments and suggestions for an earlier draft of these lecture-notes

- Most of the material will be in parts familiar to at least some of you - I hope you discover some new angle
- In the same spirit I hope you can cope with a few "forward" references

Outline

Definition of a network

A network is a collection of independent devices, which can communicate as peers with each other

- **peer**: There are no masters nor slaves on a network[1]
- **independent**:The network exists as long as there are at least two connected devices

[1] Networks are for democrats!

Examples:

- The telephone network
- Ethernet (IEEE 802.3)
- ATM (the backbone for GSM cell-phones)
- Infiniband
- Myrinet
- many, many more

Note: some of these have "bus"-features as well (Ethernet, Infiniband) Network technologies are sometimes functionally grouped

- Cluster interconnect (Myrinet, Infiniband) 15 m
- Local area network (LAN) (Ethernet), 100 m to 10 km
- Wide area network (WAN) (ATM, SONET) $\leq$ 50 km
- Storage Area network (SAN) (FibreChannel) $\sim$ 100 m (for disk access)

Terminology

Every art comes with its own language.

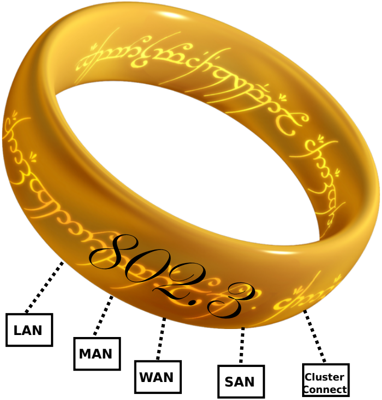
- **linkspeed**: The raw data transfer capacity of a physical link: also called bit-rate given in $\mathrm{bit/s}$ or bps
- **bandwidth**: data-transfer / second. Measured in $\mathrm{bit/s}$ or powers of ten(!): kilo, Mega, Giga, ...
- **octet**: synonym for byte (8 bits)
- **MTU**: maximum transmission unit, the maximum unit of data which can be transported as a single piece by a protocol (measured in bytes)

Terminology 2

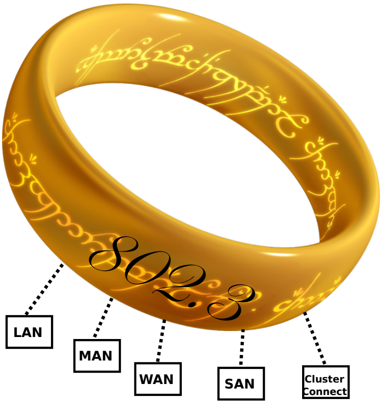
- **packet, frame**: synonyms[2] for a unit of data which is transported as a single piece
- **latency**: the time to transport a message between two points in a network (e.g. the forwarding latency of a switch is the time it takes for the packet to pass through the switch)
- **NIC**: Network Interface Card: the hardware part of a computer connected to the host-bus in charge of sending and receiving network traffic (nowadays usually not a separate "card")

[2] For the nerds: "frame" is used at the data-link layer (Ethernet) and packet at the network layer (IP))

One network to rule them all



One network to rule them all



Ethernet, a.k.a. IEEE 802.3, has become almost synonymous with networking.

More and more specialized networks are replaced by Ethernet or transported over Ethernet: Fiberchannel over Ethernet, iSCSI

Everything you want and do not want to know is in a 1000 pages document [1], in which you will find many words except one: Ethernet

The Ethernet frame

Preamble	Start-of-Frame-Delimiter	MAC destination	MAC source	802.1Q header (optional)	Ethertype / Length	Payload (data and padding)	CRC	Interframe gap
7 bytes of 10101010	1 byte of 10101011	6 bytes	6 bytes	(4 bytes)	2 bytes	46-1500 / 9000 bytes	4 bytes	12 bytes
		64-1522 / 9022 bytes						
84-1542 / 9042 bytes								

- Each device is identified by a 48 bit Media Access Controller (MAC) address (a.k.a. hardware address or Layer-2 address).
- The type length field is interpreted as a length of the frame in bytes for values below 1500 otherwise as the type of the protocol carried by Ethernet. The most famous number is 0x0800: IP.
- Note that in network protocol headers all numbers are big endian

For reference: endianess

Endianess[3] refers to the way a multi-byte number is stored in a byte-addressable memory. Example: take today's date as a number 20100205 or in hexadecimal notation $0x0132b46d$

Big Endian

Addr	01
Addr + 1	32
Addr + 2	b4
Addr + 3	6d

- also called IBM standard, used by PowerPC, Motorola CPUs, the XBox

Little Endian

Addr	6d
Addr + 1	b4
Addr + 2	32
Addr + 3	01

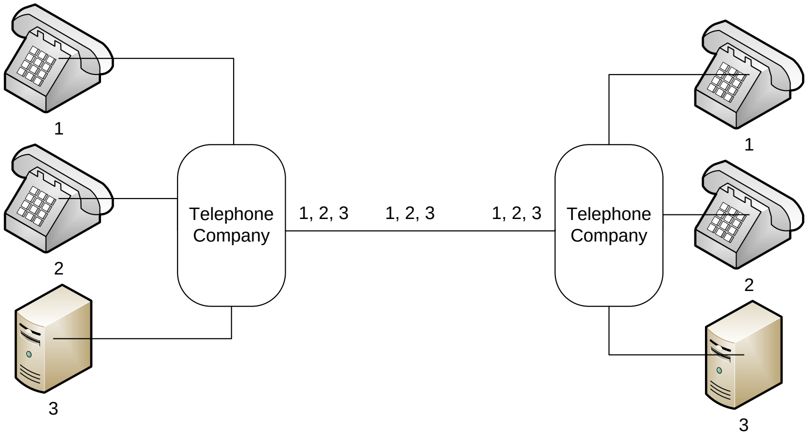
- also called non-IBM standard, used by Intel

[3] You will meet people who are fanatic about which is better. Remind about the origin of the term in Gulliver's Travels

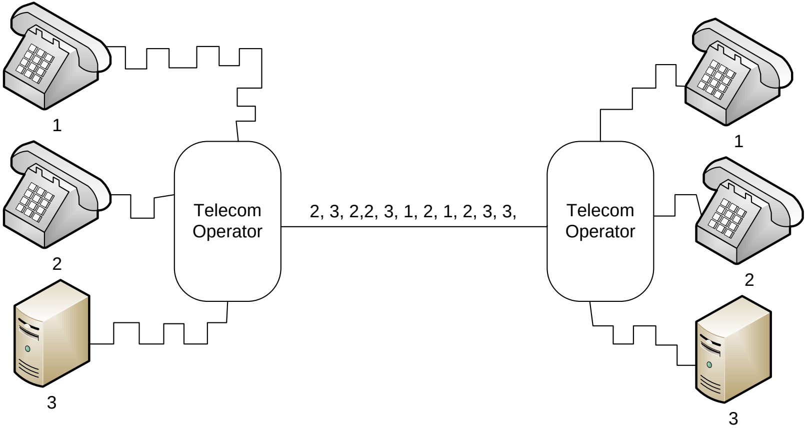
Moving data in a network

- Most modern networks do not use a shared medium anymore. They use *point-to-point* links.
- Devices connect to other devices either directly or via a *switch*
- The term *switch* comes from the world of telephony (c.f. switch-board)
- There are two main paradigms in switching: circuit-based and packet-based. Both correspond to important general communication paradigms in networking: connection-oriented and connection-less

Circuit based switching



Packet based switching



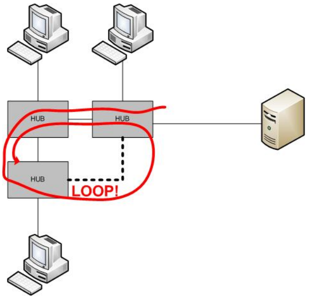
Ethernet switching

How does the switch know where to send a packet? By using a very simple algorithm

- A frame is received
- The source MAC address is added to the *MAC address table* at the entry reserved for the port on which the frame was received. This is how addresses are *learned*.
- The destination MAC address is searched in the *MAC address table*. If found the packet is sent to the port found in the table
- If not found the packet is sent to *all ports except the one on which it was received*[4].

[4] You cannot send to yourself!

Thou shalt have no loops!



- Ethernet devices have no idea about the network topology
- When there are loops sending a broadcast or any frame with an unknown MAC address will trigger an avalanche
- This can bring a network down!
- Modern switches have loop protection for example in the form of IEEE 802.1D (Spanning tree), but these are "heavy" tools. Better use VLANs (next slide) and careful design.

Broadcasts

- Normally a Ethernet host will only accept frames whose destination address matches its own MAC address[5]
- To send a frame to all devices on the connected Ethernet segment use a broadcast frame
- The destination address for a broadcast is FFFFFFFFFFFFFFFF, that $48 \times 1\mathrm{s}$
- Switches will re-transmit a received broadcast frame on all ports!
- Broadcasts the basis many configuration and discovery protocols: LLDP, ARP, DHCP, etc. . .
- There are also *multicasts* but they are much less used and not treated here

[5] MACs can be put into the so-called *promiscuous* mode, when they will accept any frame

Virtual LANs (VLANs)

- **The problem**: In a large network it is not good that broadcasts go throughout the network ("broadcast storm"). Or one wants want to create isolated networks on the same Ethernet for security reasons
- **The solution**: Create Virtual Local Area Networks (VLANs)
- VLANs can be tagged (then there is an additional 16-bit identifier field in the Ethernet header) to identify them, or untagged in which case they are identified by membership of ports
- No frame, not even a broadcast, will pass VLAN boundaries in a switch

Network Protocols

"Il semble que la perfection soit atteinte non quand il n'y a plus rien à ajouter, mais quand il n'y a plus rien à retrancher."

Antoine de Saint Exupery

"In network protocol design, perfection is achieved not when there is nothing left to add, but when there is nothing more to take away." [2]

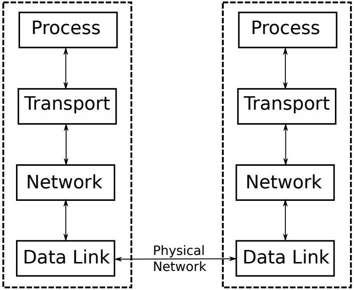
Protocols and protocol suites

7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

Figure: The OSI model

A (communication) *protocol* is a set of rules and conventions between communication participants. Layering helps to conquer complexity.

A simplified Network Model



- The link layer provides the physical interface to the communication medium (such as an Ethernet device)
- The network layer manages the movement of packets in a network
- The transport layer regulates the flow of packets between two hosts. It is accessed via a *socket*
- The app. layer sends and receives data on the socket

The Internet Protocol (IP)

Like most network protocols IP is defined in a Request For Comments (RFC) document, maintained by the Internet Engineering Task Force (IETF). These documents make interesting (albeit sometimes hard) reading.

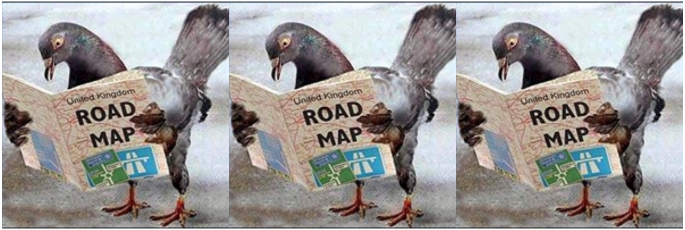
- There are two variants of IP: IPv4 and IPv6. We will be concerned only with IPv4 here - defined in [3]
- IP is connectionless and unreliable
- IP is designed to work on unreliable transport in a dynamic network (*no central management*)
- IP is designed to be encapsulated into transport layer protocols (in OSI language "data-link layer") there is IP over Ethernet, IP over WiFi, IP over serial lines and...

The Internet Protocol (IP

- most importantly:

The Internet Protocol (IP

- most importantly: IP over Avian Carriers (IPoAC) [4].



The IPv4 header

31 bits . . . 0

<table>
<tr><td>version</td><td>IHL</td><td>type of service</td><td colspan="2">total length</td></tr>
<tr><td colspan="3">identification</td><td>flags</td><td>fragmentation offset</td></tr>
<tr><td colspan="2">time to live</td><td>protocol</td><td colspan="2">header checksum</td></tr>
<tr><td colspan="5">source address</td></tr>
<tr><td colspan="5">destination address</td></tr>
</table>

- **time to live**: also known as "hop-count", integer decremented by each forwarding device in the network by 1 (limit loops)
- **type of service**: allows for differential treatment of traffic ("Quality of Service" QoS)

Subnets and subnet masks

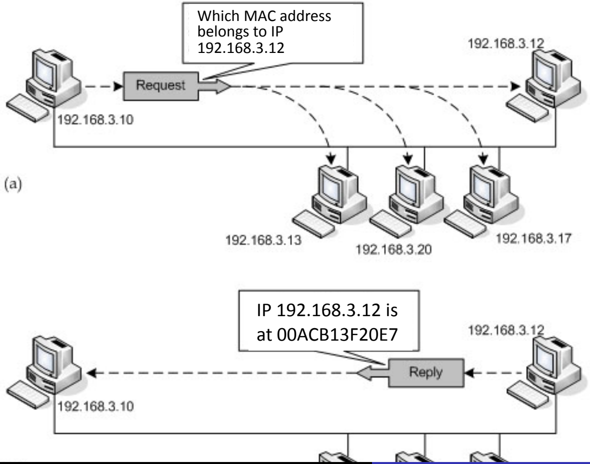
- The Internet serves to connect different networks
- The 32-bit address space is therefore global
- Different Local Area Networks connected must use different (disjoint) portions of the address-space: **subnets**

31 bits	...	0
Network prefix	host address	
Network prefix	subnet prefix	host address

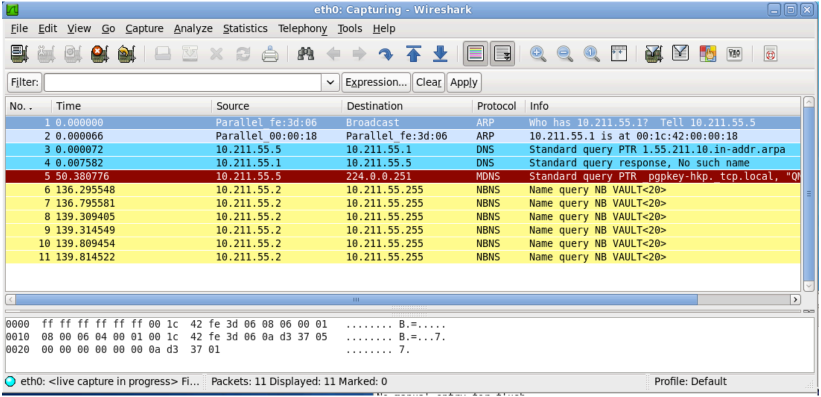
- Take the network whose *network-address* is 137.138.0.0
- We write: 137.138.0.0 `netmask` 255.255.0.0
  or 137.138.0.0/16. This is called a Class-B subnet

Running IP over Ethernet

Have an IP address, need a MAC address: enter the Address Resolution Protocol (ARP)



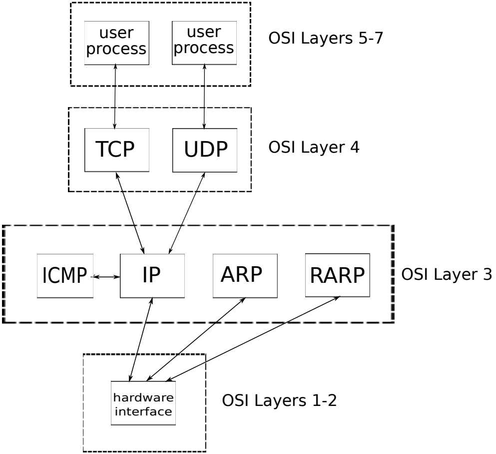
ARP in action



Transport protocols on top of IP

- **UDP** User Datagram Protocol. Messages (datagrams) of up to 64 kB. UDP is connectionless and unreliable: messages can be lost and arrive out of order. Messages are atomic. A message is transmitted and received in one piece.
- **TCP** Transmission Control Protocol. TCP implements a reliable, connection-oriented byte-stream. TCP has built-in congestion control and retransmission. This is the most used protocol of all: practically all known "Web" or "Internet" applications use TCP
- **SCTP** Stream Control Transmission Protocol. A new datagram oriented protocol combining features from UDP and TCP [5].

The TCP/IP protocol suite



The TCP/IP protocol suite (drawing from [6])

What the programmer sees: the Berkeley socket library

user-space	application code
kernel-space	network stack

- endpoints of network connections are opened with the `socket` call
- each socket comes with reserved buffers
- configuration via `setsockopt`
- can be read almost like a file with `read` and `write`. For optimal control under Unix use `recvmsg` [7]

Connecting networks: routing

- How do we connect to Ethernet LANs (or VLANs!) and in general send packets using IP between different networks?
- Routers[6] connect different IP networks.
- Routers forward between networks
- Routing decisions are either statically configured[7] or the result of learning using sophisticated *routing protocols* such as OSPF and IPIP
- Routers only look at the IP part of a packet. Any trace of the original Layer-2 (e.g. Ethernet) part of the packet is lost: for instance: all packets from a router will have the same MAC source address

[6] in the older literature sometimes called gateways
[7] often called Layer-3 switching

Efficiency

Figure: The Little Tin God also known as "Efficiency" [8]

Efficiency means using resources well. In network-ed systems we are concerned mainly with 2 resources:

- Bandwidth
- CPU and Memory on host-computers

Protocols for Data Acquisition: transport overheads

Each protocol layer will add some overhead to your network.

Protocol	bytes
Ethernet	40
IP	20
UDP	8
TCP	24

Example: Sending a 100 byte message using TCP/IP over Ethernet will use $40 + 20 + 24 + 100 = 184$ bytes on the wire. Efficiency $= \frac{100}{164} = 0.54$
Lesson: Make the payload as big as possible. Try to reach the MTU: i.e. a 1500 bytes[8] message $\frac{1500}{1584} = 0.95$

[8] Or better, use non-standard Jumbo frames of 9000 bytes, if all devices on your network support this

Bandwidth efficiency

Bandwidth eaters

- Protocol overheads (headers, trailers)
- Message rate
- Packet loss

Bandwidth savers

- Eliminate repetition and redundant information
- Reduce message rate by coalescence (packing several messages into one)
- Avoid packet loss due to congestion, make protocol tolerant to losses and/or minimize amount of retransmitted information

The Linux network stack

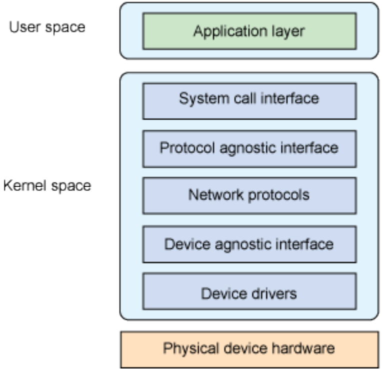


Figure: The Linux Network stack (from [9])

- In Linux packets flow from the device-driver to the sys-call interface in `sk_buff` structures.
- Transfer to and from the physical hardware from and to the `sk_buff` is done via DMA
- Layering provides efficient code-reuse and clean separation of protocols, *but* makes hardware off-load more difficult

Burning CPU cycles

- Networking is an expensive business for a host
- Checksums need to be calculated (often requires byte-wise access)
- Data need to be copied (memcpy)
- Headers need to be stripped off or added
- Protocol logic has to be implemented: the Linux TCP stack has several thousand (!) lines of code
- → partially addressed by hardware assists. In general the more expensive your NIC the more offload it will offer.

Zero-copy means transferring network data directly to and from the user application without any CPU intervention. In general this is not possible for Ethernet[9].

[9] Other technologies can do this, e.g. using remote DMA (Infiniband)

DMA and Interrupts

- **DMA** Direct Memory Access: the network card (in general peripheral devices) can read and write to the main memory of the computer without intervention from the CPU (saves a lot of CPU cycles!)
- **Interrupt** The network card needs to inform the CPU when the transfer is finished. It does this by sending an asynchronous signal (*interrupt*) to the CPU
- The CPU stops whatever it is doing and jumps to a special sub-routine (the interrupt handler)

DMA & interrupts 2

- This jumping is *very* expensive, because it breaks the current execution of the program and leads to cache-flushes
- On a heavily loaded computer the interrupt rate[10] can reach $O(100)$ kHz.
- A full-size Ethernet frame makes 12304 bits - that means a frame can come in every 12.3 μs

[10] check `/proc/interrupts`

DMA & interrupts 2

- This jumping is *very* expensive, because it breaks the current execution of the program and leads to cache-flushes
- On a heavily loaded computer the interrupt rate[10] can reach $O(100)$ kHz.
- A full-size Ethernet frame makes 12304 bits - that means a frame can come in every 12.3 μs
- We need to cut down the interrupt rate

[10] check `/proc/interrupts`

Interrupt moderation

Wait for several packets, buffering them in the card or DMAing them right away. Then notify (= interrupt) the CPU.

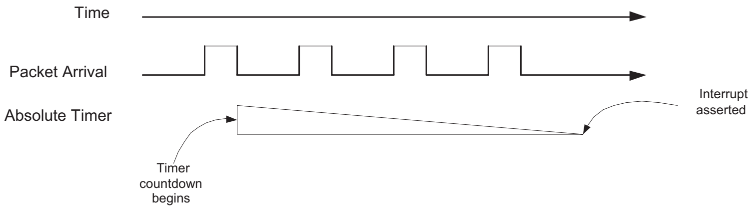


Figure: Absolute rx timer to reduce IRQs from ref. [10]

Interrupt moderation 2

Unfortunately, if for a change the traffic is light, each packet will incur on average half the latency of the moderation time. Not good for urgent packets like control messages. Add another time which fires after some inactivity

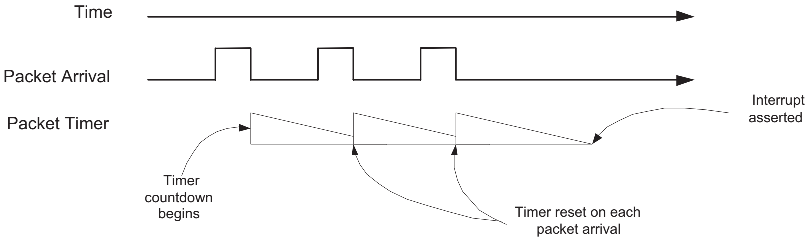


Figure: Receive packet timer to ensure timely delivery from ref. [10]

Tuning a server for DAQ traffic

- Receiving is harder than sending, in particular in a *push* protocol (c.f. lecture by E. Pasqualucci)
- In general provide for lots of buffers in the kernel, big socket buffers for the application and tune the IRQ moderation
- Examples here are for Linux, but can be done for M$-Windows if need be

```
/sbin/ethtool -G eth1 rx 1020  # set number of RX descriptors in NIC to max
# the following are set with sysctl -w
net.core.netdev_max_backlog = 4000
net.core.rmem_max = 67108864
# the application is tuned with setsockopt
```

Tuning the network devices for DAQ traffic

DAQ traffic is bursty and usually has a high average load. On the other hand there are not many different types of traffic. A DAQ network is usually “clean”.

- Examples here are shown for the HP Procurve family - your mileage may vary.
- Reduce the number of output queues, because each queue will get a minimum reserved amount of memory
- Enable jumbo-frames on all ports and VLANs. Nothing reduces interrupt rate and protocol overheads better than larger packets ☺.

```
sw-d1a03-d1> enable
sw-d1a03-d1# config
sw-d1a03-d1(config)# qos queue-config 2-queues
sw-d1a03-d1(config)# vlan 11
sw-d1a03-d1(vlan-11)# jumbo
```

Networks in the LHC DAQ

big core

small edge

- Large DAQ networks like the ones used for the LHC experiments need too many ports for a single device.
- They consist of a core, an aggregation layer and sometimes of a de-aggregation / fanout / edge-layer
- Use IP (routing) albeit mostly static

Aggregation and Trunking

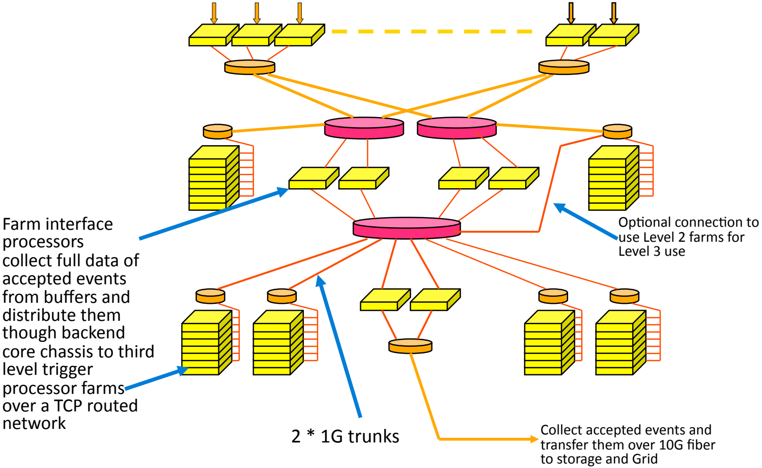
Trunking a.k.a link aggregation

- Addresses the need for a thicker pipe
- Allows “bundling” several links to one logical link
- Often used between switches
- Increases bandwidth and adds redundancy
- Defined in various standards LACP, 802.3ad

Aggregation layer

- Addresses the need for connectivity where only a limited total bandwidth to a group of devices is needed
- Use a (cheaper) switch to connect multiple hosts
- Use a fast link (or trunk) to connect to the core of the network. This link is called the uplink

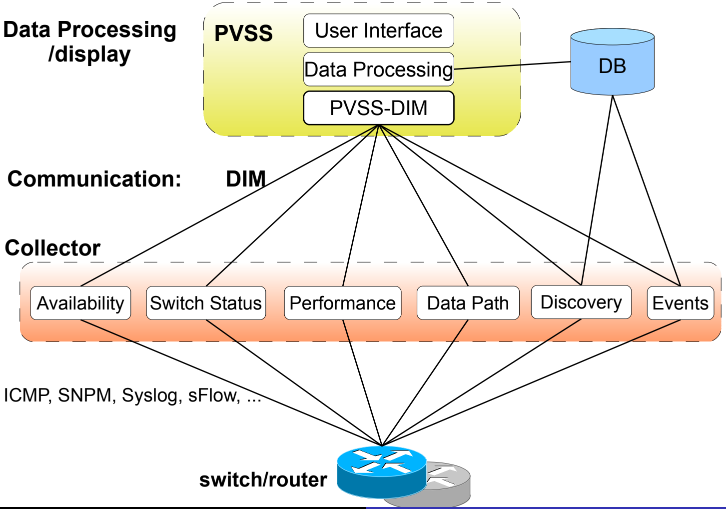
The ATLAS DAQ network



Monitoring and debugging

- Network monitoring consists of polling counters and watching out for exceptions. This is done using SNMP.
- Debugging often requires looking at (the headers of) the packets. There are two cases:
  - Detailed analysis of specific events: Wireshark (potentially using port-mirroring)
  - Statistical analysis of packets from many ports with high-speed (1 Gb/s and above) traffic: use sFlow or netflow.
- For integration of all this info there is a host of frameworks: from open-source (Nedi, Nagios) over proprietary (Spectrum) to home-made

Integrated network monitoring & control



This is the end...

- Networks are and will be the method of choice to transport large volumes of data
- Buses won't come back
- We have scratched only the surface of many topics: Ethernet and IPv4
- Efficiency in network treatment will remain important: Modern CPUs have no problem with 1 Gb/s but 10 Gb/s are not for free
- LHC and SLHC DAQ systems are / will be large specialized networks
- Many things to explore: remote DMA, Infiniband, etc...

Further Reading & References I

- Wikipedia is an excellent starting point for anything network related
- The Linux man-pages contain often very interesting details
- The RFCs can be very insightful once one has gotten used to the terse, nerdy style

[1] LAN/MAN Standards Committee, editor.
Part 3: Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.
IEEE Computer Society, 2008.

[2] R. Callon.
The Twelve Networking Truths.
http://www.ietf.org/rfc/rfc1925.txt.

Further Reading & References II

[3] Information Sciences Institute University of Southern California.
Internet Protocol DARPA Internet Program Protocol Specification.
http://www.ietf.org/rfc/rfc791.txt, 1981.

[4] D. Waitzman.
IP over Avian Carriers.
http://tools.ietf.org/html/rfc1149, 1990.

[5] R. Stewart, ed.
Stream Control Transmission Protocol.
http://tools.ietf.org/html/rfc4960, 2007.

[6] W. Richard Stevens.
UNIX Network Programming.
Prentice Hall, 1990.

[7] Marc J. Rochkind.
Advanced UNIX programming.
Addison-Wesley, 2 edition, 2004.

Further Reading & References III

[8] Henry Spencer.
The Ten Commandments for C Programmers (Annotated Edition).
http://geekhideout.com/c-ten-commandments.shtml.

[9] M. Tim Jones.
Anatomy of the Linux networking stack, 2007.
http://www.ibm.com/developerworks/linux/library/l-linux-networking-stack/.

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