Ethernet and Internet Protocol (IP) have been used in commercial networks for decades. In recent years these technologies have made their way into telecommunications and military networks as well. Ethernet, the most popular communication protocol, does have shortcomings, however; network architects have proven willing to adapt to or work around its limitations.
**Introduction**

Ethernet and Internet Protocol (IP) have been used in commercial networks for decades. In recent years these technologies have made their way into telecommunications and military networks as well. Ethernet, the most popular communication protocol, does have shortcomings; however, network architects have proven willing to adapt to or work around its limitations.

This paper is intended to introduce you to:
- Basics of data transmission using Ethernet and IP
- Concepts of network layers
- Layer-2 switching
- Layer-3 switching
- Protocols that typically run on Ethernet and IP switches
- Redundancy and failover – a critical subject for both military and telecommunication applications
- Applicable examples

**OSI Network Layers**

Communication over Ethernet and IP is based on the concept of network layers as delineated by the Open Systems Interconnection (OSI) Reference Model. Each OSI layer is independent of the others but has well defined interfaces with the layer above and the layer below. Ethernet is Layer-2 and IP is Layer-3. This segregation of communications is one of the primary reasons Ethernet and IP have been so widely adopted.

Segregation into layers allows each layer to be modified independently; the other layers are not affected. It also allows mixing and matching of layers to implement specific data transmission and communication. For example, Ethernet (Layer-2) can be used over various physical interfaces (Layer-1) such as fiber or copper.

Networking layers are defined by the ITU-T Specification as part of seven-layer OSI Reference Model. Figure 1 depicts the OSI layers.

Each communication layer has a very specific task. Layer-1, for example, is responsible for establishing a link and transferring bits over the medium (copper or fiber). It does not understand the meaning of those bits but does understand where the packet starts and where it ends. In a nutshell, Layer-1 gets a bucket of bits from Layer-2 that it must transfer to the device on the other side of the link.

Layer-2 is responsible for transferring data packets to a specific address and ensuring that it is not corrupted during transmission. Ethernet uses Medium Access Controller (MAC) addressing to direct a packet to the intended receiver. Cyclic Redundancy Checking (CRC) is used to check if the Ethernet packet is intact. It is important to note that each Ethernet port has a unique MAC address. In case of an error, Ethernet can optionally notify the upper layer of the occurrence and will drop the corrupted packet. Ethernet does not perform packet retransmission. This “Best Effort” delivery concept means that while Ethernet will do its best to transmit the packet intact, it does not guarantee that the packet will reach its intended recipient.

Layer-3, the Network layer, delivers packets to a unique global address. IP, the most prevalent Layer-3 protocol, takes a packet from the upper layer and delivers it to the specified IP address. Although Layer-2 and Layer-3 appear to essentially perform the same function they are decidedly different. Layer-2 addressing is used for a low-level local connectivity. Layer-3 provides a level of abstraction. Consider, for instance, a vehicle that has several devices interconnected using Ethernet. Because every vehicle uses a unique set of addresses, an application written to send packets using only MAC addresses would need to be reconfigured for each vehicle. Plus, every time a device is replaced, the application will need to be reconfigured for a new MAC address. IP provides abstraction for Ethernet. Once an IP address is defined, packets can be sent to without regard for the Layer-2 protocol used or the destination’s Ethernet MAC address.

Layer-4 is the Transport layer. TCP and UDP, the two most frequently used Layer-4 transport protocols, guarantee the application data delivery. An application gives TCP or UDP a chunk of data, the destination IP address and port number; the transport protocol does the rest. TCP breaks the data into smaller chunks that fit into individual packets, insures that all packets reach their destination and, in case of packet loss, will retransmit the packet. UDP, a far simpler cousin of TCP, will attempt to deliver the data but will not provide guarantees nor will it retransmit data in case of loss.

Layers 5-7, Application layers, are beyond the scope of this paper.
In the example above, the application sends data to a specific IP address and port number. Layer-4, TCP, takes this data, breaks it into segments, adds a TCP header and sends it down to Layer-3 (IP). IP adds an IP header then sends the packet down to Layer-2 (Ethernet). Ethernet adds its own header, computes CRC over the entire packet and appends it to the end. The complete packet is sent down to Layer-1 which transfers it over physical medium to a device at the other end of the link.

Basics of Ethernet Communication

If only two devices need to communicate over Ethernet, they can be interconnected directly, point-to-point. Figure 3 shows a point-to-point interconnect.

Older Ethernet technology, such as 10BaseT/100BaseTX, required that receive and transmit signal pairs be swapped on cross-over cables. Newer Ethernet technology, 10BaseT/100BaseTX/1000BaseT performs swapping in hardware; consequently, either the transmit or the receive cable can be used for interconnection. It is important to note that for twisted pair Ethernet cables, such as CAT-5e, 10BaseT and 100BaseTX use four wires: two for the transmit signal and two for the receive signal. The baud rate on such cable is 12.5 Mbit/sec (Mb/s) and 125 Mb/s. Gigabit Ethernet, 1000BaseT, on the other hand, uses all 8 wires, 4 differential pairs, to transmit and receive signals at the same time. 1000BaseT also uses 5-voltage levels to encode bits. As a result 1000BaseT uses the same fundamental transmission frequency as 100BaseTX while achieving a 10x speed increase.

Ethernet uses the concept of full-duplex and half-duplex communication. Half-duplex means that data can be either sent or received, but not both at the same time. Full-duplex means that data can be sent and received at the same time, effectively doubling the data throughput. This concept becomes important for the discussion of Ethernet Hubs versus Switches later in this paper. At this point, it is important to note that interconnecting two devices directly point-to-point enables use of either half- or full-duplex communication.

A hub or switch is needed if more than two devices must be interconnected over Ethernet. Figure 4 depicts a system consisting of four Single Board Computers (SBCs).

Note that, consistent with the previous discussion, each Ethernet port has a unique MAC address. Ethernet packets have a header consisting of:
- Destination MAC address
- Source MAC address
- Protocol type

This information is sufficient to indicate packet origination and destination. When the Ethernet device receives a packet, it compares the destination MAC address to its own MAC address. If they match, the packet is processed and forwarded up to Layer-3. If there is a mismatch, the packet is dropped.
Ethernet Switching

You may recall that in the earlier Network Layer discussion of Layer-3, I said that applications address packets to IP addresses, not MAC addresses. In fact, the application has no knowledge of MAC addresses at all. This is still true. The protocol stack has a protocol called Address Resolution Protocol (ARP) which automatically finds a MAC address associated with each IP address. More details of ARP will be provided later in this paper. At this point it is important to note that packet addressing in Ethernet is accomplished using MAC addresses, while addressing at the application is done using IP addresses, and that ARP allows automatic association of MAC and IP addresses thus providing desirable abstraction.

MAC Address versus IP Address and ARP Protocol

As noted earlier, the application sends data to an IP address and has no knowledge of MAC addresses. This abstraction provides several benefits. For instance, every time an Ethernet device is replaced (creating a new MAC address) the application does not need to change. Also, from the application perspective, Ethernet could be replaced by another protocol, such as ATM, Fibre Channel or InfiniBand, but the IP addresses would remain the same.

With Ethernet, when an application sends data to an IP address, translation must be performed by the Address Resolution Protocol (ARP) to associate the MAC address. Typically, the ARP is part of the Operating System (OS) protocol stack as are the TCP, UDP and IP protocols. Therefore, if a standard OS is used, ARP functions automatically. When a MAC address for a specific IP address is required, ARP sends a broadcast Ethernet packet to all devices on the same Ethernet network. In this case, the Ethernet packet destination MAC address is set to “all devices” (broadcast). The source MAC address is set to the sending device’s MAC address. Additionally, the packet includes a destination address and source IP addresses. Each Ethernet device receives the ARP Request packet and forwards it up the protocol stack. ARP will look at the destination IP address and, if it matches its own IP address, will send back an ARP Reply packet. All other Ethernet devices will simply drop the ARP Request packet. When the ARP Reply is sent, it is sent to a requester’s MAC address. The requester receives the ARP Reply and extracts the required MAC address. ARP stores this MAC-IP address association in its ARP table for future reference and sends a data packet to the now known destination MAC address. After a period of time, the ARP table will time out and the MAC-IP address association will need to be renewed. This is useful if in the meantime a failure has occurred and the Ethernet controller was replaced. Replacing the controller will change the associated MAC address and ARP will be able to automatically update its MAC-IP address association.

Switch versus Hub

An Ethernet Hub is an interconnect device that allows the connection of two or more Ethernet devices. Ethernet Hubs provide only half-duplex communication. In this way it is similar to a conference call; everybody dials in to a number but only one person can speak at any given time … well, humans tend to bend the rules, but Ethernet devices are more law abiding. Hence, if there are 10 Ethernet devices connected together over 100BaseT, the maximum aggregated data throughput in the system is 100Mb/s.

Ethernet Switches also are used to interconnect two or more Ethernet devices. In the case of Ethernet Switches, each device can have a dedicated full-duplex channel to other devices. So instead of having a conference call, people are called directly. Multiple conversations can take place at the same time. More importantly, it is full-duplex, people can speak and listen at the same time. Consider the same system consisting of 10 devices interconnected over the same 100BaseT Ethernet but using full duplex communication. For this system throughput is now:

\[10 \times 100\text{Mb/s} \times 2(\text{full duplex}) = 2000 \text{Mb/s}.\]

Here it is important to note that maximum realistic throughput will depend on communication patterns. For instance, if nine devices want to talk to the same device, then the maximum throughput will be 200Mb/s (considering full-duplex).

In addition to significant data throughput increase, Ethernet Switches interconnect Ethernet devices running at different speeds. For instance, a legacy controller using 10BaseT can still interconnect with a cluster of SBCs communicating using 1000BaseT. Data throughput in this case will be limited to 10Mb/s full-duplex; however, the SBCs will be able to communicate with each other at the full speed of 1000Mb/s full-duplex.

How do packets get to their destination? With Hubs, the answer is fairly simple. When a Hub receives a packet, it forwards it to every other port. Every device in the network receives the same packet. Then, the Ethernet controller checks the destination MAC address to ensure the packet has arrived at the correct destination. If the addresses match, the packet is received; if not, the packet is dropped. Consequently, communication over a Hub results in a lot of wasted packets.

Ethernet Switches, on the other hand, are more intelligent devices in that they attempt to forward a packet only to the intended destination.

Considering all of these technical aspects and today’s Ethernet technology, it is important to note that Ethernet Hubs are on the official list of endangered species and are almost completely extinct.

Ethernet Switching
Layer-2 Packet Switching
Unmanaged Ethernet Switches
Two terms are important to any switching discussion: Unmanaged Switch and Managed Switch.

An Unmanaged Switch has no management interface available to the user for configuring and controlling the switch. Unmanaged switches are also limited in terms of supported switching protocols. An Unmanaged Switch is the simplest Ethernet switching device. Once power is applied, it will switch the packets to the proper destinations.

A Managed Switch, on the other hand, has one or more management interfaces that the user can use to configure the switch. Additional switching protocols could require user intervention to setup and configure parameters. Managed Switches and the most popular protocols will be discussed later in this paper.

Figure 5 shows the same four devices interconnected through the switch. Again, note that each Ethernet device has its own unique MAC address. Power was just applied, hence the switch has no knowledge about the system architecture and which MAC addresses can be reached on what port. Its internal address table is empty.

Let’s assume that an application on SBC A wants to communicate with an application on SBC B. The application on A will send a packet to B’s IP address. As described earlier, the ARP protocol will be used to determine which MAC address should be used to reach B. At the Ethernet level, however, SBC A sends a packet with destination MAC: 01 and with source MAC: 02. When the switch receives the packet destined for SBC B MAC: 02, it does not know on which port the address is reachable. For that reason, it sends the packet on all ports except the port it was received on. By broadcasting the packet this way, the switch ensures that the packet reaches its destination, SBC B. At the same time, the switch also looks at the source MAC address the packet contains and adds it, as well as the port it was received on, to the internal Address Table. SBC B will indeed receive this packet, will compare the destination MAC address with its own MAC address and will process the packet forwarding it up to the IP layer. SBCs C and D will also receive this packet, but they will look at the destination MAC address and will drop it. At this point, the switch’s Address Table will be similar to the one shown in Figure 6.

At some point, SBC B will reply to SBC A. It will send a packet with destination MAC: 01 and source MAC: 02. The switch will receive this packet, look at the destination MAC address, which is 01, check its address table and forward it only to port 1. Also, the switch will look at the source MAC address and will store that address in its address table. From this point on, the switch knows how to reach SBC A and SBC B. All communication between these two devices will be point-to-point, excluding SBCs C and D thus conserving bandwidth. Figure 7 shows the switch’s address tables after B replies to A.

Using the same process, the Unmanaged Ethernet Switch will discover the rest of the network and gradually complete its Address Table. This discovery process is called Autolearning. switches incorporate a timer that times-out each entry in the table if a packet from a specified MAC address or associated switch port is not received for some time. In this case, if the Ethernet device is disconnected, its address is removed from the table, keeping the tables tidy. The Ethernet device can also be moved to another port on the switch and the switch will update its Address Table as soon as it sees the first packet originating from the new location.
As demonstrated, Unmanaged Ethernet Switches are able to operate straight out of the box, without any prior configuration. This ability greatly simplifies Ethernet network installation and configuration. Minimal Ethernet knowledge is needed to setup and operate the network.

**Ethernet Topology Loop Problem**

Although an Ethernet network using Unmanaged Switches is very easy to set up and operate, a few issues such as Network topology and loops need to be considered carefully.

What is a topology loop? Figures 8 and 9 show two possible topologies with loops.

Loops may be created in network topologies for very good reasons. Redundant links and increased throughput between switches are two such reasons. But unless proper Layer-2 management protocols are used (Spanning Tree Protocols, Link Aggregation Protocol) or the switches are properly configured manually, the topology loops will take down the network. Consider Example 2 in Figure 9 immediately after power up. The switches are Unmanaged and their address tables empty. Device A sends the first packet to device B. As described in the previous sections, switch 1 broadcasts the packet to switch 2 and 3. Switch 2 broadcasts the packet to switch 3 and device B. Switch 3 also broadcasts the packet to Switch 2 and devices C and D. Switch 2 will notice that A’s source MAC address moved from port 1 to port 2 and will adjust its Address table. Switch 2 also broadcasts the packet to Switch 1 and device B. Switch 3 performs a similar operation with the copy of the packet received from switch 2. The end result is packet copies circulating in an endless loop and incorrect Address Table entries.

In summary, if Unmanaged Switches are used, network topology loops must be avoided. Topology loops can be very useful for increasing connection bandwidth, adding link failover or increasing network resilience to failures when used with proper network protocols and Managed Switches.
Port Trunking and VLAN, Configuration Managed Ethernet Switches

As the name implies, Configuration Managed Ethernet switches add the ability to configure, control and monitor switch operation. Historically, such switches had only a microcontroller and offered a management interface through a Command Line Interface (CLI) over an RS232 serial port. Configuration and status checking were the only allowed management functions. Capabilities to receive and process protocol packets were not included; therefore, the switches could not run Layer-2 protocols.

Newer Configuration Managed Switches include a processor that is connected by a data bus (PCI bus or I2C and an Ethernet port bundle) to the switching silicon. Today’s Configuration Managed switches can receive, process and send Bridge Protocol Data Units (BPDUs). BPDUs are packets specifically meant for switch-to-switch communication and used to implement Layer-2 switch protocols. Layer-2 switch protocols include:

- Spanning Tree Protocol (STP)
- Rapid Spanning Tree Protocol (RSTP)
- Multiple Spanning Tree Protocol (MSTP)
- Generic Attribute Registration Protocol (GARP)
- GARP Multicast Registration Protocol (GMRP)
- GARP VLAN Registration Protocol (GVRP)
- Internet Group Management Protocol (IGMP) Snooping
- Simple Network Management Protocol (SNMP)

Configuration Managed Layer-2 Ethernet Switches typically have a few management interfaces that allow for control and monitoring of switch operation. Those interfaces can be:

- CLI over RS232 serial interface
- SNMP over a dedicated management Ethernet port or over a switch Ethernet port
- CLI/Telnet over a dedicated management Ethernet port or over a switch Ethernet port
- HTML over a dedicated management Ethernet port or over a switch Ethernet ports

Management interfaces are usually secured by a password during login or by Secure Shell protocols (SSH). Since any device connected to the switch can try to gain switch management privileges, this is a significant threat to network security and must be properly restricted.
VLAN Description

Virtual Bridged Local Area Networks (VLANs) are defined by the IEEE 802.1Q specification. VLANs:

- Allow the Ethernet network to be partitioned into smaller logical groups and restrict traffic between them. In other words, two logically separate Ethernet networks can be created from one physical network. This is typically done for security or for traffic segregation purposes (separate Control and Data networks).
- Add Quality of Service (QoS) tags to the packet, enabling up to eight traffic classes. Switches typically have up to eight packet queues per port and allow preferential treatment for high priority packets.
- Are compatible with Ethernet devices and switches that do not support the protocol.

To enable a VLAN, a VLAN Tag is added to the packet Ethernet header immediately after the MAC address and Type fields. The VLAN Tag is two Bytes long and has User Priority and VLAN Identifier fields.

VLANs can be used in two ways. The first way is for devices to send VLAN tagged packets using switches configured to prioritize packets based on the packet User Priority field. In addition, based on the VLAN Identifier field packets, switching can be within a different logical network that can be a subset of the whole Ethernet network. Although packet prioritization is very useful, most Ethernet network implementations in embedded applications do not use VLAN tagged packets.

VLAN tagged packets are more frequently used in data communication networks and Enterprise applications.

A method more applicable to embedded applications is to use untagged packets and port-based VLAN tagging within the switch. This means that the switch can take any untagged Ethernet packet and based on the switch port it arrived at, assign a default VLAN ID. Based on the VLAN ID, the switch can be logically partitioned into two or more networks without packets crossing the boundary between the two networks. A packet can be switched only to the ports that have the same assigned VLAN ID. When the packet leaves the switch, the VLAN ID is removed and the packet becomes once again an untagged Ethernet packet. Applications could use packet tagging and switch logical partitioning for traffic restriction and security. Consider the example in Figure 10 in which there are four Ethernet devices, but only device A communicates with device B, and device C only communicates with device D. More importantly, device C cannot see traffic between devices A and B.

Without switch segregation, when the switch broadcasts a packet from A, that packet goes to devices C and D as well. With port-based VLAN tagging, broadcasts are limited to the ports belonging to the same VLAN ID - in this example, only port 3. Perhaps this example is oversimplified, but the same Ethernet network segregation principle can be applied to more complex implementations involving multiple switches, multiple links between switches (avoiding loop problems) and separate data and control traffic paths.

---

**Figure 10: Port - based VLAN Tagging Example**
**Link Aggregation and Trunking Description**

Link Aggregation is defined by the IEEE 802.3-2005 specification, also known as IEEE802.3ad. Link Aggregation takes two or more links and aggregates them into one bigger pipe. The “big” pipe increases connection bandwidth and provides failover capability in case one of the links breaks. Keep in mind that as far as the IEEE specification goes, these links must be the same speed and must be connected to the same two endpoints, switch or device. Some customized implementations allow links to be aggregated between different devices, for instance, go to two different switches. Those, however, are non-standard implementations; see section 9.2 for more information.

Link aggregation can be accomplished by either running the standard Link Aggregation Control Protocol (LACP) on both end points of the link, or by configuring the Trunk manually. An example of the link aggregation is shown in the Figure 11.

In Figure 11, the link aggregation connection between switch 1 and 2 doubles the bandwidth as well as provides resiliency against link failures. This type of implementation can be used for military vehicle installations in which Switch 1 and the devices attached to it represent one compute node in one part of the vehicle. Switch 2 and attached devices represent another compute node in a different part of the vehicle. The two nodes can be connected with two links routed on different sides of the vehicle to protect against link failure.

**Spanning Tree Protocols (STP, RSTP, MSTP)**

Section 7.2 described the Topology Loop problem. Spanning Tree Protocols (STPs) are designed to address exactly that problem. When an STP is enabled, whether it is STP, Rapid Spanning Tree Protocol (RSTP) or Multiple Spanning Tree Protocol (MSTP), it resolves any network topology into a tree structure that avoids loops. Real trees do not have branches that grow in a loop, yet each leaf or node on the network has access to any other leaf or node on the network through a number of branches. To create a tree topology, switches disable certain ports that would otherwise cause creation of a loop. If a link or switch fails, the whole network topology would re-converge to a new topology, again avoiding loops. Figure 12 illustrates this process.

![Figure 11: Link Aggregation Example](image-url)
Links indicated by red crosses have been disabled by the STP, resulting in a loop-free network. In this example, all traffic flows through Switch 1, a Root Switch. Even though a direct link exists between Switches 3 and 4, this link will not be used for data communication. Now assume that the link between Switches 1 and 2 is damaged. STP will create a new topology similar to that shown in Figure 13.

Again, connectivity between all nodes is restored. The STP protocols handle all the network topology resolution issues. Users do have limited control and can enforce certain resolution of a preferred network topology by assigning switch priority, port priority and link cost through a management interface.
What are RSTP and MSTP?
Spanning Tree Protocol is problematic in that it takes time to converge and to start forwarding packets. There are a number of timers that have to expire and as a result it takes around 30 seconds before the first data packet can be forwarded. Rapid Spanning Tree Protocol (RSTP) optimizes the convergence algorithm and reduces this time to less than 1 second. Finally, as I stated in the VLAN discussion, a network can be logically partitioned into multiple networks – this is where Multiple Spanning Tree Protocol (MSTP) delivers value. MSTP runs on a partitioned network creating a different tree topology for every VLAN.

Spanning Tree Protocols along with multiple links between the switches can improve network reliability and resilience to link and switch failures. Does STP remove single points of failure? NO. Each device has only a single connection to the switch. This link and the switch are single points of failure for the device. See Concepts of Redundancy later in this paper for ways to avoid single points of failure.

Simple Network Management Protocol (SNMP)
The Simple Network Management Protocol (SNMP) is a protocol that can be used to manage the switch. Do not be misled by the name, SNMP is anything but simple. Over time SNMP has evolved to be capable of fairly complex management tasks in networking environments. For our purposes we will concentrate on the basics of SNMP and how it applies to switch management.

From the SNMP perspective, management information on the switch is stored in Management Information Bases (MIBs). MIBs contain information such as switch statistics, counters, link configuration and status, VLAN tables, Port Trunking tables and so on. In the SNMP world, there are devices that act as Agents and those that act as Managers. As the name implies, Managed Ethernet Switches are devices that are being managed; therefore, they are Agents. A station that is used to access and manage switches is called a Manager. A laptop computer connected to the network and running an SNMP Manager application would act as a Manager. The most basic operation that a Manager is allowed to perform is to read data from a MIB (Get) and to write data to a MIB (Set). An agent, or switch in our case, has the ability to send an unsolicited message (Trap) back to the Manager.

Let’s look at a simple example. Assume that the Ethernet network shown in Figure 13 is installed in a plane and we need to add a legacy sensor. Although the sensor has an Ethernet interface, that interface does not support Ethernet Autonegotiation and does not comply with the standard implementation. To properly interface with this device, the port that is used to connect to this device must be forced to operate in 10Mb/s half-duplex mode. Using a laptop computer running an SNMP Manager, after a connection to an applicable switch is established, the Set command is used to configure the port for 10Mb half-duplex operation. The sensor is supposed to broadcast a packet every second, but the Mission computer is not receiving it. To investigate the problem, the port Link status can be reviewed using the SNMP Manager and executing the Get command on the port status MIB. If status is “Up”, the Get command can be executed on the statistics counter MIB for this port. If the broadcast packets were received on this port, but dropped due to CRC errors, the cable attaching the device and the device itself can be investigated.

Another example of SNMP use would be to configure a switch to issue a Trap message when its operational conditions move outside of the limits; for instance, if board temperature was exceeded. In this case, the switch would send an SNMP Trap and the SNMP Manager would receive it. Then the management station could warn an operator of the potential failure.

It is important to note that switch manageability greatly depends on the MIBs that are implemented. In turn, MIB implementation depends on the available hardware functions. For instance, if the switching silicon does not have a specific counter, then that specific MIB can not be fully implemented. In an attempt at standardization, IETF has defined a number of standard MIBs. This is a shotgun approach that attempts to provide as many profiles as feasible. However, in real implementations not all of these profiles apply; consequently, implementations may have a subset of what is defined in a standard MIB.

Layer-2 protocols could require prior configuration, such as setting STP Bridge priority, configuring VLANs or port Trunks. Once such a configuration is created and the network design complete, configuration parameters can be stored locally on the switch or alternatively exported to a file which can be automatically loaded to new devices during manufacturing or testing.
Layer-3 Packet Switching (Routing)

Layer-3 (IP) switching, also known as routing, switches packets based on IP addresses instead of Ethernet MAC addresses. Older switches enabled IP level routing in software by forwarding a packet to the processor which in turn performed the required routing operations. Currently, IP routing is performed in hardware, at interface line rate, similarly to Layer-2 (Ethernet) switching. The processor is only used to process specific routing protocols and configure IP routing tables in the hardware. Both terms, routing and switching, are used fairly interchangeably.

Why is routing needed? Doesn’t Layer-2 switching accomplish all that is necessary, especially for embedded applications? No. Layer-3 switching supports networking protocols such as IP Multicasting that improve data transmission efficiency; see Layer-3 Packet Switching (Routing) for information on IP Multicasting. Additionally, with Layer-3 switching the network can be partitioned into multiple smaller domains, each of them having different IP Subnet numbers. Some embedded networks are quite large and complex. As the number of devices attached to the network grows it may be come necessary to partition the network into smaller pieces. Navy ships, for example, have large embedded networks. To a large extent, these networks resemble enterprise networks with multiple departments and hundreds of devices attached to the network. Now consider Layer-2 protocols, such as ARP, which use Ethernet broadcast packets. With increased size, it is desirable to limit how far these broadcast packets are transmitted, otherwise a significant chunk of network bandwidth will be consumed by broadcast traffic.

IP Address and Network Address

An IP address is a 32-bit long field. It is presented as four 8-bit values in decimal notation separated by dots (dot decimal notation), such as 192.168.0.1. Notice that since each of the four digits has only 8 bits, the maximum decimal digit is 255. In the past, the first few bits of the IP address represented the Internet Address Class, such as A, B, C, D or E. Today, Classless address notation has replaced the concept of address classes; however, Class D, which means that IP address is a Multicast address, and Class E, which means that this IP address is reserved, are still in use. A Class D IP address is indicated by the first four bits being 1110 in an 8-bit digit, which when translated to decimal notation means digits from 224 to 239. So if the IP address starts with 225, for example 225.2.3.4, it is an IP Multicast address.

Following Classless IP address notation, an IP address is split into two parts. The most significant part of the address is the Network Identifier. The remainder of the address is called the Host Identifier. The Network Identifier is the network address used by routers to route the packet. The Host Identifier is the host address within a specific network. An analogy is a phone number with the area code being equivalent to Network ID and actual number being equivalent to Host ID. A Subnet Mask indicates which part of the address is a Network Identifier. For example, when the IP address is: 192.168.1.21 and the Subnet Mask is: 255.255.240.0, the Network Identifier is the first three digits 192.168.1 and the Host Identifier is the last digit: 0.0.2.13. Consider the IP address: 192.168.178.13 and Subnet: 255.255.240.0. Figure 14 shows the IP address in binary.

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Subnet Mask</th>
<th>Network Identifier</th>
<th>Host Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.178.13</td>
<td>255.255.240.0</td>
<td>1100 0000 1010 1000 1011</td>
<td>0010 0000 1101</td>
</tr>
<tr>
<td>0.0.2.13</td>
<td>0.0.0000 0000</td>
<td>192.168.178.13</td>
<td>0.0.2.13</td>
</tr>
</tbody>
</table>

Figure 14: Network Address and Host Address Example

The resulting Network Identifier: 192.168.176.0 and Host Identifier: 0.0.2.13.
It is important to note that there are Public IP addresses, Private IP addresses, Special IP addresses and Reserved IP addresses. Public IP addresses are valid in the public Internet domain and are assigned by authorities. It is generally understood that addresses from this range cannot be arbitrarily taken and assigned to hosts that are connected to the public Internet. Private IP addresses are blocks of addresses that companies and individuals are encouraged to assign and use as they please. These addresses are not routable on the public Internet and are valid only within an organization. One of the most popular Private address ranges used today in home setups is from 192.168.0.0 to 192.168.255.255. Special IP addresses are used for a specific purpose; for instance, loopback or broadcast. They typically have all binary 1s or 0s so avoid using numbers such as 127.0.0.0 or 255.255.255.255. Reserved addresses are what the name implies, reserved for future use.

### IP Address and Network Address Assignments

This section is not intended to be a guide to network design, but is intended to give a feel for how addresses can be assigned in a larger network. The assumption here is that this is a private network that is not directly connected to the public Internet; the private address range will be used. In this example, the network is a semiconductor processing fab.

The installation consists of 10 wafer processing tools, a wafer storage facility, a control center and executive offices. Both Layer-2 and Layer-3 switches are used in the installation. Note, that semiconductor-processing tools are small networks in themselves. Such tools have a number of computers and components that communicate over Ethernet and IP. From the IP addressing perspective it makes sense to partition the network into Subnets and assign each Wafer Processing Tool its own subnet. As there are 10 tools, they are given subnets from 192.168.10.0 to 192.168.90.0 with a subnet mask 255.255.255.0 indicating that the first three decimal digits are the Network ID and last digit is the Host ID. Having one decimal digit for Host IDs allows up to 254 (excluding all 1s and 0s) individual Host IDs for each tool. A similar concept is used to assign Subnets to the Wafer Storage, the Control Center and Executive Offices. Note that individual IP assignments are not shown in this example. Individual IP addresses can be assigned either manually or automatically using the Dynamic Host Configuration Protocol (DHCP). Routers can be manually programmed for the assigned IP Subnets, then using the DHCP, each host requesting an IP address from the router would be assigned one. Since a router is aware of the Subnet the device is on, it will assign an IP address with the Network ID belonging to this Subnet. For example, in Wafer Etch Tool 1, which is on the 192.168.10.0 Subnet, a Transport module might be assigned the IP address 192.168.10.13. In this case, its Host ID would be 0.0.0.13.

In many embedded applications, implementers choose to have full control over the assignment of IP addresses and assign them manually.

The network example shown in Figure 15 will be used in the following sections that describe how Routers works. A similar network design approach can be used for other embedded networks such as airplanes and ships.

### IPv4 versus IPv6

IPv6 was developed to increase IP address space, to add new security features and to make the IP header more flexible, enabling new applications. The IPv6 specification has been around for some time, but until recently hasn’t had much traction because of the huge installed base of equipment supporting IPv4 and the enormous effort required to upgrade to IPv6. The industry has been very creative in improving IPv4 to cope with address space limitations using concepts of Network Address Translation (NAT), Subnetting and Supernetting. Today, with telecommunication and military networks switching to IP, and all of the new portable and mobile devices being introduced that expect an IP address, the time for IPv6 adoption has finally arrived.

What is the difference between IPv4 and IPv6? First of all, IPv6 increased the IP address length from 32 bits to 128 bits making the number of possible addresses extremely large. The IPv6 datagram header is different and allows for easy extension. Such extension enables Authentication and Encryption extension headers. The IPv6 header also has Quality of Service (QoS) features, which are lacking in IPv4.

Since Ethernet switches are the focus in this paper, it is important to note that the transition to IPv6 requires new Layer-3 Ethernet switching silicon. Modern routers or Layer-3 switches perform IP datagram routing in hardware at port line rate; therefore, the silicon has to support IPv6 header processing. It is also important to note, that as an interim step, some switches have a feature to recognize the IPv6 header and forward it to the processor. The processor can parse this header and perform datagram routing, but the performance level will be only a small fraction of the line rate. A word of caution — when it comes to IPv6 routing make sure the routers perform IPv6 routing in hardware and are capable of sustaining the line rate.

### Basics of Layer-3 Switching (Routing)

How does Lay-3 switching (routing) work? For this discussion refer to Figure 15. Assume a Transport Module on the Wafer Etch Tool 1 (let’s call it TM1) needs to send a request to a Wafer Storage Unit 1 (SU1). TM1 knows the IP address of SU1, which is on the Network 192.168.100.0, for instance 192.168.100.5. So the message travels down the protocol stack on TM1, arrives at the IP layer and the IP datagram...
is created. The IP datagram has the destination IP address: 192.168.100.5. Next, before the IP datagram gets to Layer-2, Ethernet needs to know this packet’s destination MAC. The protocol stack looks at the destination Network ID and realizes that the packet is on a different network; hence, the packet should be sent to the default gateway which is Layer-3 switch. The Ethernet packet is sent out with the destination MAC address of the Layer-3 switch.

The Layer-3 switch receives the packet and after looking at the destination MAC address realizes that the packet is destined to itself. Next, the switch looks into the Layer-3 (IP) header and performs a “Longest Prefix Match”, an operation that involves searching the IP Routing table, which contains the Network ID and associated Subnet Mask, and finding the match that has the longest Network ID. Note that hosts, which are on directly attached (from IP perspective) networks to the router, such as hosts in Wafer Storage, will have a longest prefix since there will be a match for the whole 32-bit IP address. This is opposed to hosts in Executive Offices or the Control Center which are behind another Router and on indirectly attached networks. Once the Longest Prefix Match is found, the entry in the Router’s table will have the port number to which the packet should
be forwarded as well as the new destination MAC address (SU1 MAC address in this case). The Router will replace source and destination MAC address fields, will re-compute IP and Ethernet headers and will send out the packet. The SU1 packet will hit another Layer-2 switch, but this switch will only perform the Ethernet switching operation, as was described in earlier sections of this paper, completely ignoring the IP header and will forward the packet to its final destination.

This example shows that the Router has to perform a number of lookups and searches, as well as re-computing new headers and check sums. These are compute intensive operations and most Routers today perform them in hardware. For routing to work properly, the Router must be configured and know what IP addresses and networks can be reached on which port. Small networks can be configured manually through management interfaces, but larger networks need to use routing protocols.

**Bandwidth Efficiency through IP Multicasting (IGMP and IGMP Snooping)**

Multicasting is a mechanism for forwarding a packet from one source to multiple destinations. Multicasting is a nice concept that enables efficient use of the bandwidth without needing to duplicate traffic. It is most frequently used in multimedia applications such as audio and video distribution.

Consider this example: an air craft has a camera or radar from which data has to be sent to storage and a few processing stations. One way to achieve this is to send Unicast packets to each of these destinations individually. The result would be that the same packet has to be sent three times. Another alternative is to broadcast the data to all devices on the network, but then bandwidth will be wasted on the links that have no devices interested in these packets. Multicasting allows sending data only to the devices that have signed up as receivers. Figure 16 illustrates this concept.

---

**Figure 16: Multicasting Example**

[Diagram of a network illustrating multicasting]
In this example, two radars continuously capture images. These images are being stored in two storage devices. A Mission computer processes both of these streams. Weapon Station 1 is only interested in Radar 2 data. The Pilot Computer and Weapons Station 2 are not interested in any data at this time. As can be seen from the Figure 16, Switch 1 will forward Radar 1 data (red) only to Switch 2, while Radar 2 data (blue) will be forwarded to both Switch 2 and Switch 3. An alternative to Multicasting would be to use Unicast, in which case Radar 1 would have to send two copies of the same data and Radar 3 would have to send three copies of the same data. Considering the bandwidth required to transmit graphical data, this would be a huge waste of bandwidth resources.

How is Multicasting implemented? Two protocols are primarily used for Multicasting: Group Multicast Registration Protocol (GMRP) and Internet Group Management Protocol (IGMP). GMRP is a Layer-2 protocol, while IGMP is a Layer-3 (IP Layer protocol). Most Multicast implementations today use the IGMP protocol.

IGMP protocol is defined by the IETF organization. As of this writing, the latest version was IGMP v3 defined by RFCs 3376. Multicast operation is based on the devices registering for a specific Multicast group. To register, a device sends out a Membership Report packet. This packet tells the routers that all Multicast traffic destined to a specified Multicast address shall be forwarded on the port on which the Membership Report packet was received. The router in turn propagates the Membership Report to other routers all the way to the source of the Multicast traffic. Routers periodically check if there are any more recipients of the Multicast traffic left attached to each port. This is the short description of the IGMP protocol. There are obviously number of rules and nuances in order to reduce Membership Report duplication from multiple devices and to avoid all devices sending Membership Reports simultaneously.

IGMP Snooping is a method by which the switch takes a look at the IGMP Membership Report packet, processes it and updates its Layer-2 switching tables accordingly. To support IGMP Snooping, the switch does not necessarily need to run the complete IGMP protocol. Consider the previous example (Figure 16). When subscribing to the Multicast Group for Radar 2 data, Weapons Station 1 will send out a Membership Report packet, Switches 3 and 1 will conduct IGMP Snooping on the packet, Radar 2 will receive the packet and start the Multicast transmission.

IP Routing Protocols
The main goal of IP routing protocols is to exchange information on how to reach specific networks. A number of routing protocols are used today and at the top level they can be grouped into two groups: Interior Gateway Protocols and Exterior Gateway Protocols. Interior Gateway Protocols are used to exchange route information within an Autonomous System (a set of routers and networks under the same...
administration, for instance a company network. Examples of Interior Routing Protocols are RIP and OSPF. Exterior Gateway Protocols are used to exchange route information between Autonomous Systems; for example a BGP-4 routing protocol.

Routing Information Protocol (RIP)
Routing Information Protocol (RIP) is one of the simplest routing protocols used today. RIP is based on the “Cost” to reach another network. Typically one hop has an associated cost of 1, in which case RIP will converge to a shortest path, the least number of hops, to reach another network. In RIP, each router advertises destinations it can reach and the associated cost. Once advertisements from the neighboring routers are received, the router computes the new shortest paths to reach each destination by adding its own costs in addition to the costs received from neighbors and sends out updates to each neighboring router. Over time, each destination and the shortest path to reach it is known by each router. If there is a change in the network, routes will be recomputed and a new equilibrium will emerge.

RIP is a good fit for small embedded networks.

Open Shortest Path First Protocol (OSPF)
Open Shortest Path First protocol (OSPF) improves on RIP by providing faster and loop-less convergence, allowing multiple path “cost” metrics and by supporting multiple routes to the same destination and enabling load balancing. OSPF reduces protocol related traffic in large networks since each router in OSPF has a complete map of the network and performs route computation internally. In OSPF, each router advertises the state of its interfaces (their “costs” and their connectivity). These link state advertisements are flooded through the network. Each router computes its own routing table.

OSPF is the most popular Interior Gateway Protocol used today.

Border Gateway Protocol (BGP-4)
Border Gateway Protocol version 4 (BGP-4) is used mostly to communicate routing information between Autonomous Systems, such as a company network and Internet Service Providers, or between different Internet Service Providers. Use of BGP-4 in embedded networks is fairly limited. BGP-4 operates by establishing separate TCP connections with each router with which it needs to communicate. In BGP-4, each router informs neighbors about new active routes, routes that were deactivated, and confirms that it is still alive. BGP-4 allows a high degree of control and preference setting. Preferences can be based on Service Level Agreements that are set in place between different ISPs; hence, route selection is impacted by conditions in such agreements.

Concepts of Redundancy
Many embedded applications require redundancy and, often, complete avoidance of a single point of failure. Some standard protocols provide a level of redundancy, but typically they do not eliminate single points of failure. The purpose of this section is to discuss some of these implementations and what proprietary implementations can offer.

Spanning Tree Protocols in Context of Redundancy
In the section that described Spanning Tree Protocols I mentioned that these protocols do provide redundancy features. For instance, if a link between two switches is broken, the network will converge to a new tree topology. RSTP offers such conversion in less than a second. To take advantage of STP, the network should have redundant links between the switches. An example of such a network is shown in Figure 18.

One obvious issue is that STP does not eliminate single points of failure. There is only one link connecting the Ethernet controller and the Switch. Also, there is only one switch to which the Ethernet controller can connect. One alternative implementation is to have two Ethernet controllers on each device and route them to a different switch. Figure 19 depicts this example.

Keep in mind that both Ethernet ports will be visible to the Operating System. The Operating System will not be able to choose one port for communication and failover to the other port if failure occurs. The applications will need to be aware of failover requirements and implement such functionality. Often this is not very desirable.
Ethernet Switching

Figure 18: Example of Network with STP based Redundancy

Figure 19: STP Based Redundancy Example with Multiple per Host Ports
**Link Aggregation in Context of Redundancy**

Link Aggregation is another way to gain redundancy. In Link Aggregation two or more ports can be bundled into the Trunk and provide both higher bandwidth and failover functionality. The advantage of using Link Aggregation is that once the Trunk is formed the Operating System has no knowledge of the fact that there are two ports working as one. If one port fails and the link is damaged, all traffic will automatically fail over to the second link. The biggest drawback of Link Aggregation is that the IEEE defined standard implementation requires that both links be connected to the same device, the same switch as far as our discussion is concerned. Such a switch becomes a single point of failure. Figure 20 is an example of redundancy through Link Aggregation.

**Software Based Failover Implementation**

The Spanning Tree Protocol and Link Aggregation discussions show that these two standard technologies can improve Ethernet network reliability and provide some level of redundancy. However, standard technologies do not eliminate single points of failure. To address this issue, some companies have designed proprietary Ethernet controller implementations that do eliminate single points of failure. Software-based port failover is one such proprietary implementation. The concept behind this approach is to provide a driver-level software that would take two Ethernet controllers, bundle them into one link similar to Link Aggregation, but allow them to connect to different switches. Figure 21 is an example of this implementation.

---

![Figure 20: Redundancy through Link Aggregation](image1)

![Figure 21: Vendor Specific Failover Implementation, eliminating Single Points of Failure](image2)
How to Build a Real-time Ethernet Network

The advantage of this implementation is that the Operating System only sees one interface. A software shim layer resides right on top of the Ethernet drivers and performs link monitoring and failover tasks. Failover time depends on where in the network the failure occurs. If one of the directly attached switches or links fails, then the failover software is notified through an interrupt and reroutes the traffic over the second interface. To protect against the failure of the link between two switches, software periodically sends out link monitoring packets. If a link failure occurs, failover time depends on how often the link-monitoring packet is being sent. This time interval is software controllable and can be as low as 50ms.

The main drawback of this implementation is that all devices requiring single point of failure avoidance must use a failover software shim layer, which is only available on a limited number of controllers and for a limited number of operating systems. Software implementation also adds processor overhead, the amount depending on the size of the network. This overhead is fairly small and for a ten device network it is well under 10% of the CPU utilization.

Hardware Based Failover Implementation

Although software-based failover implementation eliminates single points of failure in the network, it requires a software shim layer to run on the host processor. In some cases, it may be desirable to perform failover functions completely in hardware. To enable hardware-based failover functionality, an intelligent network interface card has to be used. This card has two or more Ethernet controllers and an onboard processor that runs failover firmware. To the host system the controller looks like a single-port Ethernet controller. Failover firmware running on the host processor selects which Ethernet port should be used for communication, monitors the health of both Ethernet links and switches to the second Ethernet port as soon as failure occurs.

Network implementations using Ethernet controllers with hardware-based failover are similar to the example shown in Figure 21. Since hardware-based failover schemes do not add additional overhead to the host processor, such an approach is very scalable for large networks.