This paper examines the lower layers of the OSI model. It is meant to be an introductory level paper.
Section 1 – OSI Lower Layers and the TCP/IP Model

The International Organization for Standardization (IOS) is an internationally represented standards body founded in the late 1940’s. The IOS created the Open Systems Interconnection (OSI) model to describe the very complex task of moving computer data from something you understand on a monitor, to bits on a wire, and back to readable data on a remote system.

The OSI model describes the interdependency of networking protocols. This is a conceptual model only and simply and describes, not prescribe how networking can be broken up into understandable pieces. The lower layers are defined as layers 1-3 for the purpose of this paper.

**Layer 1, The Physical Layer** describes the hardware used to move data from one machine to another. This includes cables, connectors, pin functions, voltage and higher level functions such as bit level flow control and signaling (modulation/demodulation). Data in Layer 1 is *bit* level data.

**Layer 2, The Data Link Layer** describes two primary functions, or sublayers. The first lower sublayer which communicates directly with the physical layer is the Media Access Control (MAC) sublayer. The MAC is responsible for supplying orderly access to the network media. This includes sending and receiving data to/from the physical layer, detecting errors from the physical layer and using MAC addressing at an operational level.

The second sublayer, Logical Link Control (LLC) is responsible for MAC address control, IP to MAC address translations, and forming the Layer 2 frame by appending the upper level protocol packet with the Layer 2 frame header. Data in Layer 2 is *frame* level data.

**Layer 3** is known as The Network Layer. The main responsibility of the network layer is to route traffic from one network to another using network level addressing. In TCP/IP networking this is the job of IP. Routing traffic at level 3 includes maintaining and sharing logical routing tables and applying the rules of the appropriate routing algorithms. The network layer also enacts the QoS requested by the transport layer and fragments packets when they are too large to be transmitted by lower level protocols. Even though the network layer is responsible for upper layer requests, Layer 3 communications are connectionless and therefore, unreliable. Data at Layer 3 is *packet* level data.
Similar to the OSI model, the TCP/IP model describes the functions of network communications, but focuses on the lower layers. The TCP model does not address physical entities and so has no physical layer. In the TCP/IP model, the Network Access Layer, Internetworking layer and Transport Layers are analogous to OSI layers 2-4. The TCP/IP application covers all functions higher than transport layer ones. The TCP/IP model compared to the OSI model is shown on the right.

For the purpose of the example shown in this paper, I will assume the network consists of Cisco devices. Similar troubleshooting commands are available in other vendor’s equipment.

Section 2 – Lower Layer Protocols and Network Engineering

Why is it important to examine the lower portions of the OSI and TCP/IP models? As a Network Engineer you will spend most of your time working in the lower layer protocols. Great deals of network problems are solved at Layer 1. I have seen network problems as simple as the user moving a PC and not moving the network cable, a switch isolated when a technician plugged the up port into a user port of the same switches, and open twisted pair token ring segments causing a hub to constantly reconfigure leaving no resources for packet reception and distribution. These were all Layer 1 cabling issues. As the SolarWinds Head Geek, Josh Stephens says, “The first three steps in network troubleshooting are 1) Check the cable. 2) Check the cable and 3) Check the cable”.

Layer 2 also has a fair share of areas where things can go wrong. These may include expired entries in device’s ARP cache, which prevents packets from being mapped to the correct MAC addresses, or a system sending large numbers of malformed packets due to a problem in a NIC driver. When a network issue has been isolated down to level where you are certain the devices in question cannot be reached at all, the problem is most likely at Layer 1 or 2. A case like this might present itself in the following way: A remote gateway router and all systems behind it are showing as Down in the NMS, and you are not able to ping the devices from your workstation. At first glance we may assume the gateway router is down causing all the other devices to appear down, and as far as IP goes, they are.

But if we remember, IP is dependent on Layers 1 and 2 functioning correctly. The lack of IP reachability may only be a symptom of the issue, not the cause. To check if the device is power-out-dead or just not IP-dead we could telnet to a reachable, direct-neighbor router and examine the Cisco Discovery Protocol (CDP) and /or the Link Layer Discovery Protocol (LLDP) information. CDP and LLDP area layer 2 protocols that allow devices to track neighbor device connectivity at Layer 2. By issuing the `show CDP neighbors` command from the router that is a neighbor to the router in question, we’ll see one of two possibilities, the device is listed as a neighbor or the device is not listed. If the device is not listed then it is most probably unreachable at Layer 2. Seeing that the device cannot be reached by IP and the neighbor
router has no CDP/LLDP entry, it is probably safe to assume the router is dead or the connection to the remote site is down. From there it is a matter of remotely accessing the router if possible and/or contacting the carrier to check the WAN connection.

Below is a sample of a CDP/LLDP map from SolarWinds Engineer’s Toolset v10.6. The device being interrogated is Tex-3750. Once the device is seen in CDP or LLDP any available IP information is added.

What we have done here is started with a Layer 3 issue (the NMS can no longer ping the remote router) and moved the troubleshooting down to Layer 2 (CDP) then Layer 1 (examining the router and the circuit). This is a very common task. As you become familiar with a network, or a carrier, it becomes easier and quicker to get to the root cause of an outage, so some of these steps may be skipped to look for the usual suspects. It is when the usual suspects are not found to be the cause then careful, stepwise analysis of the issue using knowledge of the lower layer protocols will help you isolate the issue.

Section 3 – Lower Layer Communication Functions

Layer 3 routes data to a subnet but at that local segment level has no ability to determine how to deliver data to the end station. Layer 2 accepts Layer 3 packets, prepends the packets with Layer 2 MAC header and gives the resulting Layer 2 frame to Layer 1 for encoding and transmission onto the media. On the receiving side Layer 1 accepts properly formed packets and hands them up the stack to Layer 2. Layer 2 then strips off the Layer 2 header and hands the data up the stack to Layer 3. For each segment in an IP network these steps are repeated.

The main responsibility of Layer 2 is the addressing and control of local, non routed data. By non-routed, I do not mean data that has not been, or will not be routed in Layer 3, but rather local segment data movement that does not require routing. Layer 2 has no routing functionality. Layer 2 protocols include protocols that describe hardware and software functions such as Fiber Distribute Data Interface (FDDI) and Ethernet as well as protocols that are software only, such as Link Layer Discovery Protocol (LLDP) and Point-to-Point Protocol (PPP). The software-only protocols are hardware agnostic, requiring no specific hardware.
The below diagram depicts a transfer of data from a remote endpoint (not shown), to the endpoint at 10.1.11.152. Items such as MAC addressing have been simplified for clarity.

Here is how this data is transferred step-by-step:

1. R1 receives the packet at Layer 1 Ethernet as electrical or optical signals. Layer 1 converts the signals to bits, strips off signaling bits, and hands the remaining bits to Layer 2. Layer 2 Ethernet forms a packet from the bits and determines if R1 is the destination by inspecting the destination MAC address in the packet. If R1 is not the destination then the packet is dropped. If R1 is the destination the Cyclic Redundancy Check (CRC) is performed to detect packet formation errors. Layer 2 can do nothing further with the packet and so strips off the Layer 2 header and hands the packet up to Layer 3 IP.

2. IP on R1 reads the IP header and finds that the packet has a non-local IP address. Referring to the routing table on R1, IP finds that the 10.1.11.0 network is reachable via the next hop, 10.1.10.2 on interface 1. IP then hands the packet down to Layer 2 for addressing and delivery over segment 1. Layer 2 refers to the Address Resolution Protocol (ARP) table and finds that 10.1.10.2 is found at MAC address 2 on segment 1. Layer 2 addresses the packet for delivery to R2 by prepending the IP packet with the Layer 2 header and handing the packet to Layer 1 for encoding and delivery. Layer 1 sends the data onto segment 1 as electrical signals.

3. R2 receives the electrical signals and repeats the process seen in step 1 in order to get the packet up to Layer 3. IP on R2 finds in the routing table that 10.1.11.0 is locally attached via the Ethernet interface. R2 sends the packet down the stack for transmission as seen in step 2, sending the packet to MAC address 7 of segment 2.

4. Sw1 receives the packet, and then refers to the switch lookup table for the port that MAC address 12 is connected to. The lookup table indicates that MAC address 12 is on port 5 so the switch transmits the packet out port 5. The packet sent out port 5 is almost the same as the one received by the switch. The only difference being that a new CRC has been calculated so the CRC field is updated.

5. The end station receives the packet and passes it up the stack to the original OSI layer. The packet may never go any further than IP as is the case if it was an ICMP echo (Ping). If the end station were negotiating a window size the packet would only be passed up to TCP at Layer 4. Higher level functions such as session control or data format messages would be handled by Layers 5 and 6 respectively. If the data was part of an application, the data goes all the way up to Layer 7.

This movement of information from Layer 1 to 2 to 3 and back down again is repeated at each hop an IP packet transverses. Simplified, these steps are:

1. Receive signals at Layer 1.
2. Determine if MAC address is for this station at Layer 2.
3. Send up to Layer 3 IP for determination of next hop IP (if any) and output interface.
4. Map next hop IP to MAC using ARP table at Layer 2.
5. Address for delivery on local segment at Layer 2.
6. Transmit on Layer 1.

All of this happens in a matter of milliseconds or even microseconds on very fast systems. The signals then travel at the speed of light for optical media or radio, or at the speed of electrical transmission for electric signals.

**Encoding - Placing the Signal on the Wire**

In several places I refer to signals being transmitted at Layer 1. It is worth understanding how this happens to have a more accurate understanding of network data exchange. Remember at the physical layer data units are bits, ones, and zeros. But ones and zeros are only conceptual things - they must be represented by something in the physical world to convey information. This is what layer one encoding does. Encoding protocols have two primary functions:

1. Provide a repeatable method of transmitting and receiving signals that can be interpreted as one of two states, one or zero.
2. Provide a method of synchronizing clocks between transmitter and receiver so that signals may be properly anticipated and interpreted.

For these examples I will use only electrical (wire media) encoding. Encoding protocols have to agree on both ends of a wire what a 1 will look like and what a 0 will look like. Perhaps the simplest method to accomplish this is to set 0 volts = 0 and +5 volts = 1. Here is the resulting voltage pattern for a series of alternating ones and zeros. The bit boundary has been added to show where bit signals begin and end.

![Bit Boundary Diagram](image)

There are a couple of good points to this encoding method. It is very simple and so easy to read and easy to transmit, and It provides a non ambiguous method of representing the two levels. There are also a couple of issues this method has. Consider the line voltage pattern below.

![Bit Boundary Diagram](image)

Bit Value: 0 0 0 0 0 0 0 0 0 0 0 0
Is this representing the transmitter sending a string of zeros or is it the transmitter not sending anything? Perhaps the transmitter is shut down. Will the receiver continue to count zeros received forever? This is an obvious issue. Several encoding algorithms have incorporated a *non-return to zero* (NRZ) rule. This means that zero volts cannot be used to represent any data. Instead, data could be represented by +5V and -5V, creating a polar NRZ encoding. This solves one issue for the above encoding but there is another problem. If we simply reassign a 0 bit as -5V, the shape of a long string of 0s or 1s will both look like the straight line voltage at a single level seen above. One advantage of this encoding is that it is easy to visualize binary data as encoded data so it is often used to as a visual guide when examining more complex encoding. We will call this encoding simple encoding and use it to help understand other encoding methods. From here forward we will not refer to absolute voltage levels with the understanding that zero voltage will not be used.

This gives no solution for the second requirement of encoding; providing a method of synchronizing clocks between sender and receiver. If the above encoding is a station sending 500 zeros in a row, then the receiving device may not be ready when a few 1 bits come across. This happens when a receiver has lost its clocking and so cannot determine where bits end or begin. A string of 2000 zeros may be interpreted as a string of 2011 zeros if the receiver is not properly marking bit boundaries. Synchronization is a method used to minimize errors in reading encoded data. By making sure both sides on a connection agree on what a signal transition looks like and when to expect them the devices as said to be clock synchronized. While I will not go into all the different encoding methods, (there are quite a few), I will use a couple of common ones as examples and show how they provide good signaling and strong synchronization.

**Manchester Encoding**, the encoding method for 10 Mbps Ethernet follows these rules:

- Bit code voltage transitions occur at mid bit.
- A zero bit is represented a low to high mid bit voltage transition, a one bit is represented a high to low mid bit voltage transition.
- Voltage transitions which occur at the beginning of a bit boundary serve to return the voltage to the start point for the next bit. These may be used for additional clocking.

Here is a string of bits and the associated Manchester encoding.

![Manchester Encoding Diagram]

Note that the longest that the signal can remain at one voltage level is 1 bit in length and that every bit has a change in voltage mid-bit, even for strings of 1s or 0s. This provides strong clocking and non-ambiguous bit representation. In this example we have overcome the server limitation of long strings at a single voltage as seen in simple encoding. One problem remains for Manchester Encoding. It is reliant on the phase of the signal. If a wiring technician were to install cable with the leads swapped, the encoding would flip the encoding so zeros would be received as ones and vice versa.
A solution for the problem came in the form of Differential Manchester. In this encoding a one is marked by no transition at a bit boundary and a zero is marked as a transition at the bit boundary. There is also a mandatory transition at mid bit for clocking only. Because the direction of a voltage transition does not matter, only the presents or absence of a transition, either polarity will work. The required transition at mid bit provides strong clocking. Given the bit pattern below, can you complete the Differential Manchester encoding?

<table>
<thead>
<tr>
<th>Bit Value</th>
<th>Differential Manchester Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 1 0 1 1</td>
<td>Clocking</td>
</tr>
<tr>
<td>0 0 1 0 0 1 1 1 0</td>
<td>Simple Encoding</td>
</tr>
</tbody>
</table>

These encoding examples are very efficient for sending data streams but leave little room for Layer 1 signaling. Signaling is used to communicate information between stations that is not part of a data packet. Signaling is limited to a small number of bit patterns such as the Ethernet preamble, to be examined further later. The preamble functions to align clocks using a regular pattern of bits. This type of signaling requires reserving bit patterns and as so is self limiting. Sooner or later a reserved bit pattern would be duplicated as data, causing a protocol error.

One solution for advanced encoding and signaling is 8b/10b. This encoding method adds a couple of steps before encoding bits onto a media; it translates each 8 bits to be transmitted into a 10 bit symbol. This is accomplished by mapping the most significant five bits of the octet to a set six bit code. The remaining three bits are assigned to a set four bit code. For example the octet 00000010 is split is to two sections, 00000 and 010. The first five bits are mapped to 10011 according to the 8b/10b translation protocol. The remaining bits are mapped to 0101, again according to the 8b/10b translation. This 10 bit sequence is then reverse translated at the receiving station, yielding the original eight bits. The goal of these assignments is to:

1. Limit the possibility for runs of consecutive bits being transmitted. 8b/10 b never has more than five consecutive like bits in sequence.
2. Provide for bit parity. 8b/10b. The rules used in 8b/10b ensure that 50 percent of the bits transmitted will be one, and 50 percent will be zero.
3. In any five consecutive bits the maximum difference between one bit frequency and zero bit frequency is two.
4. By adding two bits 1024 codes are left as unused due to undesirable bit patterns (such as 11111). Some codes with acceptable bit patterns are use for out of band signaling.
8b/10b encoding is used in Gigabit Ethernet, SAS, SATA, Fiber channel adapters HDMI, digital audio tape, and USB3. 10Gigabit Ethernet uses a similar method, 64b/66b encoding. A variety of encoding methods can then be used to place 8b/10b data onto a wire.

**Section 4 - A focus on Layer 2 and Ethernet**

A great deal of changes took place in the Layer 2 MAC sublayer during the 1980’s and 1990’s. Protocols such as Token Ring and ARCNET and FDDI had large market shares. This was partially due to some of the limitations of Ethernet at that time. Ethernet in its earliest form used a very thick cable called 10-BASE-5 to for multiple access of data up to 500 meters per segment. Each station on the network segment had to be connected by driving a large network tap into the cable to make connections with the positive copper core and the negative wrapped shielding material. It looked like a cable TV cable that was as big around as a golf ball. The cable was heavy and stiff, making it difficult to work with, so much so that it got the nick-name of “frozen garden hose”. The tap and tap cable which connected the PC to the tap were also no fun. A thinner version of the cable called 10-BASE-2 was introduced. While this cable was much easier to deal with, it limited the segments to no more that 185 meters. The introduction of hubs and repeaters helped alleviate some of the problems Ethernet faced but to understand the largest problem we need to look into the packet format and protocol operations.

**Ethernet Addressing and Packet Format**

The Ethernet II packet header is used to indicate the MAC source and destination addresses as well as information about what the packet contains. Local segment Ethernet traffic uses Ethernet MAC source and destination addresses six bytes in length and expresses in hex format. Each station has a MAC Ethernet address which is usually hard coded into the adapter. Some adapters allow for manual setting the MAC address but this is not normally necessary. Ethernet reserves two sets of special addresses classes, a broadcast address of FF.FF.FF.FF.FF.FF, and multcasts addresses. Stations on a shared Ethernet network take in all packets, examine the destination address. They discard any packets not directly addressed to that station, broadcast packets, or applicable multicast packets. Cisco Discovery Protocol has a reserved Ethernet multicast address. Other protocols used to control the flow of traffic at Layer 2, such as Spanning Tree Protocol (STP) also have assigned multicast addresses. The standard form for an Ethernet packet is shown below.

<table>
<thead>
<tr>
<th>Preamble</th>
<th>Frame Delimiter</th>
<th>Destination MAC</th>
<th>Source MAC</th>
<th>802.1Q Header</th>
<th>EtherType</th>
<th>Payload data</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 octets of 10101010</td>
<td>10101011</td>
<td>6 octets</td>
<td>6 octets</td>
<td>Optional 4 octets</td>
<td>2 octets</td>
<td>Up to 1500 octets</td>
<td>4 octets</td>
</tr>
</tbody>
</table>

**CSMA/CD Operations**

Early Ethernet used a Carrier Sense, Multiple Access/Collision Detection (CSMA/CD) protocol. The name describes how Ethernet access the media. Multiple access Ethernet allowed a number of stations access to transmit and receive data over a common cable. To accomplish this some rule were put into place regarding how to access the media. Here is how a multiple access Ethernet network interface would operate.

- Once a station has a packed formed and ready for transmission, the network adapter listens to the network to determine if another station is already transmitting. This is called **carrier sensing**.
If a transmission is detected, the station waits until the transmission is completed and then waits a small additional period of time called the interframe gap. This is added to account for a station that may be transmitting multiple frames, allowing the transmitting station to complete the frame sequence.

If no traffic is detected the adapter begins transmitting a preamble consisting of seven octets of alternating 1 and 0 bits (10101010), starting with 1. The preamble is followed by a single octet called the start of frame delimiter. The delimiter is similar to the preamble in that it starts with the 1 and 0 pattern but ends with two 1 bits (10101011). The preamble sets up strong clocking using a repeating and regular series and the two 1 bits in the delimiter mark the end of the preamble and start of the data frame.

As the adapter is transmitting it is also listening for another station transmitting at the same time, this is called a collision. If a collision is detected the station transmits a jam signal. All stations detecting the jam signal stop transmitting and increment their retransmit counter.

All stations that were transmitting during the collision set a random time period and hold off retransmitting for that period.

Once the wait period is complete the stations attempt to transmit again. If a station’s retransmit counter reaches the maximum value the adapter dumps the packet from memory.

While this seems a bit messy, it worked well for a small number of stations sharing a segment. As the number of stations increased and the traffic from each station increased, Ethernet would lose effective bandwidth and suffer from excessive collisions and retransmissions. A shared segment where all of the stations have the possibility to cause collisions is called a collision domain. In early Ethernet implementations of the only way to control the collision domain was to limit the number of stations on each segment. With all stations having to examine all packets, respond to jamming signals and wait for the ability to transmit shared segment, Ethernet would lose efficiency quickly under heavy load. This is where ACRNET and Token Ring had an advantage. These two protocols used the concept of a shared token, a defined set of bits that was passed around from station to station. The exact ruled differed depending on the implementation but in general, when a station had control of the token that station could transmit onto the media. The receiving station would strip the packet off of the token and add a confirmation message to the token. The original transmitter would strip off the confirmation and if it had no further data to send, release the token to the next station in line, thus a ring. Token passing protocols have rules as to how long a single station can control the token. Because of this, there is a maximum time any station had to wait to transmit. This time can be defined as the number of stations multiplied by the maximum token retention time and added to the ring latency. This is an advantage token passing protocols had over Ethernet. With Ethernet you eventually would gain access to the media but access is not assured. Ethernet is considered to be probabilistic; whereas token passing is deterministic.
Ethernet Enhancements – Winning the Layer 2 Wars

In the early 1990’s, with standard Ethernet transmitting at 10 Mbps at best and converging to 0 Mbps in a worst case collision storm and Token Ring and ARCNET allowing for deterministic access at 16 and 20 Mbps respectively, Ethernet had to evolve or be replaced. The first of the Ethernet enhancements, Repeaters, actually made the problems with contention worse but it did extend the overall length of a segment. Repeaters would be placed on the end of an Ethernet cable in the place of a terminating resistor and a second Ethernet cable would be attached to the other side of the repeater. The repeater would receive electrical signals from one port and repeat them out to the segment on the other side. Repeaters would send jam signals out all ports when a collision was detected on any port. Multiport repeaters (hubs) acted the same way. This lengthened the network and increased the collision domain to include all stations connected to any segment on the repeater. Repeaters and hubs were not much of a solution as they only operated at Layer 1 and had no way of reading or understanding Layer 2 information, most importantly, Layer 2 addressing.

The Ethernet bridge was created to help control the problem of growing collision domains. Bridges connect network segments together but add functionality by operating at Layer 2. Bridges read all packets coming into each port and use a bridge table to determine which port the destination station is on. Bridges have the ability to decode bits to packets and read MAC addresses. If the destination is on the same segment where the bridge received the packet, the bridge simply assumes the destination received the packet and does nothing. If the destination is on a different segment the bridge sends that packet to the segment with the receiver only. When a collision occurs on a segment the bridge participates in the jamming and back-off steps in the segment experiencing collisions only. The other segments attached to the bridge are insulated from the collision. Early bridges had to be manually configured with the MAC address of each station and a mapping of which port they were attached to. These bridges were replaced with learning bridges. These are bridges that listen to packets and build a bridge table from the source addresses of received packets. Considering that hundreds of end stations could be attached to a multiport 10BASE2 bridge, this was a real time saver. Learning bridges would discard any packet destined for an address they have not yet learned. This means that stations could not receive any packets until they had transmitted at least one. Bridges control the collision domain but forward all broadcast packets out all ports and so do not alter the broadcast domain. Broadcast domains are defined by Layer 3 edges, router connections.

Early bridges had one limitation that limited market adoption. Compared with hubs bridges they added significant delay as they had to read each packet, look up the address in the bridge table and then properly reform and forward the packet including recalculating a new CRC. Processing speed was slow compared to today’s standards and the bridge had to read the whole packet before performing a lookup. Ethernet switching helped solve this problem. Switches were introduced with a feature to speed packet forwarding called cut through switching. Switches still used a bridge table to determine which address are on which ports, but they added the ability to make the forwarding decision immediately after reading the destination address in the received packet. Cut through switches could begin retransmitting packets onto the destination segment before the whole packet was received. This eliminated the need to store packets, check the CRC to validate the packet, and the need to recalculate a new CRC before retransmitting. Once a destination address was determined the cut through switch simple retransmitted the bit as-received. The price for this increase in speed was a decline in reliability. Malformed packets and jabber, the result of a malfunctioning transmitter, would be retransmitted as long as the problem happened after the destination address was read.
Another feature of switches that helped Ethernet is the ability to allow full duplex conversations between an end device and a switch by limiting each switched segment to two devices. In multiple access Ethernet, devices have to listen for collisions while transmitting, using an electrical shunt and two of the possible connections for passing data. If there are only two devices connected to a four wire twisted pair connection, then the connection can be set up as full duplex, with a dedicated pairs for transmitting and receiving at both ends on the segment. These dedicated pairs eliminate the possibility of collisions, eliminating collision domains from the whole network in fully switched Layer 2 environments. While all these improvements were being adopted, Ethernet bandwidth was rapidly increasing from the original 10 Mbps to 100 Mbps and beyond. There was still one huge problem holding Ethernet back from completely winning the Layer 2 wars. With an explosion of devices and speeds in a fully switched network, the problem of broadcast storms became apparent. Because Layer 2 broadcasts can be initiated by Layer 2 or any higher layer protocol requiring network wide information, the need to control the broadcast domain became critical.

**Overview of VLANs – Controlling the Broadcast Domain**

A Virtual Local Area Network (VLAN) is a method of abstracting the physical media segment(s) from network numbers. Before VLANs the only way to break up broadcast domains was to add routed, physical segments. Below is a common depiction of a hub/router network compared to a VLAN network.
The addition of VLAN capable switches allows the broadcast domains to be broken up in a manner independent of the cabling. VLANs can transverse segments and even geographical locations. VLAN switches exchange changes to VLAN using 802.1q VLAN trunking protocol. This allows changes on a single switch to be properly implemented on some or all VLAN switches throughout a network. These changes are controlled by assigning switches to VLAN control domains. Changes to a switch in a single domain are limited to the switches belonging to that domain. Before 802.1q VLANS had to be manually configured on each switch on a network. Several proprietary VLAN control protocols exist but the standard 802.1q has become the industry standard.

VLANs are implemented by assigning one or more virtual network number to a port on a switch or router. Devices on the physical segments are then assigned to the virtual network. In the LAN Network shown above, when a device broadcasts onto LAN 1, all devices on that hub experience the broadcast. In the VLAN Network only stations belonging to a particular VLAN will experience broadcasts on that VLAN. This has solved the last remaining problem for adoption of Ethernet over other access types. Today, FDDI and Token Ring are obsolete and ARCNET has a limited implementation base among automation systems such a Programmable Logic Controllers (PLCs). Ethernet in its various forms has replaced them all.

**Switch Technology Today**

The main advantage of a switch over a router is speed. Switches are able to make very fast decisions on traffic forwarding and move traffic to the next segment in the low single millisecond range and faster. Originally switches obtained this advantage by using cut through forwarding. With the increased speed of Application Specific Integrated Circuits (ASICs), store and forward switches now rival the speed of cut through switches. Store and forward switches receive the entire packet, calculate the CRC and retransmit a new packet in almost the same time cut through switches do and store and forward switches are less prone to forwarding on bad frames. Improvements to error checking have been implemented in cut-through switches helping the switch better detect common problems in frames without having to calculate a CRC. With the increased speed of store and forward switches these are not highly adopted.

Recently, High Performance Computing (HPC) has emerged with applications requiring latencies in the single microseconds. Cisco has introduced the Nexus 5000 with extremely low latency cut-through switching to meet this need. Cisco claims the forwarding latency in these switches is 2 to 3 microseconds.

**Section 5 - Layered Network Design and Redundancy**

High availability, highly redundant switched networks are ubiquitous today. In general these networks follow the below design.
The core level provides a dedicated, very high speed switch environment for the rapid movement of data. The distribution level provides redundant distribution of data between distribution layer services such as servers, Internet connections, WAN connections and access level switches. The distribution level also involves routing functions where required. In reality, VLANs have a one-to-one association with IP subnets so routing is required to interconnect these VLANs. This is accomplished by embedded routers in switches or Layer 3 switching. Access level switches connect end-user devices to the network. Recent designs do not differentiate between core and distribution levels, using a single, collapsed core/distribution level. Data loops are created by the high number of redundant connections at the access and distribution layers. If these loops were left unchecked the result would be switched traffic looping throughout each possible circular path. Loops also cause problems with forwarding decisions - if loops exist, all stations involved in the loop appear to be on both sides of a switch loop. While looping traffic would eventually age out, the sheer volume of it would kill the entire network. This is the job of Spanning Tree Protocol (STP).

STP creates a tree that defines the position of each switch in a network. Spanning tree is a self configuring protocol where the switches in a network share information about their neighbors. Here is how it operates:

1. One switch is determined to be the “root” switch and the shortest path to the root switch is calculated for each switch.
2. The switch closest to the root is selected and all data to the root passes through this switch.
3. Each switch calculates its fastest path to the root and blocks other ports with paths to the root.

As switch ports are added to the network they need to participate in the tree. Here is a simplified explanation of what a switch does to have a new port enter the tree, or go into blocked state.

1. The port initializes and loads communication protocols.
2. The port listens for the root to place it into blocking state.
3. If the port receives no block command then it begins learning the tree but blocks any forwarding until it part of the tree.
4. The port begins forwarding as the tree allows or enters blocking state if required to avoid a loop.

**Section 6 – Your Crystal Ball - The Packet Capture.**

While there are many tools available to the network engineer, I always make sure I have three things available:

1. Access to a Network Management System such as SolarWinds Network Performance Monitor.
2. SolarWinds Engineer’s Toolset
3. A packet capture program, either Wireshark or Microsoft Network Monitor.

The NMS and Engineer’s Toolset are very helpful in narrowing down but once I have a problem narrowed down to a specific piece of equipment, I almost always finish the job using a packet capture. Packet captures operate at Layer 2 and have the ability to gather information on upper level protocols by capturing entire packets at Layer 2. Once these packets are captured, they have to be decoded using a series of parsers. There parsers determine the protocol in use at each layer and decode that protocol into human readable format. Here is an example.
Sample Packet Capture

There is a lot of information here. This is what we are seeing, layer to layer.

The very top section is the summary of the capture showing a particular packet we are expanding on, packet number 280, an STP packet, and several other packets.

The middle section has the human readable decoding of the Layer 2 packet bytes. These bytes are seen in the lower section. As the capture operates at Layer 2, it has no possibility of capturing or understanding Layer 1 issues. All we really can conclude about Layer 1 is that it is passing data as we are able to capture it at Layer 2.

This packet is communication information about STP to all devices using STP (notice the STP multicast destination address). The transmitting station is passing information on the root bridge address and its STP state, learning and forwarding.

I have clicked on the Port Role section of the decode which highlights the raw Layer 2 data below that indicates port role. Without a parser in the software, all you would see is the raw Layer 2, hexadecimal bytes. Good luck reading that!

Normally when you use a packet capture you will already have a suspect in mind, a station failing to transmit a certain protocol such as NetFlow or SNMP, a protocol issue such as STP loops. To properly capture the issue keep these guidelines in mind:

1. Switched networks channel traffic down specific paths. You must plan where to capture packets according to the issue at hand. Spanning ports or capturing packets from application servers involved in the issue are typical methods.
2. Understand normal vs. abnormal capture results. You should be using the packet capture to prove or disprove your theory of the particular issue. Capturing packets without a theory of what may be wrong will bury you in data. While the issue may be captured, not knowing where or what to look for leaves the needle in the haystack issue.

Notice that this capture also shows ARP in action in packet 279, SNMP requests in packets 282 and 283, and TCP packet reassembly in many packets. This is only a few packets from a 30-second capture of 2345 total packets. You had better have some idea of what you want to find before you look.

Section 7 – Review

OSI and TCP/IP models describe how network communications operate at various levels. Most network engineering takes place in the lower layers, OSI layers 1-4. Many issues are solved at Layer 1 with problems as simple as an unplugged cable or a device unplugged.

Working up through the layers from the Layer 1 (cables) to Layer 2 protocols to Layer 3 services can help isolate an issue. Being familiar with a network is very helpful in troubleshooting. If you understand how the network operates, you can quickly focus on problem areas.

When packets transverse multiple hops they are passed down the protocol stack from the initiating layer, to the next lower all the way to bits in the media at the physical layer. Once the bits are received at the destination they are translated back up the layers to the layer providing the required service, such as IP providing the next hop information. This process is repeated as many times as required to pass information in an orderly fashion. For the case of an IP packet moving across multiple hops here are the steps:

1. Receive signals at Layer 1.
2. Determine if MAC address is for this station at Layer 2.
3. Send up to Layer 3 IP for determination of next hop IP (if any) and output interface.
4. Map next hop IP to MAC using ARP table at Layer 2.
5. Address for delivery on local segment at Layer 2.
6. Transmit on Layer 1.

Bits are transmitted on physical media as signals. These signals are created using encoding methods such a Manchester, differential Manchester, 8b/10b and many others.

Ethernet had a good amount of problems in its early stages but improvements in speed, the emergence of switches and VLANs as well in lower costs led to Ethernet being the defacto standard for LAN technology, eliminating previous LANs, such as Token Ring, FDDI and others.

High levels of redundancy in Ethernet switched networks may cause looping traffic. Spanning Tree Protocol is used to block loop paths.

Solving network issues requires the proper tool and a good understanding of the network and protocols in operation. A packet capture tool is a must for troubleshooting network issues. Packet captures must be placed properly to capture the issue at hand. Improper placement will not discover the issue.
Answer to Differential Manchester Encoding Exercise

Correct Answer

Incorrect Answer
Related SolarWinds Products

SolarWinds Orion Network Performance Monitor
Orion NPM enables you to quickly detect, diagnose, and resolve network outages and performance issues. It offers network-centric views that are designed to deliver the critical information you need most. And Orion NPM is the easiest product of its kind to use and maintain, meaning you will spend more time managing networks, not supporting Orion NPM. NPM:

- Monitors and analyzes real-time, in-depth, network performance statistics for routers, switches, wireless access points, servers, and any other SNMP-enabled devices
- Simplifies network issue investigation with drill down maps and Top 10 views of your global network
- Gets you up and running in less than an hour with do-it-yourself deployment
- Scales to accommodate growth and management needs with a hot standby engine, multiple polling engines, and additional web servers
- Enables advanced alerting for correlated events, sustained conditions, and complex combinations of device states
- Monitors the energy consumption of Cisco® EnergyWise-enabled network devices and displays policies that regulate energy consumption
- Extends management capabilities to NetFlow traffic analysis, IP SLA monitoring, IP address management, network configuration management, and application and server performance

http://www.solarwinds.com/products/orion

SolarWinds Engineer’s Toolset
Once you’ve experienced Engineer’s Toolset, you’ll never troubleshoot your network the same way again. It includes a collection of powerful network management tools, all of which can be easily accessed through an innovative Workspace Studio to quickly resolve issues right from your desktop.

With the Workspace Studio, you can organize your tools to tackle troubleshooting challenges and then recall that workspace with a click when you need it – saving valuable set-up time. And shared credential management saves even more time as you switch easily between the tools you need without re-entering credentials.

Engineer’s Toolset

- Cut troubleshooting time in half using the Workspace Studio, which puts the tools you need for common situations at your fingertips.
- Monitor and alert in real time on network availability and health with tools including Real-Time Interface Monitor, SNMP Real-Time Graph, and Advanced CPU Load.
- Perform robust network diagnostics for troubleshooting and quickly resolving complex network issues with tools such as Ping Sweep, DNS Analyzer, and Trace Route.
- Deploy an array of network discovery tools including Port Scanner, Switch Port Mapper, and Advanced Subnet Calculator.
- Manage Cisco® devices with specialized tools including Real-time NetFlow Analyzer, Config Downloader, and Config Compare.

http://www.solarwinds.com/products/toolsets/
About SolarWinds

SolarWinds is rewriting the rules for how companies manage their networks. Guided by a global community of network engineers, SolarWinds develops simple and powerful software for managing networks, small or large. Our company culture is defined by passion for innovation and a philosophy that network management can be simplified for every environment.

SolarWinds products are used by more than one million network engineers to manage IT environments ranging from ten to tens of thousands of network devices. Comprised of fault and performance management products, configuration and compliance products, and tools for engineers, the SolarWinds product family is trusted by organizations around the globe to design, build, maintain, and troubleshoot complex network environments.

SolarWinds is headquartered in Austin, Texas, with sales and product development offices around the world. Join our online community of experts at thwack.com!

About the Author

Andy McBride is a Technical Specialist for SolarWinds focusing on making knowledge of networking and network management accessible to customers and prospects of all levels. The “New to Networking” series is specifically written for an audience with limited prior exposure to these technologies. His technical background includes seven years at International Network Services (INS) as a Network Engineer and Managing Consultant, three years as a Novell Certified Instructor and five years as a Network Performance Products Manager with BT-Infonet. Prior to entering technology, Andy worked in aerospace on projects such as the SR-71, F-117, F-22, L-1011, F-18, and the space shuttle main engine and has a degree in Chemistry from California State University Northridge. Go Matadors!