- Ethernet Technologies -

What is Ethernet?

Ethernet is a family of technologies that provides data-link and physical specifications for controlling access to a shared network medium. It has emerged as the dominant technology used in LAN networking.

Ethernet was originally developed by Xerox in the 1970s, and operated at 2.94Mbps. The technology was standardized as **Ethernet Version 1** by a consortium of three companies - DEC, Intel, and Xerox, collectively referred to as **DIX -** and further refined as **Ethernet II** in 1982.

In the mid 1980s, the **Institute of Electrical and Electronic Engineers (IEEE)** published a formal standard for Ethernet, defined as the **IEEE 802.3** standard. The original 802.3 Ethernet operated at 10Mbps, and successfully supplanted competing LAN technologies, such as Token Ring.

Ethernet has several benefits over other LAN technologies:
- Simple to install and manage
- Inexpensive
- Flexible and scalable
- Easy to interoperate between vendors

(References: http://docwiki.cisco.com/wiki/Ethernet_Technologies; http://www.techfest.com/networking/lan/ethernet1.htm)

Ethernet Cabling Types

Ethernet can be deployed over three types of cabling:
- **Coaxial** cabling – *almost entirely deprecated in Ethernet networking*
- **Twisted-pair** cabling
- **Fiber optic** cabling

Coaxial cable, often abbreviated as *coax*, consists of a single wire surrounded by insulation, a metallic shield, and a plastic sheath. The shield helps protect against **electromagnetic interference (EMI),** which can cause **attenuation,** a reduction of the strength and quality of a signal. EMI can be generated by a variety of sources, such as florescent light ballasts, microwaves, cell phones, and radio transmitters.

Coax is commonly used to deploy cable television to homes and businesses.

Ethernet Cabling Types (continued)

Two types of coax were used historically in Ethernet networks:

- **Thinnet**
- **Thicknet**

Thicknet has a wider diameter and more shielding, which supports greater distances. However, it is less flexible than the smaller *thinnet*, and thus more difficult to work with. A **vampire tap** is used to physically connect devices to thicknet, while a **BNC** connector is used for thinnet.

Twisted-pair cable consists of two or four pairs of copper wires in a plastic sheath. Wires in a pair *twist* around each other to reduce **crosstalk**, a form of EMI that occurs when the signal from one wire *bleeds* or *interferes* with a signal on another wire. Twisted-pair is the most common Ethernet cable.

Twisted-pair cabling can be either **shielded** or **unshielded**. Shielded twisted-pair is more resistant to external EMI; however, all forms of twisted-pair suffer from greater signal attenuation than coax cable.

There are several *categories* of twisted-pair cable, identified by the number of *twists per inch* of the copper pairs:

- **Category 3** or **Cat3** - three twists per inch.
- **Cat5** - five twists per inch.
- **Cat5e** - five twists per inch; pairs are also twisted around each other.
- **Cat6 –** six twists per inch, with improved insulation.

An **RJ45** connector is used to connect a device to a twisted-pair cable. The *layout* of the wires in the connector dictates the function of the cable.

While coax and twisted-pair cabling carry *electronic* signals, **fiber optics** uses *light* to transmit a signal. Ethernet supports two fiber specifications:

- **Singlemode fiber** – consists of a very small glass *core*, allowing only a single ray or *mode* of light to travel across it. This greatly reduces the attenuation and dispersion of the light signal, supporting high bandwidth over *very* long distances, often measured in kilometers.
- **Multimode fiber** – consists of a larger core, allowing multiple modes of light to traverse it. Multimode suffers from greater dispersion than singlemode, resulting in shorter supported distances.

Singlemode fiber requires more *precise* electronics than multimode, and thus is significantly more *expensive*. Multimode fiber is often used for high-speed connectivity within a datacenter.

Network Topologies

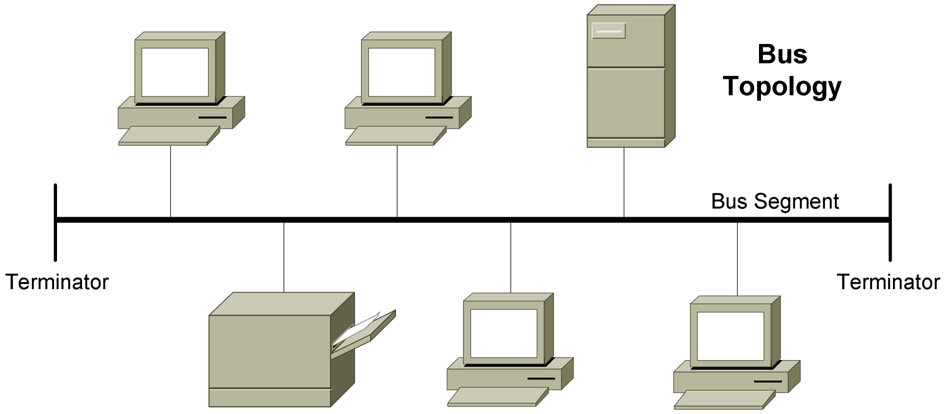
A **topology** defines both the *physical* and *logical* structure of a network. Topologies come in a variety of configurations, including:

- Bus
- Star
- Ring
- Full or partial mesh

Ethernet supports two topology types – **bus** and **star**.

Ethernet Bus Topology

In a **bus topology**, all hosts share a single physical segment (the *bus* or the *backbone*) to communicate:



A frame sent by one host is received by *all other* hosts on the bus. However, a host will only *process* a frame if it matches the destination hardware address in the data-link header.

Bus topologies are inexpensive to implement, but are almost entirely deprecated in Ethernet. There are several disadvantages to the bus topology:

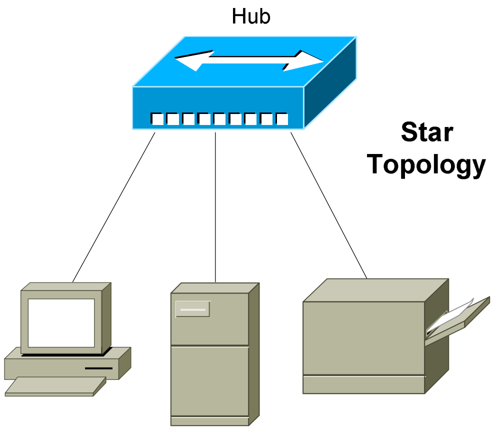
- Both ends of the bus must be **terminated,** otherwise a signal will *reflect* back and cause interference, severely degrading performance.
- Adding or removing hosts to the bus can be difficult.
- The bus represents a single point of failure - a break in the bus will affect *all* hosts on the segment. Such faults are often very difficult to troubleshoot.

A bus topology is implemented using either thinnet or thicknet coax cable.

* * *

Ethernet Star Topology

In a **star topology**, each host has an individual point-to-point connection to a centralized *hub* or *switch*:



A **hub** provides no intelligent forwarding whatsoever, and will always forward every frame out every port, excluding the port originating the frame. As with a bus topology, a host will only *process* a frame if it matches the destination hardware address in the data-link header. Otherwise, it will discard the frame.

A **switch** builds a **hardware address table,** allowing it to make intelligent forwarding decisions based on frame (data-link) headers. A frame can then be forwarded out *only* the appropriate destination port, instead of *all* ports.

Hubs and switches are covered in great detail in another guide.

Adding or removing hosts is very simple in a star topology. Also, a break in a cable will affect *only that one host*, and not the entire network.

There are two disadvantages to the star topology:
- The hub or switch represents a single point of failure.
- Equipment and cabling costs are generally higher than in a bus topology.

However, the star is still the dominant topology in modern Ethernet networks, due to its flexibility and scalability. Both twisted-pair and fiber cabling can be used in a star topology.

The Ethernet Frame

An Ethernet frame contains the following fields:

Field	***Length***	***Description***
Preamble	7 bytes	Synchronizes communication
Start of Frame	1 byte	Signals the start of a valid frame
MAC Destination	6 bytes	Destination MAC address
MAC Source	6 bytes	Source MAC address
802.1Q tag	4 bytes	Optional VLAN tag
Ethertype or length	2 bytes	Payload type or frame size
Payload	42-1500 bytes	Data payload
CRC	4 bytes	Frame error check
Interframe Gap	12 bytes	Required idle period between frames

The **preamble** is 56 bits of alternating 1s and 0s that synchronizes communication on an Ethernet network. It is followed by an 8-bit **start of frame delimiter** (10101011) that indicates a valid frame is about to begin. The preamble and the start of frame are *not considered* part of the actual frame, or calculated as part of the total frame size.

Ethernet uses the 48-bit **MAC address** for hardware addressing. The first 24-bits of a MAC address determine the manufacturer of the network interface, and the last 24-bits uniquely identify the host.

The *destination* MAC address identifies who is to receive the frame - this can be a single host (a *unicast*), a group of hosts (a *multicast*), or all hosts (a *broadcast*). The *source* MAC address indentifies the host originating the frame.

The **802.1Q tag** is an *optional* field used to identify which **VLAN** the frame belongs to. VLANs are covered in great detail in another guide.

The 16-bit **Ethertype/Length field** provides a different function depending on the standard - Ethernet II or 802.3. With Ethernet II, the field identifies the type of payload in the frame (the *Ethertype).* However, Ethernet II is almost entirely deprecated.

With 802.3, the field identifies the *length* of the payload. The length of a frame is important – there is both a *minimum* and *maximum* frame size.

(Reference: http://www.techfest.com/networking/lan/ethernet2.htm; http://www.dcs.gla.ac.uk/~lewis/networkpages/m04s03EthernetFrame.htm)

* * *

The Ethernet Frame (continued)

Field	***Length***	***Description***
Preamble	7 bytes	Synchronizes communication
Start of Frame	1 byte	Signals the start of a valid frame
MAC Destination	6 bytes	Destination MAC address
MAC Source	6 bytes	Source MAC address
802.1Q tag	4 bytes	Optional VLAN tag
Ethertype or length	2 bytes	Payload type or frame size
Payload	42-1500 bytes	Data payload
CRC	4 bytes	Frame error check
Interframe Gap	12 bytes	Required idle period between frames

The absolute **minimum frame size** for Ethernet is **64 bytes** (or **512 bits**) including headers. A frame that is smaller than 64 bytes will be discarded as a **runt**. The required fields in an Ethernet header add up to 18 bytes – thus, the frame **payload** must be a minimum of 46 bytes, to equal the minimum 64-byte frame size. If the payload *does not* meet this minimum, the payload is **padded** with **0 bits** until the minimum is met.

Note: If the optional 4-byte 802.1Q tag is used, the Ethernet header size will total 22 bytes, requiring a minimum payload of 42 bytes.

By default, the **maximum frame size** for Ethernet is **1518 bytes –** *18 bytes* of header fields, and *1500 bytes* of payload - or **1522 bytes** with the 802.1Q tag. A frame that is larger than the maximum will be discarded as a **giant.** With both runts and giants, the receiving host will *not* notify the sender that the frame was dropped. Ethernet relies on higher-layer protocols, such as TCP, to provide retransmission of discarded frames.

Some Ethernet devices support **jumbo frames** of **9216 bytes**, which provide less overhead due to fewer frames. Jumbo frames must be explicitly enabled on *all* devices in the traffic path to prevent the frames from being dropped.

The 32-bit **Cycle Redundancy Check** (**CRC**) field is used for error-detection. A frame with an invalid CRC will be discarded by the receiving device. This field is a *trailer,* and not a *header*, as it follows the payload.

The 96-bit **Interframe Gap** is a required idle period between frame transmissions, allowing hosts time to prepare for the next frame.

(Reference: http://www.infocellar.com/networks/ethernet/frame.htm)

CSMA/CD and Half-Duplex Communication

Ethernet was originally developed to support a **shared media** environment. This allowed two or more hosts to use the same physical network medium.

There are two methods of communication on a shared physical medium:
- **Half-Duplex –** hosts can transmit or receive, but *not simultaneously*
- **Full-Duplex –** hosts can both transmit and receive simultaneously

On a half-duplex connection, Ethernet utilizes **Carrier Sense Multiple Access with Collision Detect (CSMA/CD)** to control media access. *Carrier sense* specifies that a host will monitor the physical link, to determine whether a *carrier* (or *signal)* is currently being transmitted. The host will *only* transmit a frame if the link is **idle**, and the Interframe Gap has expired.

If two hosts transmit a frame simultaneously, a **collision** will occur. This renders the collided frames unreadable. Once a collision is detected, both hosts will send a **32-bit jam sequence** to ensure all transmitting hosts are aware of the collision. The collided frames are also discarded.

Both devices will then wait a *random* amount of time before resending their respective frames, to reduce the likelihood of another collision. This is controlled by a **backoff** timer process.

Hosts *must* detect a collision before a frame is finished transmitting, otherwise CSMA/CD cannot function reliably. This is accomplished using a consistent **slot time,** the time required to send a specific amount of data from one end of the network and then *back*, measured in bits.

A host must continue to transmit a frame for a *minimum* of the slot time. In a properly configured environment, a collision should *always* occur within this slot time, as enough time has elapsed for the frame to have reached the far end of the network and back, and thus all devices should be aware of the transmission. The slot time effectively limits the physical length of the network – if a network segment is too long, a host may not detect a collision within the slot time period. A collision that occurs after the slot time is referred to as a **late collision.**

For 10 and 100Mbps Ethernet, the slot time was defined as **512 bits,** or 64 bytes. Note that this is the equivalent of the *minimum Ethernet frame size* of 64 bytes. The slot time actually defines this minimum. For Gigabit Ethernet, the slot time was defined as **4096 bits.**

(Reference: *http://www.techfest.com/networking/lan/ethernet3.htm*)

* * *

Full-Duplex Communication

Unlike half-duplex, **full-duplex** Ethernet supports simultaneously communication by providing separate transmit and receive paths. This effectively *doubles* the throughput of a network interface.

Full-duplex Ethernet was formalized in IEEE 802.3x, and *does not use CSMA/CD* or slot times. Collisions should *never* occur on a functional full-duplex link. Greater distances are supported when using full-duplex over half-duplex.

Full-duplex is only supported on a point-to-point connection between two devices. Thus, a bus topology using coax cable *does not* support full-duplex.

Only a connection **between two hosts** or between **a host and a *switch*** supports full-duplex. A host connected to a *hub* is limited to half-duplex. Both hubs and half-duplex communication are mostly deprecated in modern networks.

Categories of Ethernet

The original 802.3 Ethernet standard has evolved over time, supporting faster transmission rates, longer distances, and newer hardware technologies. These *revisions* or *amendments* are identified by the letter appended to the standard, such as 802.3*u* or 802.3*z*.

Major categories of Ethernet have also been organized by their speed:

- **Ethernet** (10Mbps)
- **Fast Ethernet** (100Mbps)
- **Gigabit Ethernet**
- **10 Gigabit Ethernet**

The *physical* standards for Ethernet are often labeled by their transmission rate, signaling type, and media type. For example, *100baseT* represents the following:

- The first part (*100*) represents the transmission rate, in Mbps.
- The second part (*base)* indicates that it is a baseband transmission.
- The last part (*T)* represents the physical media type (*twisted-pair).*

Ethernet communication is **baseband**, which dedicates the entire capacity of the medium to one signal or channel. In **broadband**, multiple signals or channels can share the same link, through the use of *modulation* (usually frequency modulation).

Ethernet (10 Mbps)

Ethernet is now a somewhat generic term, describing the entire family of technologies. However, Ethernet traditionally referred to the original 802.3 standard, which operated at **10 Mbps**. Ethernet supports coax, twisted-pair, and fiber cabling. Ethernet over twisted-pair uses **two** of the four pairs.

Common Ethernet physical standards include:

IEEE Standard	***Physical Standard***	***Cable Type***	***Maximum Speed***	***Maximum Cable Length***
802.3a	10base2	Coaxial (thinnet)	10 Mbps	185 meters
802.3	10base5	Coaxial (thicknet)	10 Mbps	500 meters
802.3i	10baseT	Twisted-pair	10 Mbps	100 meters
802.3j	10baseF	Fiber	10 Mbps	2000 meters

Both 10baseT and 10baseF support full-duplex operation, effectively doubling the bandwidth to 20 Mbps. Remember, only a connection **between two hosts** or between **a host and a *switch*** support full-duplex. The maximum distance of an Ethernet segment can be extended through the use of a **repeater**. A hub or a switch can also serve as a repeater.

Fast Ethernet (100 Mbps)

In 1995, the IEEE formalized **802.3u**, a **100 Mbps** revision of Ethernet that became known as **Fast Ethernet.** Fast Ethernet supports both twisted-pair copper and fiber cabling, and supports both half-duplex and full-duplex.

Common Fast Ethernet physical standards include:

IEEE Standard	***Physical Standard***	***Cable Type***	***Maximum Speed***	***Maximum Cable Length***
802.3u	100baseTX	Twisted-pair	100 Mbps	100 meters
802.3u	100baseT4	Twisted-pair	100 Mbps	100 meters
802.3u	100baseFX	Multimode fiber	100 Mbps	400-2000 meters
802.3u	100baseSX	Multimode fiber	100 Mbps	500 meters

100baseT4 was never widely implemented, and only supported half-duplex operation. 100baseTX is the dominant Fast Ethernet physical standard. 100baseTX uses **two** of the four pairs in a twisted-pair cable, and requires Category 5 cable for reliable performance.

Speed and Duplex Autonegotiation

Fast Ethernet is backwards-compatible with the original Ethernet standard. A device that supports both Ethernet and Fast Ethernet is often referred to as a *10/100* device.

Fast Ethernet also introduced the ability to **autonegotiate** both the speed and duplex of an interface. Autonegotiation will attempt to use the *fastest* speed available, and will attempt to use *full-duplex* if both devices support it. Speed and duplex can also be **hardcoded**, preventing negotiation.

The configuration *must* be consistent on both sides of the connection. Either both sides must be configured to autonegotiate, or both sides must be hardcoded with *identical* settings. Otherwise a **duplex mismatch** error can occur.

For example, if a workstation's NIC is configured to autonegotiate, and the switch interface is hardcoded for 100Mbps and full-duplex, then a duplex mismatch will occur. The workstation's NIC will sense the correct speed of 100Mbps, but will not detect the correct duplex and will default to *half-duplex.*

If the duplex is mismatched, collisions will occur. Because the full-duplex side of the connection does not utilize CSMA/CD, performance is *severely* degraded. These issues can be difficult to troubleshoot, as the network connection will still function, but will be excruciatingly slow.

When autonegotiation was first developed, manufacturers did not always adhere to the same standard. This resulted in frequent mismatch issues, and a sentiment of distrust towards autonegotiation.

Though modern network hardware has alleviated most of the incompatibility, many administrators are still skeptical of autonegotiation and choose to hardcode all connections. Another common practice is to hardcode server and datacenter connections, but to allow user devices to autonegotiate.

Gigabit Ethernet, covered in the next section, provided several enhancements to autonegotiation, such as hardware flow control. Most manufacturers **recommend autonegotiation** on Gigabit Ethernet interfaces as a best practice.

Gigabit Ethernet

Gigabit Ethernet operates at 1000 Mbps, and supports both twisted-pair **(802.3ab)** and fiber cabling **(802.3z)**. Gigabit over twisted-pair uses **all four pairs**, and requires Category 5e cable for reliable performance.

Gigabit Ethernet is backwards-compatible with the original Ethernet and Fast Ethernet. A device that supports all three is often referred to as a *10/100/1000* device. Gigabit Ethernet supports both half-duplex or full-duplex operation. Full-duplex Gigabit Ethernet effectively provides 2000 Mbps of throughput.

Common Gigabit Ethernet physical standards include:

IEEE Standard	***Physical Standard***	***Cable Type***	***Speed***	***Maximum Cable Length***
802.3ab	1000baseT	Twisted-pair	1 Gbps	100 meters
802.3z	1000baseSX	Multimode fiber	1 Gbps	500 meters
802.3z	1000baseLX	Multimode fiber	1 Gbps	500 meters
802.3z	1000baseLX	Singlemode fiber	1 Gbps	Several kilometers

In modern network equipment, Gigabit Ethernet has replaced both Ethernet and Fast Ethernet.

10 Gigabit Ethernet

10 Gigabit Ethernet operates at 10000 Mbps, and supports both twisted-pair **(802.3an)** and fiber cabling **(802.3ae)**. 10 Gigabit over twisted-pair uses **all four pairs**, and requires Category 6 cable for reliable performance.

Common Gigabit Ethernet physical standards include:

IEEE Standard	***Physical Standard***	***Cable Type***	***Speed***	***Maximum Cable Length***
802.3an	10Gbase-T	Twisted-pair	10 Gbps	100 meters
802.3ae	10Gbase-SR	Multimode fiber	10 Gbps	300 meters
802.3ae	10Gbase-LR	Singlemode fiber	10 Gbps	Several kilometers

10 Gigabit Ethernet is usually used for high-speed connectivity within a datacenter, and is predominantly deployed over fiber.

Twisted-Pair Cabling Overview

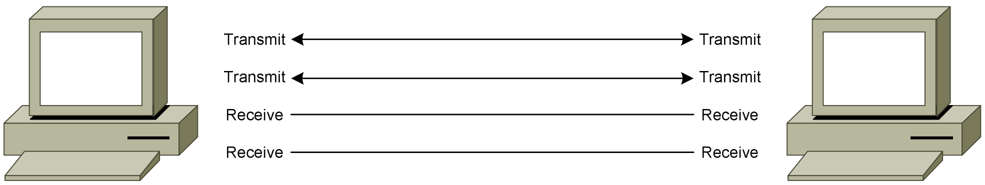
A typical twisted-pair cable consists of **four pairs** of copper wires, for a total of **eight wires**. Each side of the cable is terminated using an RJ45 connector, which has eight **pins.** When the connector is *crimped* onto the cable, these pins make contact with each wire.

The wires themselves are assigned a *color* to distinguish them. The color is dictated by the cabling standard - **TIA/EIA-568B** is the current standard:

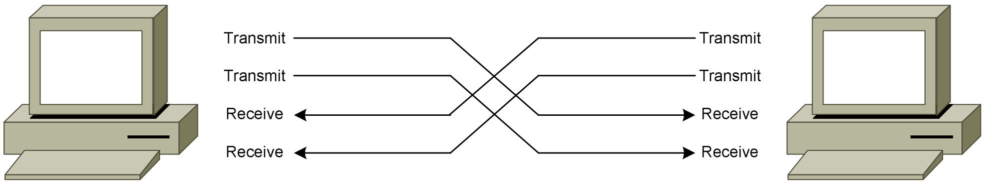
Color	***Pin#***
White Orange	1
Orange	2
White Green	3
Blue	4
White Blue	5
Green	6
White Brown	7
Brown	8

Each wire is assigned a specific purpose. For example, both Ethernet and Fast Ethernet use two wires to transmit, and two wires to receive data, while the other four pins remain unused.

For communication to occur, *transmit* pins must connect to the *receive* pins of the remote host. This does not occur in a **straight-through** configuration:



The pins must be **crossed-over** for communication to be successful:



The *crossover* can be controlled either by the cable, or an intermediary device, such as a hub or switch.

Twisted-Pair Cabling – Cable and Interface Types

The *layout* or *pinout* of the wires in the RJ45 connector dictates the **function** of the cable. There are three common types of twisted-pair cable:

- **Straight-through** cable
- **Crossover** cable
- **Rollover** cable

The network *interface* type determines when to use each cable:

- **Medium Dependent Interface (MDI)**
- **Medium Dependent Interface with Crossover (MDIX)**

Host interfaces are generally MDI, while hub or switch interfaces are typically MDIX.

Twisted-Pair Cabling – Straight-Through Cable

A **straight-through** cable is used in the following circumstances:

- From a host to a hub – *MDI to MDIX*
- From a host to a switch - *MDI to MDIX*
- From a router to a hub - *MDI to MDIX*
- From a router to a switch - *MDI to MDIX*

Essentially, a straight-through cable is used to connect *any device* to a hub or switch, *except* for another hub or switch. The hub or switch provides the *crossover* (or *MDIX*) function to connect transmit pins to receive pins.

The pinout on each end of a straight-through cable **must be identical**. The TIA/EIA-568B standard for a straight-through cable is as follows:

Pin#	***Connector 1***		***Connector 2***	***Pin#***
1	White Orange	------------------------	White Orange	1
2	Orange	------------------------	Orange	2
3	White Green	------------------------	White Green	3
4	Blue	------------------------	Blue	4
5	White Blue	------------------------	White Blue	5
6	Green	------------------------	Green	6
7	White Brown	------------------------	White Brown	7
8	Brown	------------------------	Brown	8

A straight-through cable is often referred to as a **patch cable.**

Twisted-Pair Cabling – Crossover Cable

A **crossover** cable is used in the following circumstances:

- From a host to a host – *MDI to MDI*
- From a hub to a hub - *MDIX to MDIX*
- From a switch to a switch - *MDIX to MDIX*
- From a hub to a switch - *MDIX to MDIX*
- From a router to a router - *MDI to MDI*

Remember that a hub or a switch will provide the crossover function. However, when connecting a host directly to another host (MDI to MDI), the crossover function must be provided by a crossover cable.

A crossover cable is often required to uplink a hub to another hub, or to uplink a switch to another switch. This is because the crossover is performed *twice,* once on each hub or switch (MDIX to MDIX), negating the crossover.

Modern devices can now **automatically detect** whether the crossover function is required, negating the need for a crossover cable. This functionality is referred to as **Auto-MDIX**, and is now standard with Gigabit Ethernet, which uses all eight wires to both transmit *and* receive. Auto-MDIX requires that autonegotiation be enabled.

To create a crossover cable, the transmit pins must be swapped with the receive pins on **one** end of the cable:

- Pins 1 and 3
- Pins 2 and 6

Pin#	***Connector 1***		***Connector 2***	***Pin#***
1	White Orange	------------------------	White Green	3
2	Orange	------------------------	Green	6
3	White Green	------------------------	White Orange	1
4	Blue	------------------------	Blue	4
5	White Blue	------------------------	White Blue	5
6	Green	------------------------	Orange	2
7	White Brown	------------------------	White Brown	7
8	Brown	------------------------	Brown	8

Note that the Orange and Green pins have been swapped on Connector 2. The first connector is using the TIA/EIA-568B standard, while the second connector is using the TIA/EIA-568A standard.

Twisted-Pair – Rollover Cable

A **rollover** cable is used to connect a workstation or laptop into a Cisco device's **console** or **auxiliary** port, for management purposes. A rollover cable is often referred to as a *console* cable, and its sheathing is usually flat and light-blue in color.

To create a rollover cable, the pins are completely reversed on one end of the cable:

Pin#	***Connector 1***		***Connector 2***	***Pin#***
1	White Orange	------------------------	Brown	8
2	Orange	------------------------	White Brown	7
3	White Green	------------------------	Green	6
4	Blue	------------------------	White Blue	5
5	White Blue	------------------------	Blue	4
6	Green	------------------------	White Green	3
7	White Brown	------------------------	Orange	2
8	Brown	------------------------	White Orange	1

Rollover cables can be used to configure Cisco routers, switches, and firewalls.

Power over Ethernet (PoE)

Power over Ethernet (PoE) allows both data and power to be sent across the same twisted-pair cable, eliminating the need to provide separate power connections. This is especially useful in areas where installing separate power might be expensive or difficult.

PoE can be used to power many devices, including:
- Voice over IP (VoIP) phones
- Security cameras
- Wireless access points
- Thin clients

PoE was originally formalized as **802.3af**, which can provide roughly 13W of power to a device. **802.3at** further enhanced PoE, supporting 25W or more power to a device.

Ethernet, Fast Ethernet, *and* Gigabit Ethernet all support PoE. Power can be sent across either the *unused* pairs in a cable, or the data transmission pairs, which is referred to as **phantom power.** Gigabit Ethernet requires the phantom power method, as it uses all eight wires in a twisted-pair cable.

The device that *provides* power is referred to as the **Power Source Equipment (PSE).** PoE can be supplied using an **external power injector**, though each powered device requires a separate power injector.

More commonly, an **802.3af-compliant network switch** is used to provide power to many devices simultaneously. The power supplies in the switch must be large enough to support both the switch itself, and the devices it is powering.

(Reference: *http://www.belden.com/docs/upload/PoE_Basics_WP.pdf*)