The Infrastructure Stack

Ross Alexander

August 2017

Contents

Chapter 1

Introduction to the Infrastructure Stack

Infrastructure is generally defined as everything below the Operating System. Historically this has been the network, servers (aka tin) and storage. In recent years this has also evolved to include virtualisation. The Stack is therefore the three physical components, Storage, Network and Compute, with a virtualisation layer spanning above all three.

These talks will focus on the actual environments which we manage, specically the Gen1, Gen2 and BDC environments. These all have common infrastructure components.

1. Intel x86_64 CPU architecture.
2. IP over Ethernet.
3. Fibre Channel (FC) storage with SAS, SATA and SSD disks.
4. VMWare and Oracle VM hypervisors.

The Gen2 environment has several additional components.

1. FCoE (Fibre Channel over Ethernet) for storage
2. VXLAN for network vitualisation.

The terms of information and data processing each component has a specific purpose.

1. Storage is “Data At Rest”. Its primary purpose it to keep data safe from loss, mostly though redundancy.
2. Network is “Data In Motion”. Its primary purpose it to move data between compute, or between compute and storage.
3. Compute is “Data Manipulation”. It between the network and storage, manipulating data in a reproducible fashion.

Chapter 2

Networks

The "Network" is glue that holds everything together. Many of the basic concepts within networking cross over to storage and compute.

At its most basic form a network is a collection of nodes (called vertices in graph theory) and connections (called edges in graph theory). Networks (or graphs) have a "shape" or topology. A network topology is concerned with how the various elements are connected rather than their actual physical locations. A good example is the London Tube Map, which shows how all the lines are connected to each other but is only roughly based on real distances.

2.1 Types of networks

Networks can be categorised by a number of methods, but most common attributes are transmission medium and physical scale.

1. Data Centre network. The network we will be focusing on. This is always wired and is at the scale of a single data center. With the rise of high bandwidth WAN connections data centre networks now often span multiple physical locations.
2. Campus network. This is an extention of the historic Local Area network. Campus networks can cover a number of buildings and now often include Wireless LANs.
3. Wide Area Networks. These networks span geographically large areas, from city wide networks (often in subcategory of Metro networks) to intercontinental connections. Most WANs are provided by specialised providers but some of the Cloud-scale companies own and run their own WANs.
4. Wireless networks. Mostly 3G and 4G networks but also includes satellite networks and other high altitude devices.

2.2 History

To understand IP over Ethernet it is useful to go back in time and look at the telephone network. The original telephone network was a simple two node network with a single point to point connection, in this case a physical electrical circuit.

Local telephone networks grew not by connecting two phone directly but instead to connect each phone to a central office. Lines could be patched to form an actual circuit with a physical cable, forming an end to end circuit.

Extending the network further Central Offices needed to connect to each other. It became impractical to dedicate a physical wire for ever call between COs. In the original analog system voice channels were restricted to a 4kHz band. Each of these voice channels could be used to modulate a high frequency carrier (in much the same way as AM or FM radio) so multiple individual voice channels could be carried over a single wire. This technique is called Frequency Division Multiplexing. Lines carrying multiplexed circuits became known as Trunk circuits.

With the advent of transisters digital telephony emerged. Analog voice was digitized at 8bits per sample, and with a 4kHz voice band the sampling rate was 8kHz, giving a bitrate of 64kbps (note that is 64000 rather than 65536).

Unlike analog trunks digital trunks packed multiple voice channels together by giving each one a slice of time on a higher bandwidth channel. Specifically the US T (for Trunk) system specified that a frame containing 24 8-bit samples (i.e. a maximum of 24 voice channels) be sent every 0.000125 (i.e 1/8000) seconds, giving a total bitrate of 1,536,000 bps (actually 1,544,000 bps due to an additional framing bit).

As local telephone networks grew to become regional and then national networks the effecient routing of calls became intractibly difficult. To simplify the problem the network design became extremely centralised and hierachical. To ensure reliablility the top level nodes were required to be extremely complex and yet robust, making them extremely expensive.

Work in the early 1960s suggested this was a poor design for a Cold War Command and Control (C&C) network. Instead of a highly centralized network a much more mesh like network should be used with each connected node making their own routing decisions by "learning" the overall state of the network.

The facilitate this connections would need to be chopped into size limited messages, each with their own addressing, to be routed individually and then re-assembled at the far end. One consequence of this design was the ability to route around damage

so links no longer had to be completely reliable. This idea became the basis of Packet Switched networking and the idea that you could have a reliable network constructed from unreliable components.

This idea culminated in the creation of ARPANET in 1969, which would later evolve into the Internet. This will be covered later with the section on IP (Internet Protocol).

ALOHAnet

In the early 1970s developed a packet radio network to allow campuses on the outlying islands to connect to a central Time Sharing System. This was done UHF radio. However unlike most radio systems at the time all the radios shared the same frequency, known as a shared medium. Access to this was entirely distributed. Because there was no central control of access it was possible for two radios to try to send simultaneously. Each station had to determine if a collision had occurred and backoff, trying to resend a random interval later.

2.3 Ethernet

The ideas from ALOHAnet were translated from a broadcast radio medium to a shared physical cable during the early 1970s and became Ethernet. The basic concepts of passive medium with multiple access and distributed collision detection was kept. The Ethernet cable was effectively a shared broadcast bus, so that every station on the wire received the transmission, even if they were not the intended recipient.

Standard Ethernet II frame

The frame format confusingly has two standards. The original standard is called Ethernet II (aka DIX Ethernet for DEC, Intel & Xerox). In Ethernet II the two octets following the MAC addresses identify the payload and is called the Ethertype. However when the IEEE standardized Ethernet has 802.3 this field is the frame length field and and additional 802.2 LLC/SNAP header is then required for the Ethertype. As the received frame size (as seen by the NIC) the IEEE format is almost never used.

ThinNet

Early Ethernet implementations used coax cabling with taps for each station. Each end of the wire required a terminating resistor to avoid signal reflections. The maximum distance 185 metres with a bitrate of 10 Mbps and was standardized as 10Base-2.

The requirement that all end stations needed to be connected to effectively a single cable limited the number of stations and made cabling existing office building difficult. The development of multiport repeaters allowed the network to expand to multiple cabling segments. The repeaters effectively copied bits from each segment to all the others, expanding both the broadcast and collision domain across all the connected cables.

10Base-T and hubs

In the late 1980s there was work to utilize existing Unshielded Twisted Pair (UTP) cabling, common in building for telephony, for data transmission. The final result was the 10Base-T Ethernet standard, which was electrically compatible to 10Base-2 but was no longer a shared bus. Instead it was a point to point connection. This required an active component in the network between all the end stations. Called a hub this resulted in a star topology and acted much like multiport repeaters in 10Base-2. It is possible to connect a 10Base-T hub to a 10Base-2 segment (and was often done).

Bridges

While hubs and repeaters allowed Ethernet Local Area Networks (LAN) to grow to several hundred stations the limitation of a single shared medium remained, so that the 10Mbps bandwidth had to be divided between all the competing senders. Also as more stations were added the chance of a collision increased and broadcast protocols consumed an increasing amount of the available bandwidth. In addition 10Base-2 networks were prone to faults, which often caused the entire network to fail.

To improve reliability of these large Ethernet segments the concept of Ethernet bridging was developed. An Ethernet station was connected to two (or more) seperate Ethernet segments. Based on the OSI model these are called L2 (Layer 2: Link) bridges, with each attached network a L1 (Layer 1: Physical) network.

In its simplest form bridges receive an Ethernet frame on one interface and sent it out on all other interfaces. This store and forward method improved fault and collision isolation but did not solve the shared bandwidth problem. To improve this bridges evolved to “learn” where each end station was on the network by recording the sender MAC address (every Ethernet interface has a 48bit Media Access Control [MAC] address) and what interface it came in on. Using the destination MAC address the bridge would then forward the frame to the L1 network where the end station with that address resided. If it didn’t know the destination or it was a broadcast address then the bridge would “flood” the frame to all other interfaces. This technique is called L2-Learning and is still fundamental to all Ethernet switches.

Interlute for 100Base-TX and later standards

In 1995 the 100Base-TX standard was released, increasing the speed from 10Mbps to 100Mbps. The standard supported the use of hubs but the increasing processing power had allowed for the creation of high port count bridges, known as Ethernet switches. Switches allowed for individual ports to run at different physical settings (such as speed and duplex). With the arrival of Gigabit Ethernet, initial over Fibre and later over copper, switches would often have a small number of interfaces which supported the faster speeds. These ports are often known as Uplink ports, from their use in tree topologies.

Spanning Tree

With the advent of gigabit Ethernet all physical connections became full duplex point to point connections, both on copper and fibre. To maintain the fiction of shared medium broadcast switches employed L2 learning.

Every since the invention of repeaters the issue of Ethernet loops existed, however it didn't practise due the way cabling was generally run. With bridges the problem became much prevelent. If a loop is created within a bridged network then any broadcast (or flooded unicast) frame will loop around the network forever, as each bridge forwards it another bridge over one link, only for it to be forwarded back to the original bridge over the second link.

This behaviour became known a broadcast storm and should it happen will cause the entire L2 network to effectively melt down and become unusable.

The solution was to detect and block links that would create a loop. A topological structure known as a spanning tree is used. In STP (Spanning Tree Protocol) a root node is elected and this becomes the root of the tree. All other nodes have only one path to the root. The tree is a spanning tree because every node is reachable from the root. All links between bridges not part of the spanning tree are blocked. A spanning tree is a guarenteed to be loop free but may not be optimal (the choice of the root makes a big difference).

Link Aggregation

Many networks designs a based on a tree topology to specifically avoid the possiblity to of loops and therefore didn't need to rely to STP. The standard reference design became the three layer three, known as Core / Distribution / Access. In many ways this looks like the traditional telephone network.

However to maintain redundency multiple links between layers is required. To avoid these forming a loop these links are bonded together so that the L2 (and STP) see them as a single link. This increased the amount of bandwidth available while avoiding worrying about STP going wrong. The Link Aggregation Control Protocol (LACP) is used to signal between switches (or multiply connected hosts) to form a bonded link. For this to work each switch needs run a control process. When the links are going between multiple physical switches then the switches are end need to aware of each other, normally in a form of cluster. This is often called MultiChassis LAG and used in the Gen2 cross-site Data Center Interconnection (DCI).

VLANs

There are many cases where there is a need to seperate and isolate traffic. This would have once required seperate physical networks to be constructed. With the advent of switches and the end of physical shared media it became possible to put interfaces into seperate forward domains. These became known as Virtual LANs. Within an single switch VLANs are fairly trivial, with each interface being assigned into one or more VLANs.

When traffic needs to go between switches then additional information needs to be send with the frame to indicate which VLAN that frame belongs to. This is done by inserting an additional header into the Ethernet frame, known as a VLAN tag (often called a dot-q tag from the 802.1q IEEE standard). Known as a shim header, it is inserted into the Ethernet frame after the sender MAC address (the standard Ethernet header is constructed of the destination MAC, the sender MAC and a 16-bit Ethertype field).

Standard Ethernet II frame

By using 802.1q Ethertype the switch recognises the VLAN shim. This is a total of 32 bits in size, 16 bits for the Ethertype, 4 bits for class of service (which nobody ever uses), leaving 12 bits for VLAN ID. The VLAN IDs of 0 and 1 are not available for general use, leaving 4094 possible VLANs. The addition of four octets (network speak for 8-bit byte) pushes the Maximum Transmission Unit beyond the original Ethernet standard of 1518 (counting the CRC). Frames larger than 1518 octets are known as jumboframes. Where a frame is only slightly larger it is often called a baby jumboframe.

2.4 Routed networks

In the OSI model the next layer about Link is the Network layer (L3). When it was designed there were many different link layers, from Ethernet, FDDI and Token Ring to serial connections over a multitude of different cables or radio systems. Today almost all L2 links are form of Ethernet. The Network Layer covers end to end transmission of packets, and may use multiple L2 networks on the way.

Addressing and location

Addressing in networks serve two functions, identity and location. Identity says who the end point is, the locations indicates how to reach them. Often a full address is constructed of both an identity and a location. An example of a purely identifying address is an IEEE MAC address. Every physical NIC has a globally unique MAC address (theoretically). This will guarentee uniqueness of identity. However a MAC address gives no indication of location.

This is known as a flat addressing space. On a local scale it is possible to build a large networks using just MAC addresses (L2 forwarding), but it eventually they collapses under their own weight as the number of tables entries the forwarding elements must hold and number of changes becomes too much.

For networks to scale it necessary to create an addressing scheme which hides information. The simplest form after a flat address space is two level scheme. In this scheme addresses are broken into a network address and a host address. The IP, IPX and AppleTalk are protocols that do this. Possibly the most extreme case of explicit address hierachy was the OSI GOSIP addressing, which had nine components (it was effectively a tri-level addressing scheme but it never took off). By splitting up the address only those elements are their respectively layers need that part of the address.

Most addresses are of fixed bit length, for example IPX has a 32 bit network ID and a 48 bit host address, while AppleTalk has a 16 bit network ID and 8 bit host address. IPv4 (and IPv6) differ in while their total address lengths are fixed (32 and 128 bits respectively) the split between the network and node addresses is not fixed, but is indicated by an explicit prefix length. However this was not originally the case.

IP

The Internet Protocol (IP) grew out of ARPANET. One of the major design changes was reliable transmission was split off from network function of forwarding packets and given to end hosts, with routers (aka gateways) only responsible for forwarding.

IPv4 addresses are 32 bits wide. In the original design three main classes, named simply as A, B & C. The first bits of the address indicated the class. Class A addresses had an 8 bit network address, class B addresses had a 16 bit network address and class C addresses had a 24 bit network address. With various exclusion there are 126 class A addresses, 16384 class B addresses and 2097152 class C addresses. This was designed to give the reasonably effecient allocation while maintaining some flexibility.

Subnets and CIDR

In practise the address class system turned out to be the wrong size for almost everybody. The early internet was still mostly based around academic and scientific institutions. Class A was too big and there were only a few anyway, class C was too small, with only a maximum of 254 hosts. Class B networks were the right size for most institutions but did not fit well into their internal networks.

Within these institutions there were many different local networks, and most LANs could not suppose more than a few hundred nodes. The quick and dirty solution would be to break the class B address into 255 class C networks. This number of nodes matched the size of LANs at the time and 255 networks be plenty for even large institutions like a major university.

This involved a considerable change to both the hosts and routers. The solution was to assign a 32 bit mask to each network. This mask would be used to mask (using bitwise and) the network portion of the address. In most cases the mask for a class B network was 255.255.255.0, so the network was 24 bits long and the host 8 bits (the same size as a class C but now technically classless). Hosts then needed to know their subnet mask and routing protocols needed to pass around both the network address and its mask.

The advantage to this scheme is was the institution still only advertised its class B address, so while subnetting (as it is called) was done within an internal network it was hidden to the rest of the internet. This effectively created a three level hierachy for most IP routing.

In the early 1990s subnetting was replaced with a more general standard called Classless Inter-Domain Routing [CIDR]. Specifically the subnet mask, which could (in theory) be arbitary, became a left continuous prefix only (so there are only 32 possibly prefixes/masks). This allowed addresses not only to be subnetted, but also aggregated together. By doing this large blocks of historic class C addresses could be allocated to ISPs as a single prefix within the global routing table. It also removed a large number of edge cases which the original subnetting introduced simplified routing to Longest Prefix Match routing.

IP over Ethernet and ARP

For IPv4 (and IPv6) to work over Ethernet it is necessary for each host to resolve local (i.e. on the same subnet) addresses. This involves finding the Ethernet MAC address of a particular host. In IPv4 this is done using a helper protocol called Address Resolution Protocol (ARP). ARP uses a different EtherType (0x0806 vs 0x0800) to IPv4 itself. It relies on local broadcasts to flood the ARP request to all the other hosts on the network. The request contains the senders IP and MAC address along with the IP address it wants to resolve. If a host has that address (or a router doing proxy ARP) then that host will reply with MAC address for that address.

In IPv6 ARP has been integrated into ICMPv6 as Neighbour Discovery (ND) but functions in a very similar fashion. The ARP table (or ARP cache) is fundamental to working of IP over Ethernet. In modern switches ARP snooping can be employed to do various tricks, including broadcast supression over DCI links.

2.5 Tunnels, VPNs and Overlay nteworks

In the OSI model there is a strict layering, and this is suppose to be reflected in the structure of data streams (at L5 and above) or the actual packets (for L1 - L4). So for HTML over TLS the TCP stream consists of the initially the TLS connection details, then the HTTP headers then the HTML. At the packet level the TCP packet is encapulated inside an IP packet, which in encapulated inside an Ethernet frame, which is encapulated in a PHY level bitstream.

There are instances where it is necessary break this strict hierachy. when this is done by packet/frame/PDU from lower or equal layer and embedding into another packet/frame/PDU is often described at layering violation. The general description for this is called tunnelling.

Application tunnelling

Many TCP based applications have their own network protocol stack and often don't support encryption. Rather than use network level encryption (such as IPSec) the stream in via a pair of proxies that accept in inbound connection, create a new outbound connection using TLS and then forward the stream through as required. This allow insecure traffic to traverse an untrusted network (such as the Internet) without needing to modify the application itself.

Network Level tunnelling

With network level tunnelling a packet or frame (for IP and Ethernet respectively) is embedded a the payload of another packet or frame. Because of the additional headers tunnelled packets are larger than the standard 1514 octets. Either the base MTU is deceased, the link MTU needs to be increased, or the embedded packet is fragmented.

Ethernet Q-in-Q

This is an extension of the existing 802.1Q VLAN protocol and provides additional levels of VLAN tags. Is allows networks to scale to more than 4094 VLANs. The main disadvantage is does not hide the original MAC addresses so core switches need to know every MAC address so the L2 FIBs need to be large enough to have an entry for every host on the network. Another problem in environments with virtual machines is VM MAC addreses are not gobally unique so this is a chance in a large network two VMs will have the same MAC address.

Ethernet MAC-in-MAC

With MAC-in-MAC an Ethernet frame in embedded within another Ethernet frame. The 802.1AH standard defined an additional header between the two frames. Is allowed for a new 24 bit Service ID (I-SID). This standard is known an Provider Backbone Bridging (PBB). The use of I-SIDs allows for multi tenancy (up to 24M I-SID values). Because the customer Ethernet frames are embedded inside a provider Ethernet frame the core switches do not see any of the customer's MAC addresses or VLAN tags. This allows for very large scale Ethernet networks called Metro or Carrier Ethernets.

Ethernet Shortest Path Bridging (SPB)

This is an extension of PBB that uses the OSI IS-IS protocol to create an a full topology map of an Ethernet network. This allows loop free Equal Cost MultiPath (ECMP) forwarding tables. In an ECMP topology if there any multiple paths of the same minimum cost they are treated like an LACP group and traffic is hashed out over each of the links.

IP-in-IP

With IP-in-IP an IP packet is embedded inside another IP packet without any additional transport header. With IPv6 there are four combinations of IP-in-IP, 4in6, 4in4, 6in4 and 6in6.

6in4 was common in the early days of IPv6 where carrying IPv6 traffic over IPv4 was necessary. There are many variations on this theme for the IPv6 transition but with most major carriers now supporting IPv6 natively the reverse (4in6) is now becoming more popular, where carriers have converted their core networks to be IPv6 only.

IPSec tunnelling

IPSec has two modes of operation. In transport mode only the payload is encrypted. With tunnel mode a new IP header is prepended to the packet, with the original packet as the encrypted payload. This is the basic of all IPSec VPNs and allows private IP addressed traffic to cross the internet.

Generic Routing Encapsulation (GRE)

GRE was developed by Cisco as a way of running routing protocols over VPNs tunnels with IOS based routers. It consists of a 32 to 128 bit header which follows IP header. It can be keyed to allow multiple tunnels between the same endpoint pairs. A limitation of GRE is only that only point to point.

VXLAN

VXLAN is a UDP based encapulation protocol. In terms of extra headers it is fairly heavy weight but use of UDP allows for the injection of entropy for ECMP hashing. XVLAN is specifically designed to carry Ethernet frames over IP. It differs from most other tunnelling protocols in that is explicitly supports multicast.

VXLAN has a 24 bit VNI (Virtual Network Identifier), which plays a similar role to the VLID in 802.1q VLANs. Some implementations rely on IP multicast to do flooding while others use host based replication and unicast.

VTEPs

VXLAN Tunnel Endpoints are switches (virtual or physical) that can encap/decap (encapsulate / decapsulate) VXLAN packets into Ethernet frames. After decap a frame is then feed back into the network stack from a virtual interface. This is called recirculation. Some hardware switches avoid recirculation by adding specific encap/decap stages to their processing pipeline.

Overlay networks

When VXLAN (or any other tunnelling protocol) is coupled with an automated control system (called the control plane) then it is often described as an an overlay network. The current vShield implementation uses a distributed control plane and IP multicast to work. With NSX an centralized controller is required and can work without the need for the underlay network to support IP multicast.

2.6 Optical Networking

Optical networking is the use of light (normally lasers) to transmit data over plastic or glass fibre. In practise all optical networking uses laser light with glass fibres. Glass fibres contain two strands of glass, one inside the other. The inner strand is called the core while the other strand is called the cladding.

There are two classes of fibre, both with a cladding diameter of 125 microns. For short distances multimode fibre can be used. It is standardized as OM3 and OM4 (for Optical Multimode) and has a core diameter of 50 microns. For long haul fibre the core size is around 9 microns. This is called single mode fibre and is standardized as OS1.

In simple terms with single mode the narrowness of the core only allows a single path for the light to travel, where as in multimode fibre the light can bounce around within the core with more than one path. This greatly limits the distance can travel before it is too attenuated to detect but allows for relatively simple and cheap transmitters and receivers.

As a result Ethernet and Fibre channel over multimode fibre is restricted to approximately 300M for OM3 and 400M for OM4 at 1Gb and 10Gb and 75M and 100M for 25Gb on OM3 and OM4 respectively. Therefore multimode fibre is restricted to data centre and campus networks.

Chapter 3

Compute

3.1 IBM PC (October 1981)

All x86 systems are derived (however distantly) from the original IBM PC. It did have five channel I/O slots on an expansion bus (achronistically called the ISA bus). The original system is fairly modest and the base system required a minimum of a video card (either MDA for monochrome or CGA for colour) and a floppy controller. Optional ISA cards included parallel port for printers, serial controller for modem (or any other serial device) and a number of different harddisk controllers. Early harddisks each had their own different controllers but the Shugart (later Seagate) ST-506 was a semi-standard.

1. 4.77MHz 8088 (8bit data bus, 20bit address bus)
2. 16K memory, maximum 256K
3. 5 ISA slots
4. DMA controller
5. PIC for interrupts
6. PIT timer
7. KB, casette and speaker

3.2 IBM XT (March 1983)

The five ISA slots proved to be too few so the XT model was released with 7 slots. It also had more base memory and 10MB harddisk as standard. Memory could be updated by either ISA card (since the ISA bus was shared with the RAM) or by replacing the memory chips, provided they were socketed rather than soldered on.

1. 4.77MHz 8088
2. 128K memory
3. 7 ISA slots
4. Serial adapter
5. Floppy controller
6. 10MB MFM HDD
7. Additional memory by individual chips

3.3 IBM AT (August 1984)

The first major upgrade came with the IBM AT. The Intel 80286 had a 24bit address bus, allowing access up to 16MB of memory. However DOS had already set the maximum size of an running process to 640KB. Various methods (EMS & XMS) could be used to get around this limit but they were non-trivial.

The ISA bus was extended to 16bits wide and reclocked at 6MHz. To handle the increased number of I/O cards the number of DMA and PIC (Programmable Interrupt Controller) chips were doubled. The cassette interface was replaced by a Real Time Clock, which with a battery backup, would hold the currently date and time.

1. 6MHz 80286 (16bit data bus, 24bit address bus)
2. RTC
3. Double PIC
4. Double DMA
5. EGA graphics
6. 256K base memory

3.4 Super I/O and SIMMs

Improved fabrication methods allowed multiple functional blocks (along with internal buses) to be placed on a single silicon die. This gave rise to Super I/O ISA cards with multiple functions on the one card. Not only did these reduce the number of slots required but were cheaper overall.

Individaul memory chips were difficult to handle. Improved fabrication made them small enough to be placed on a small PCB to form a memory module. Early SIMMs (Single Inline Memory Module) where only 8bits wide so normally had to come in quads.

1. Integration of UART (serial), parallel, floppy and game onto single ISA/AT card
2. Additional memory changes from individual chips (SIPP) to multiply chip modules (SIMM)

3.5 Intel 80386 (1985)

The Intel 80386 was a leap forward in capability over the 286. By extending the internal registers to 32bits wide and implementing a new flat memory model with paging virtual memory it was able to address a maximum of 4GB and support true process memory isolation.

Released at the same time was a caching memory controller. The memory was moved off the ISA bus to the memory controller, which could also access much faster, but less dense, SRAM. This allowed the CPU to be clocked faster than the ISA bus, which was nominally fixed at 6MHz.

The CPU initially started in "real mode", the 8086 segmented memory model mode, and the BIOS only supported read mode operations. Once the BIOS had managed to load an operating system from disk it could then switch into "386 protected mode". The CPU was socketed rather than directly soldered onto the motherboard so it could be upgraded without a full motherboard replacement.

1. 32bit registers
2. SX 16bit external bus
3. DX 32bit external bus (132 pins)
4. Supported external cache
5. Enabled clockrate faster than AT bus (12MHz to 33MHZ)
6. Embedded paged MMU, enabled full process memory isolation and flat 32bit address space

3.6 IBM PS/2 (June 1986)

With the advent of the 80386 the CPU was outpacing the I/O. IBM decided to introduce an completely new I/O bus called Micro Channel Architecture (MCA). However it came with onerous licensing requirements and was never taken up by other manufacturers. The PS/2 line did however introduce a number of other improvements which were widely adopted, including the PS/2 keyboard (and later mouse), VGA graphics and 32bit wide memory modules (72 pin SIMMs).

1. 20 MHz 80386
2. 2MB memory
3. MCA bus
4. VGA graphics
5. PS/2 keyboard
6. 72-pin SIMM

3.7 ATA/IDE (1986)

After a proliferation of harddisk controllers and standards it became possible to integrate the controller directly onto the hard disk unit. This was called Intelligent Drive Electronics (IDE) and allowed for a much simpler interface with the operating system. To enable access to the drive a 40 wire ribbon cable effectively extended the AT bus directly to the drives (maximum of two) and became known as AT Attached (ATA) bus. The ATA interface was extremely simple and was quickly adapted. The emulation of the ST-506 interface by the IDE drives allowed them to work without major operating system changes.

1. Drive controller electronics moved from I/O card to disk
2. AT bus extended to drive over 40 wire ribbon cable
3. Two drives (master / slave) on single cable
4. Compatible with exsiting ST-506 interface

3.8 EISA (1988)

Rather than pay IBM licensing fees for MCA a group of nine manufacters (The Gang of Nine) extended the AT bus from 16bits to 32bits. While it made its way into growing "server" class systems and a number of highend graphics workstations it never reached mass adoption.

1. 32bit extension of the ISA bus
2. 8.33 MHz
3. Limited take up

3.9 Intel 80486 (1990)

The Intel 80486 was the first x86 CPU to support multiprocessing. With the integration of FPU (Floating Point Unit), previously an optional co-processor, the 80486 became an contender for heavy number crunching. Improvements in fabrication allowed for the intergation of SRAM onto the CPU die, creating a hierachy of memory, with L1 cache on chip, L2 cache on the motherboard and then DRAM main memory.

With the increasing size of L1 cache it became possible to run the CPU internally a multiple (initially x2 but later x3) the CPU bus speed and gain considerable performance improvement for very little system design changes (the CPU ran hotter but a clock doubled or tripled CPU could run on an existing 25MHz or 33MHz motherboard).

1. Support for multiprocessing instructions (atomic operations)
2. Intergated FPU
3. On chip cache
4. Initially 20/25 MHz
5. DX2 models had clock multiplier for 40/50/66 MHz internal clock
6. DX4 (actually 3x) 75/100 MHz released 1994

3.10 VLB (VESA Local Bus) (1992)

With the ISA bus becoming a critical bottleneck, especially with the advent of Super VGA, which required 23MB/s bandwith for a 30Hz refresh of a 1024x768x8 frame buffer. The VESA Local Bus adapters connected directly into the local CPU bus, so where 32bits wide and ran at 25 or 33 MHz. However it had to compete with memory access and was tied to the pinouts of the 386/486.

1. Slot attached directly to 486 bus
2. Mostly graphics but some SCSI / ATA cards available
3. Tied to processor
4. Competed with overclocked ISA bus

3.11 PCI (Peripheral Component Interconnect) (1992)

With the coming of the Pentium processors Intel launched a complete replacement for all the existing I/O buses. Unlike IBM and MCA Intel did not charge for royalties. In addition to being 32bits wide and clocked at 33MHz it had much richer configuration and addressing symantics, allowing for the creation of bus bridges.

The ability for the PCI bus to support a bridge to the ISA bus enabled motherboards to maintain backward compatibility with existing ISA cards and onboard Super I/O chips. Initially many motherboards came with only a couple of PCI slots.

Because of the way most system designs are drawn, with the CPU at the top, the local to PCI bus in the middle and the PCI/ISA bridge at the bottom the naming convention of Northbridge for local bus to PCI (and memory) and Southbridge for PCI to ISA bridge are often used.

The combined North and South bridges are often called the system chipset and in the case of Intel come as a specific pair with their own custom bus between them. Various functions will move between parts of the chipset.

1. Replaces ISA/EISA/MCA/VLB
2. Initially 32bit wide @ 33MHz
3. Supports autoconfiguration via Configuration Space
4. Message assing Interrupts (in later versions)

3.12 Pentium (1993)

The Pentium initially considerable teething troubles with bugs and high power draw from the 5V CPUs.

1. Original clocked at 60 MHz and 66 MHz
2. Updated by 3.3V with local APIC to enable multiprocessing
3. MMX instructions introduced in 1996

3.13 APIC (1994)

The Advanced Programmable Interrupt Controller (APIC) enabled Symmetric MultiProcessing for the x86. Prior to the APIC the distribution of the PIT interrupt could only go to a single processor. The APIC solved this by having an APIC per CPU, each with its own local interval timer. I/O interrupts are steering to any of the processors by programming the IO-APIC. Early multiprocessor systems used the MPS BIOS tables to tell the OS where to find the APIC controllers.

1. Augmented existing PIC
2. Enabled SMP
3. Requires one IO-APIC and one LAPIC per CPU
4. Each LAPIC has own timer

3.14 SMBus (1994)

The System Mangement Bus (SMBus) is a slow but low power and complixity bus for monitoring and controlling system components. With the advent of switched mode power supplies it became possible to run the SMBus and BMC (Baseboard Management Controller) without full system power.

1. Two wire bus based on I2C bus
2. Used for system management (PSU/temp/fans etc)

3.15 AGP (1996)

Improvements in graphics displays and the use of TrueColor (24bit colour) was causing the PCI bus to become saturated, limiting the frame rate. AGP solved this by isolating the graphics card to its own slot and making the bus single device only. AGP keep the PCI symantics but its simplified design allowed for much faster memory to memory transfers. A large proportion of the transfers were from main memory to graphics card offscreen texture memory for later on card copy to the frame buffer.

1. PCI interface modified for graphics
2. Point to point rather than bus

3.16 PIIX4 (1997)

Along with the 82443BX Northbridge this chip has become the Southbridge which is emulated by vmware. This chipset has builtin IO-APIC for multiprocessing support.

1. Supports USB
2. Supports ACPI for full hardware description

3.17 LPC (1998)

The demise of physical ISA slots left motherboards with a wide, slow bus it didn't need but the need for backward compatibility. A faster, narrow bus was developed by Intel called simply the Low Pin Count (LPC) bus. It functioned logically like the AT ISA bus but only had a total of 10 pins.

1. Replaces ISA bus with 10 pin bus @ 33MHz
2. Connections Southbridge to Super I/O and BIOS ROM

3.18 SATA (2003)

With PCI the ATA bus was bridged from the Southbridge. Over time two ATA buses were common (to allow for a second harddisk and a CD-ROM) and its speed was improved by increasing the clock rate of the bus and specialized DMA controllers. The limitations of wide parallel buses required the adoption of high bitrate serial point to point connections desirable so ATA was renamed Parallel ATA (PATA) and a Serial ATA (SATA) standard was developed. It also improved physically cabling by removing the need for a wide ribbon cable.

1. Replace (Parallel) ATA with 1.5Gbps serial

3.19 AMD64 (2003)

Intel initially created a new 64bit architecture called IA-64 and the Itanium processor but its cost and the lack of compatibility with existing Windows software limited its appeal. AMD introduced its own 64bit extensions to the existing x86 (aka IA-32) architecture. This was a relatively simple extension which allowed for existing 32-bit code to run in parallel with 64-bit code (provided the OS was 64-bit).

Intel effectively adopted it for their x86 processors as Intel64 and the two are close enough that amd64 and x86-64 are mostly symonomous.

1. 64bit extension of X86 ISA (Instruction Set Architecture)
2. Intel implemented as Intel64 in 2004.

3.20 HyperTransport (2003)

Another innovation from AMD was to move from a parallel local bus (aka Front Side Bus or FSB) to a bonded serial connection. Bonded serial connections are individually clocked serial “lanes” that are combined in a similar fashion to Ethernet bonding.

AMD also moved the memory controllers from the Northbridge directly onto the CPU. To increase memory bandwidth the server class CPUs had multiple memory controllers.

Within a generation most AMD chipsets had combined the Southbridge with the Northbridge. These still kept a LPC Super I/O chip for low speed functions.

1. AMD introduced HyperTransport (bonded serial bus) for CPU to Northbridge
2. Memory controllers moved to CPU

3.21 PCIe (2004)

PCI followed the trend of moving from a parallel bus to high speed point to point serial connections. PCI Express (PCIe) initially replaced AGP but later improvements allowed it to replace other system connections.

1. Bonded serialize PCI with 1/2/4/8/16 2.5 G/T lanes
2. Point to point star topology

3.22 Multicore (2005)

With heat killing the MHz race performance improvements had to come from more transisters. The shrinking of transister feature size enabled more logic to be put on a die. This culminated in two or more CPU cores being placed in a single CPU die. Most cores came with their own L1 cache and LAPIC, shared L2 SRAM cache, shared L3 eDRAM cache and shared memory and bus controllers.

1. First Xeon is Paxville with duel core

3.23 QPI (2009)

QuickPath Interconnect an Intel equivilent to HyperTransport and it used between both CPUs and the CPU to Northbridge. With QPI the memory controllers moved onto the CPU so memory access is now NUMA (Non Uniform Memory Access).

With the memory controllers removed the Intel chipsets moved to a single chip.

1. Replaces FSB on the Nehalem Xeon processes
2. Memory controllers moved to CPU
3. Integrated North and South bridges

3.24 System On a Chip (SOC)

1. AMD Epyc CPU
2. Move chipset functionality onto CPU die
3. CPU supports SATA/SAS/USB3.1/10GbE directly
4. Access to EEPROM via SPI
5. UP to 190 PCEe lanes
6. Connect to Super I/O BMC

Chapter 4

Storage

4.1 Block Storage

In block storage data is storaged as a collection of blocks (historically 512 bytes in size, recently 4096 bytes). On disks this could be addressed by using the disks geometry (cylinder, head and sector) or by logical block number. All storage is ultimately block storage of some sort, either disk or flash.

IBM PC and ST-506

The original IBM PC came with a Shugart (later Seagate) ST-506 hard disk. This consisted of a ISA controller connected to the drive by two ribbon cables. The command set involved controlling the movement of individual read/write heads. Is is often called the CHS specification (cylinder, head, sector), which identified the block (normally 512 bytes) on the disk by its physical geometry. Most drives where 5.25in in size.

3.5 inch drives

The first 3.5 inch drive with 10MB was released in 1983.

SCSI (Small Computer System Interface)

Standardized in 1986 it was initially a 8-bit parallel bus running at 5MHz. Original bus had a maximum of 7 devices (targets) but each target could have 255 Logical Units. SCSI was common for external hard disk as the bus could be up to 6 metres long.

IDE/ATA

By combining the controller with the physical disk allowed the interface to the system to be similified. Similar to SCSI disks could be accessed by LBA (Logical Block Address) rather than CHS. By removing disk geometry from the OS the drives could evolve without needing the OS to be constantly updated.

2.5 inch drives

In 1988 the first 2.5 inch drives appeared in laptops. This was 20MB in size.

7200rpm drives

In 1992 Seagate released a 2.1GB 7200 rpm drive.

SCSI-2

In 1994 the SCSI-2 standard is released. This halved the maximum length of the bus but doubled the number of targets to 16, bus width to 16 bis and clockrate to 10 MHz.

ATAPI (ATA Packet Interface)

This allowed for SCSI commands to be carried over the ATA device. As most CD-ROM drives at the time where SCSI based this allowed them to be put onto the ATA bus.

Over time most motherboards gained a second ATA bus (and connector), allowing for up to four ATA devices. CD-ROMs drives, which had often been connected via the sound card, could now be attached to one of the ATA buses.

USB attached storage

Around 2000 USB flash memory devices became available. Initially they very limited capacity compared to hard disks (first USB drives had 8MB). With the advent of USB2 and maximum rate of 480Mbps transfer rates improved dramatically.

RAID

The concept of using multiple drives to improve reliability had been around since the 1970s, initially with drive mirroring and later with parity. The terminology was standardized in the 1980s. By using SCSI Logical Units RAID sets could be sliced into multiply Logical Disks.

Fibre Channel

Fibre Channel is a transport layer which allows SCSI commands to be sent over optical fibre. First available around 1997 with a transfer rate of 1Gpbs early implementations were based on a Arbitrated Loop topology.

Serial ATA

The ATA interface had been extended up to 133 MBps but the problems of clock synchonisation on the parallel bus made increasing the clockspeed prohibitive. Advances in highspeed (> 1Gbps) SIDES (serial / deserial) logic enabled to move to serial rather parallel connections. The original SATA specification in 2003 was for point to point connection at 1.5Gbps, giving a maximum transfer rate of 150Mbps.

SATA conversion kits allowed for a smooth transition and subsequent updates have increased the speed from 1.5Gbps to 3Gbps, 6Gbps and 16Gbps. The latest specification is SATA Express, which uses PCIe directly to the device.

500GB disks

In 2005 the first 500GB hard disk was released.

Serial Attached SCSI (SAS)

Arriving in 2005, at the same time at 3Gbps SATA, SAS used the same physical connector but different more robest electrical signalling. This has allowed SATA drives to be connected to SAS controllers but not the opposite. SAS also supported bus extenders (aka bridges), allowing up to 16535 targets. SAS tracked the 6Gbps and is now available at 12Gbps.

FC Storage Area Networks (SAN)

FC switches, which make up SANs, are similar to Ethernet switches. They generally use same SFP module form factor but use different optics. FC emulates SCSI so unlike Ethernet cannot drop frames. To ensure this the switches have a much more complex buffer management system, based on buffer credits.

Topologically SANs differ from Ethernet in that they normally consist of two isolated networks. Each network consists of a number of switches, which form a fabric using a modified IS-IS Shortest Path First algorithm. This works because early SCSI disks supported dual connections, allowing for more than one path to the drive. The general extension to this is multipathing. SANs rely on multipathing within Operating Systems for fault tolerance.

Access control on a SAN is done by Zoning and Masking. Every node on SAN has a unique World Wide Number (WWN). Zoning restricts access based on either WWN or port. Only devices in the same zone can see each other but a device may be in multiple zones. Masking allows a device only to see certain LUNs. With the current Storage Controllers zoning and masking is less of an issue as they allow individual LUNs to be seen by certain WWNs.

Fabric switches will normally autoconfigure and maintain a consistent configuration between all members. Keeping fabrics isolated is therefore important as they will try to merge if connected together.

iSCSI

iSCSI is an alternative method of buiding a SAN based on TCP/IP. By using TCP it an guarentee delivery, a requirement of SCSI. Often seen as a low cost alternative to FC it allowed for multiple hosts to access a storage backend. Normally iSCSI traffic would be on an isolated VLAN with both the host and storage having multiple NICs and IP interfaces. By using existing multipathing techniques iSCSI could be highly resilient while using simple Ethernet switches.

FC over Ethernet (FCoE)

With the rise of 10Gbps Ethernet it was mooted to carry FC frames over Ethernet. To do this Ethernet switches needed to be aware they were carrying FC traffic and never drop the frame. To get the throughput required without sacrificing a compute core the FCoE functinality was baked into the NIC, often called a Converged Network Adapter (CNA).

FCoE networks generally need switches which also supported native FC to bridge between FCoE and FC, as most storage systems only support native FC. All FC switches need to have very low latency, normally under 1ms to switch a frame between interfaces.

Flash Storage (Solid State Disks)

Flash Storage is an evolution of EEPROMs, memory chips which would maintain their data without the need for power. Early flash devices were mostly memory cards or USB devices, which traded capacity for speed and the number of writes.

In SSDs this tradeoff is reversed, where speed and write capacity is traded against capacity. Because flash memory degrades with the number of writes all flash fails over time. To get around this flash based devices have more capacity than advertised and map out failed blocks.

SSDs generally emulate existing disks, either SATA or SAS. This allows them to replace hard disks without any changes to existing operating systems. However because flash memory working is very different from actual hard disks it is non optimal. A standard called Non Volatile Memory Express (NVMe) uses PCIe lanes directly to the flash device. This allows for considerable optimisation as the operating system knows it is accessing flash memory.

M.2

This is a physical interface on the motherboard which allows 4x PCIe lanes to a plug in device. It can also carry SATA and USB3 signals. Devices can either be SSDs (emulating a SATA device) or NVMe.

4.2 File Storage

Network Attached Storage devices take block storage and present it as file storage, normally over TCP/IP and Ethernet. They often use a clustered file system such as GPFS or OCFS2 underneath to provide resliency to node failure.

Network File System (NFS)

The defacto standard in unix network file systems, it was undergone a considerable number of revisions. Originally from Sun, it emulates POSIX file symantics closely. Originally based on Sun Remote Procedure Calls and a number of secondary protocols (such as statd, lockd and mountd) in NFSv4 these were all folded into a single protocol on a known TCP port.

Server Message Block (SMB)

Based on an extension to DOS SMB has gone on a number of revisions. The original protocol has largely been abandoned and was heavily revised with SMB2. The current version (SMB3) is an extension of SMB2 rather than an completely new version.

4.3 Object Storage