

Network Basics

[Dashboard](#) / [Courses](#) / [Miscellaneous](#) / [Network Basics](#) / [IP Routing](#) / [Routing Basics](#)

Routing Basics

Routing Basics

Once you create an internetwork by connecting your wide area networks (WANs) and local area networks (LANs) to a **router**, you need to configure logical network addresses, such as IP addresses, to all hosts on the internetwork so that they can communicate via routers across that internetwork.

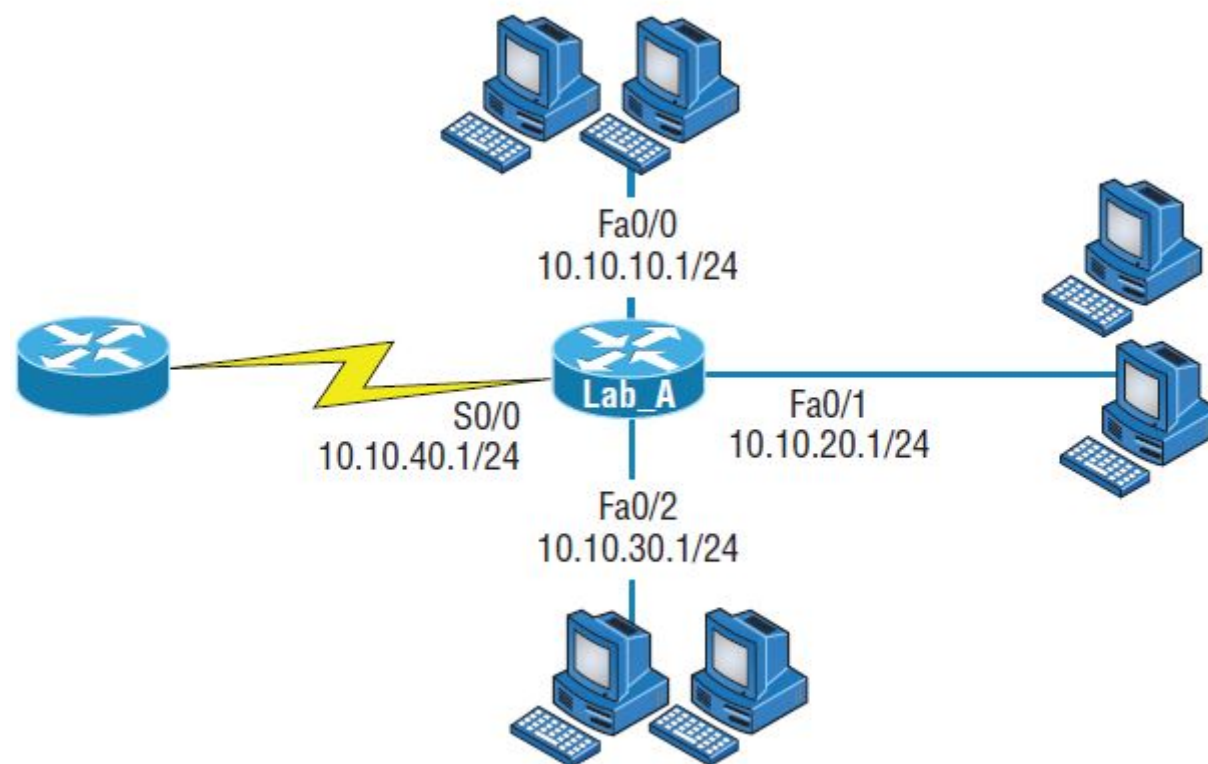
In IT, routing essentially refers to the process of taking a packet from one device and sending it through the network to another device on a different network. Routers don't really care about hosts—they care only about networks and the best path to each network. The logical network address of the destination host is used to get packets to a network through a routed network, and then the hardware address of the host is used to deliver the packet from a router to the correct destination host.

If your network has no routers, then it should be apparent that, well, you are not routing. But if you do have them, they're there to route traffic to all the networks in your internetwork. To be capable of routing packets, a router must know at least the following information:

- Destination network address
- Neighbor routers from which it can learn about remote networks
- Possible routes to all remote networks
- The best route to each remote network
- How to maintain and verify routing information

The router learns about remote networks from neighbor routers or from an administrator. The router then builds a **routing table** (a map of the internetwork) that describes how to find the remote networks. If a network is directly connected, then the router already knows how to get to it.

If a network isn't directly connected to the router, the router must use one of two ways to learn how to get to it. One way is called **static routing**, which can be a ton of work because it requires someone to hand-type all network locations into the routing table. The other way is **dynamic routing**. In dynamic routing, a protocol on one router communicates with the same protocol running on neighbor routers. The routers then update each other about all the networks they know about and place this information into the routing table. If a change occurs in the network, the dynamic routing protocols automatically inform all routers about the event. If static routing is used, the *administrator* is responsible for updating all changes by hand into all routers.



Routing example

The above figure shows a simple two-router network. Lab_A has one serial interface and three LAN interfaces. Can you figure out which interface Lab_A will use to forward an IP datagram to a host with an IP address of **10.10.10.10**? By using the Cisco IOS command `show ip route`, we can see the routing table (map of the internetwork) that router Lab_A will use to make all forwarding decisions:

```
Router_A#show ip route
[output cut]
Gateway of last resort is not set
C 10.10.10.0/24 is directly connected, FastEthernet0/0
C 10.10.20.0/24 is directly connected, FastEthernet0/1
C 10.10.30.0/24 is directly connected, FastEthernet0/2
C 10.10.40.0/24 is directly connected, Serial 0/0
```

The **c** in the routing table output means that the networks listed are “directly connected,” and until we add a routing protocol—something like *RIP*, *EIGRP*, and so on—to the routers in our internetwork, or use static routes, we’ll have only directly connected networks in our routing table.

By looking at the figure and the output of the routing table, can you tell what Lab_A will do with a received packet that has a destination IP address of **10.10.10.10**? If you answered, “The router will packet-switch the packet to interface FastEthernet 0/0, and this interface will then frame the packet and send it out on the network segment,” you’re right.

Tip:

- When the routing tables of all routers in the network are complete (because they include information about all the networks in the internetwork), they are considered converged or in a steady state.

You have completed 0% of the lesson

0%

Jump to...



Network Basics

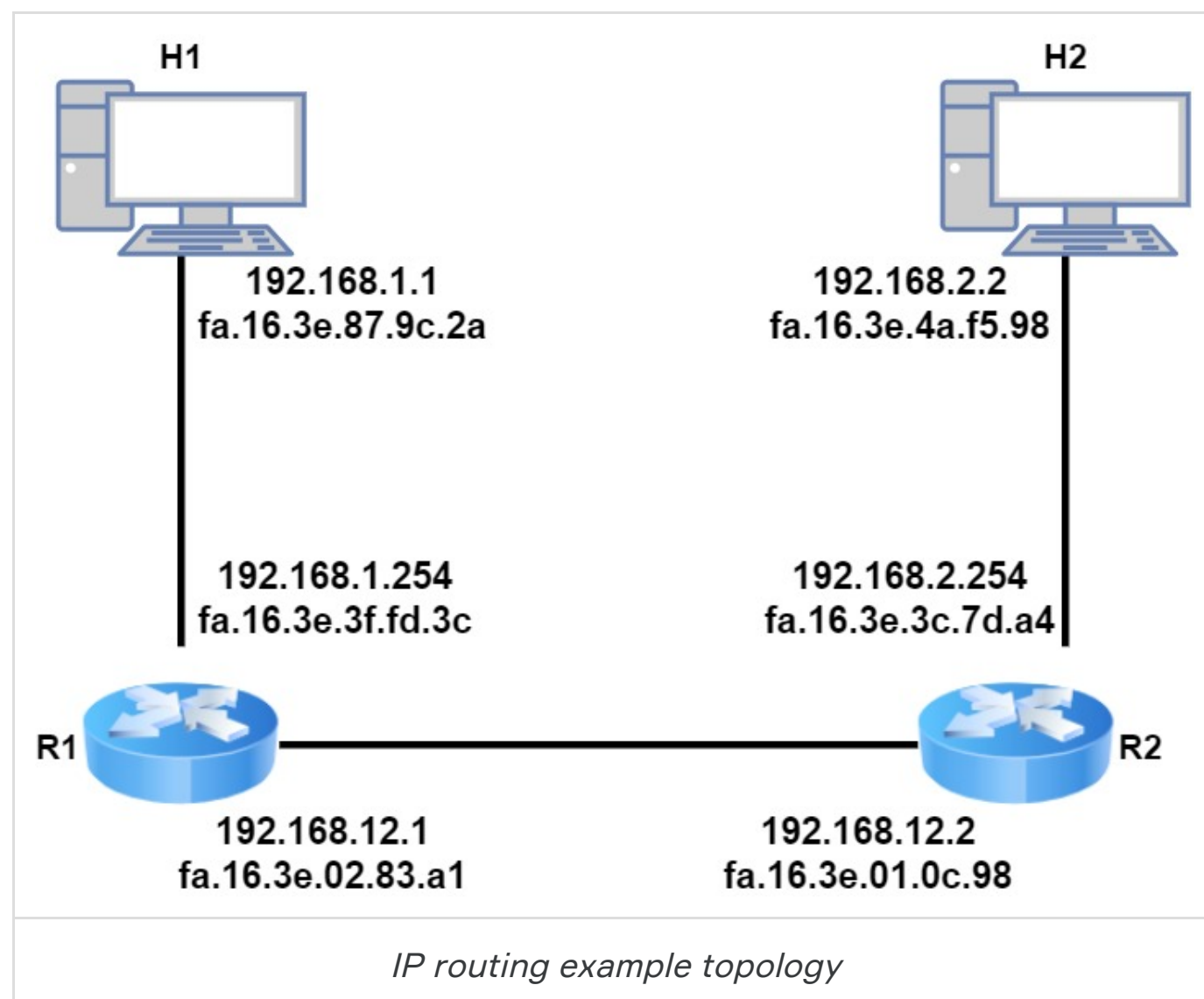
[Dashboard](#) / [Courses](#) / [Miscellaneous](#) / [Network Basics](#) / [IP Routing](#) / [The IP Routing Process](#)

The IP Routing Process

The IP Routing Process

The actual forwarding of IP packets by routers is called **IP routing**. This has nothing to do with the “learning” of network routes through static or dynamic routing protocols but has everything to do with the steps that routers have to take when they forward an IP packet from one interface to another.

In this lesson, we will walk through an example and show all the steps that occur. To do this, we will use the following topology:



Above we have two host computers and two routers. H1 is going to send an IP packet to H2 which has to be routed by R1 and R2.

H1

Let's start with **H1**. This host creates an IP packet with its own IP address (**192.168.1.1**) as the source and **H2** (**192.168.2.2**) as the destination. The first question that **H1** will ask itself is:

- *Is the destination local or remote?*

It answers this question by looking at its own IP address, its subnet mask, and the destination IP address:

```
C:\Users\H1>ipconfig
Windows IP Configuration
Ethernet adapter Ethernet 1:
    Connection-specific DNS Suffix  . : nwl.local
    Link-local IPv6 Address . . . . . : fe80::88fd:962a:44d6:3a1f%4
    IPv4 Address. . . . . : 192.168.1.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.254
```

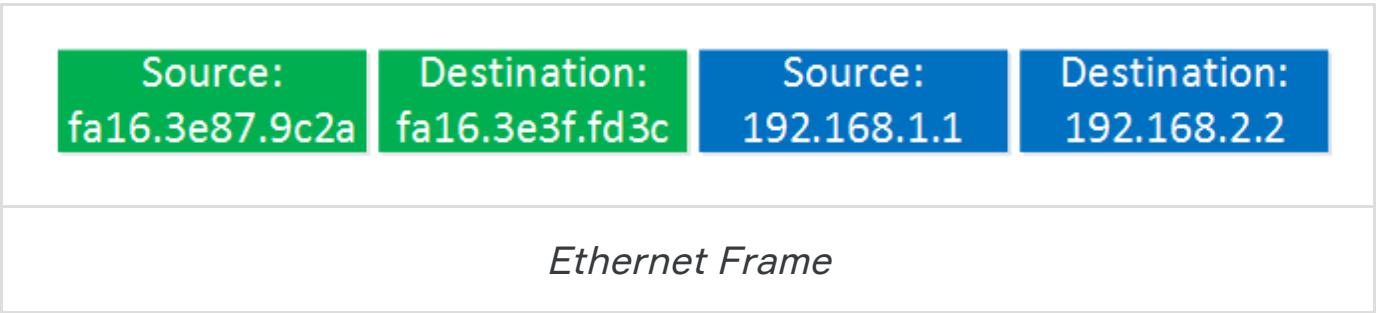
H1 is in the network **192.168.1.0/24** so all IP addresses in the **192.168.1.1 - 254** range are local. Our destination (**192.168.2.2**) is outside of the local subnet so that means we have to use the default gateway.

H1 will now build an *Ethernet frame*, enters its own *source MAC address* and asks itself the second question, *do I know the destination MAC address of the default gateway?*

It checks its ARP (Address Resolution Table) table to find the answer:

```
C:\Users\H1>arp -a
Interface: 192.168.1.1 --- 0x4
    Internet Address      Physical Address      Type
    192.168.1.254         fa-16-3e-3f-fd-3c    dynamic
    192.168.1.255         ff-ff-ff-ff-ff-ff    static
    224.0.0.22            01-00-5e-00-00-16    static
    224.0.0.251          01-00-5e-00-00-fb    static
    224.0.0.252          01-00-5e-00-00-fc    static
    239.255.255.250      01-00-5e-7f-ff-fa    static
```

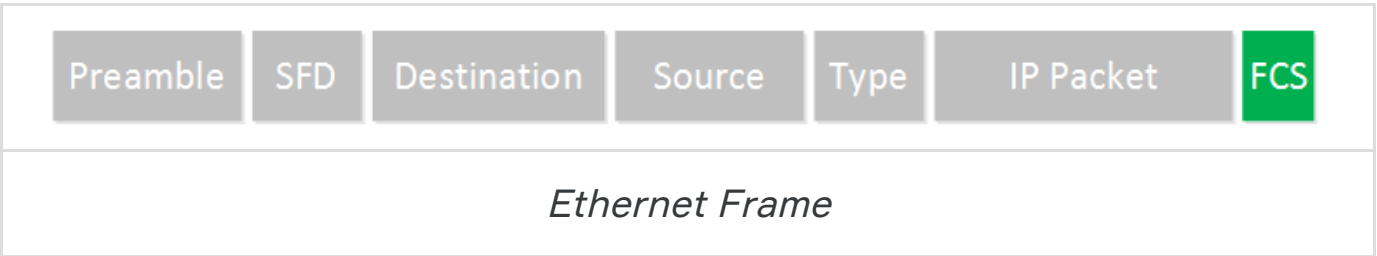
H1 has an ARP entry for **192.168.1.254**. If not, it would have sent an ARP request. We now have an Ethernet frame that carries an IP packet with the following addresses:



The frame will be on its way to **R1**.

R1

This Ethernet frame makes it to **R1** which has more work to do than our host. The first thing it does is check if the FCS (Frame Check Sequence) of the Ethernet frame is correct or not:

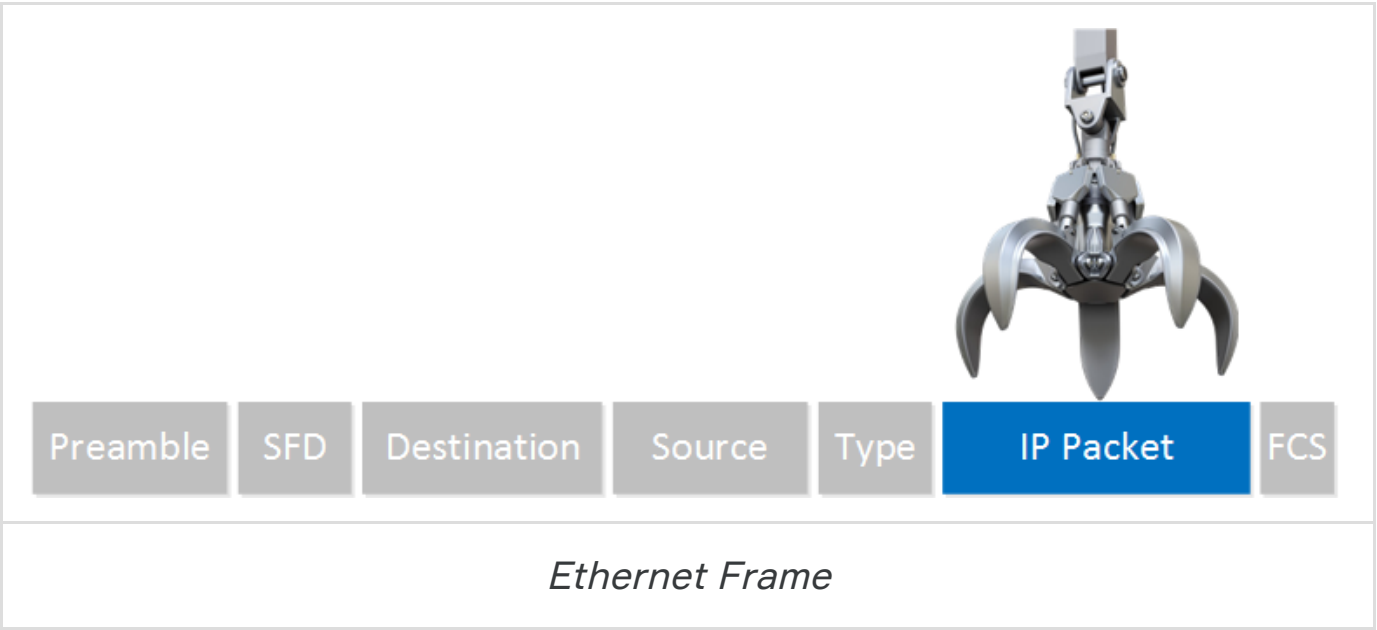


If the FCS is incorrect, the frame is dropped right away. There is no error recovery for Ethernet, this is something that is done by protocols on upper layers, like TCP on the transport layer.

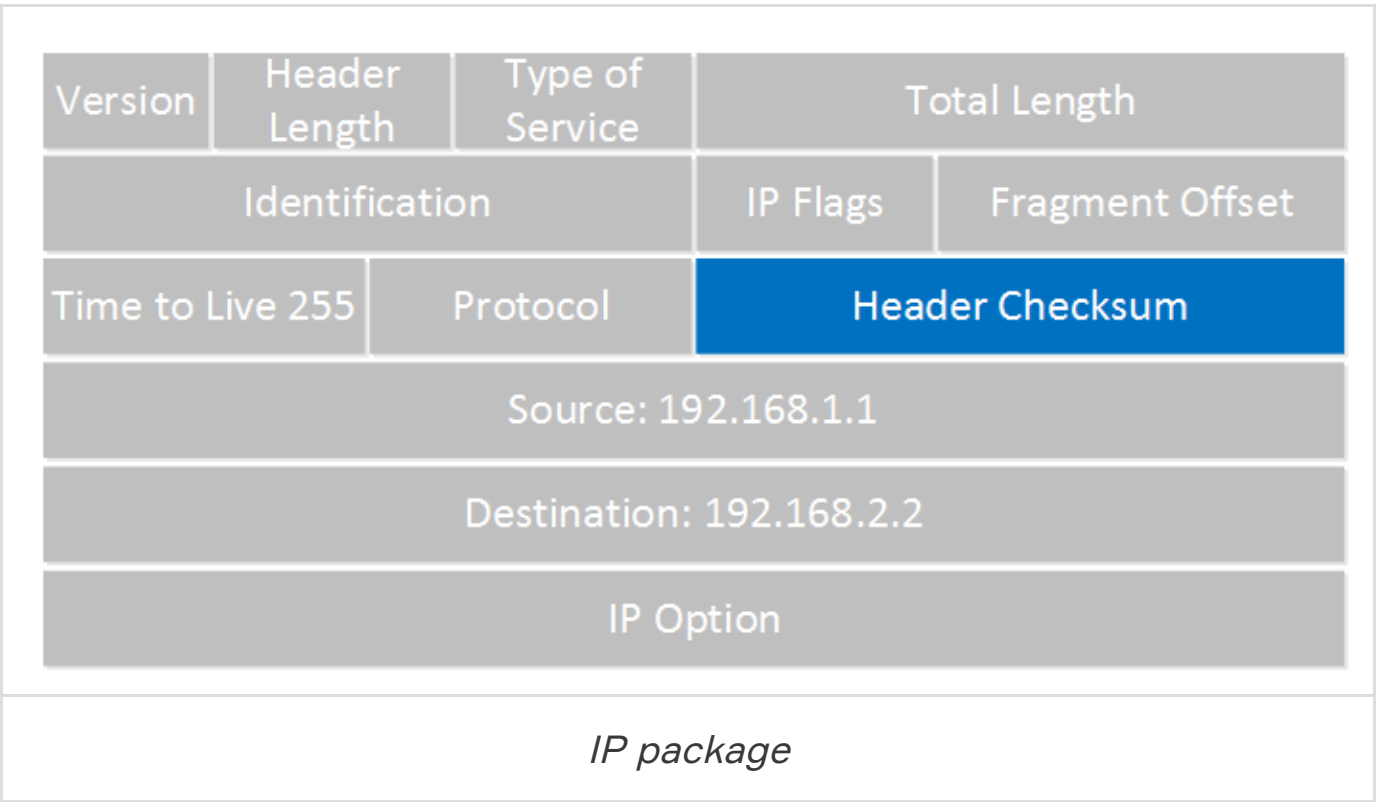
If the FCS is correct, we will process the frame if the destination MAC address is:

- the address of the interface of the router
- is a broadcast address of the subnet that the router interface is connected to
- is a multicast address that the router listens to.

In this case, the destination MAC address matches the MAC address of **R1**'s GigabitEthernet 0/1 interface so we will process it. We de-encapsulate (extract) the IP packet out of the Ethernet frame which is then discarded:



The router will now look at the IP packet, and the first thing it does is check if the header checksum is OK:



If the header checksum is not correct, the IP packet is dropped right away. There is also no error recovery on the network layer, we rely on upper layers for this. If the header checksum is correct, we continue by looking at the destination IP address:

Version	Header Length	Type of Service	Total Length	
Identification			IP Flags	Fragment Offset
Time to Live 255	Protocol		Header Checksum	
Source: 192.168.1.1				
Destination: 192.168.2.2				
IP Option				
IP package				

R1 now checks its routing table to see if there is a match:

R1#show ip route	
Gateway of last resort is not set	
	192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C	192.168.1.0/24 is directly connected, GigabitEthernet0/1
L	192.168.1.254/32 is directly connected, GigabitEthernet0/1
S	192.168.2.0/24 [1/0] via 192.168.12.2
	192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C	192.168.12.0/24 is directly connected, GigabitEthernet0/2
L	192.168.12.1/32 is directly connected, GigabitEthernet0/2

Above you can see that **R1** knows how to reach the 192.168.2.0/24 network, the next-hop IP address is 192.168.12.2. It will now do a second routing table lookup to see if it knows how to reach 192.168.12.2, we call this recursive routing. As you can see, there is an entry for 192.168.12.0/24 with GigabitEthernet 0/2 as the interface to use.

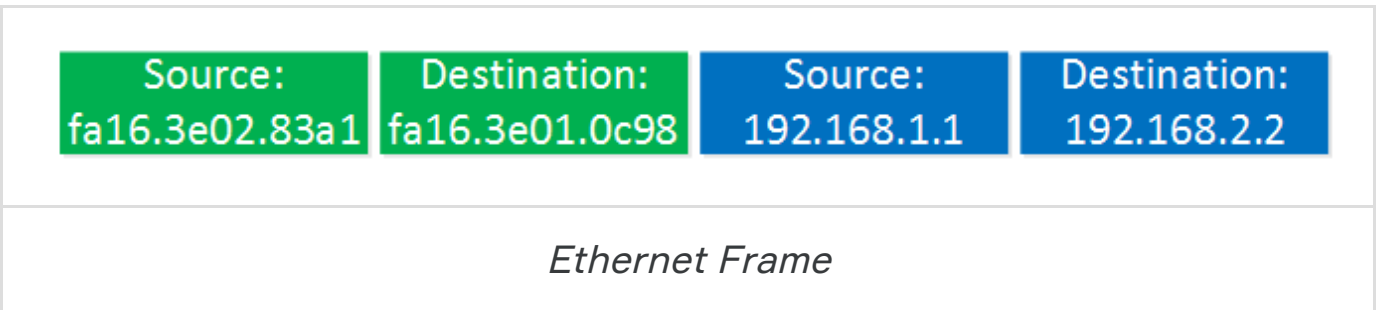
There is one thing left to do with the IP packet before we can forward it. Since we are routing it, we have to decrease the TTL (Time to Live) field by one. **R1** will do this and since this changes the IP header, we have to calculate a new header checksum.

Version	Header Length	Type of Service	Total Length	
Identification			IP Flags	Fragment Offset
Time to Live 254	Protocol		Header Checksum	
Source: 192.168.1.1				
Destination: 192.168.2.2				
IP Option				
IP package				

Once this is done, **R1** checks its ARP table to see if there is an entry for **192.168.12.2**:

```
R1#show ip arp
Protocol Address      Age (min)  Hardware Addr  Type   Interface
Internet 192.168.1.1      58        fa16.3e87.9c2a  ARPA   GigabitEthernet0/1
Internet 192.168.1.254    -         fa16.3e3f.fd3c  ARPA   GigabitEthernet0/1
Internet 192.168.12.1     -         fa16.3e02.83a1  ARPA   GigabitEthernet0/2
Internet 192.168.12.2     95        fa16.3e01.0c98  ARPA   GigabitEthernet0/2
```

No problem there, we have an entry in the ARP table. If not, **R1** will send an ARP request to find the MAC address of **192.168.12.2**. **R1** builds a new Ethernet frame with its own MAC address of the GigabitEthernet 0/2 interface and R2 as the destination. The IP packet is then encapsulated in this new Ethernet frame.



And the frame will be on its way towards **R2**.

R2

This Ethernet frame makes it to **R2**. Like **R1** it will first do this:

- Check the FCS of the Ethernet frame.
- De-encapsulates the IP packet, discard the frame.
- Check the IP header checksum.
- Check the destination IP address.

In the routing table, we find this:

```
R2#show ip route

Gateway of last resort is not set

S    192.168.1.0/24 [1/0] via 192.168.12.1
     192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C     192.168.2.0/24 is directly connected, GigabitEthernet0/1
L     192.168.2.254/32 is directly connected, GigabitEthernet0/1
     192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C     192.168.12.0/24 is directly connected, GigabitEthernet0/2
L     192.168.12.2/32 is directly connected, GigabitEthernet0/2
```

Network **192.168.2.0/24** is directly connected to **R2** on its GigabitEthernet 0/1 interface. **R2** will now reduce the TTL of the IP packet from 254 to 253, recalculate the IP header checksum and checks its ARP table to see if it knows how to reach **192.168.2.2**:


```
R2#show ip arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	192.168.2.2	121	fa16.3e4a.f598	ARPA	GigabitEthernet0/1
Internet	192.168.2.254	-	fa16.3e3c.7da4	ARPA	GigabitEthernet0/1
Internet	192.168.12.1	111	fa16.3e02.83a1	ARPA	GigabitEthernet0/2
Internet	192.168.12.2	-	fa16.3e01.0c98	ARPA	GigabitEthernet0/2

There is an ARP entry there. The new Ethernet frame is created, the IP packet encapsulated and it has the following addresses:

Source: fa16.3e3c.7da4	Destination: fa16.3e4a.f598	Source: 192.168.1.1	Destination: 192.168.2.2
Ethernet Frame			

The frame is then forwarded to H2.

H2

H2 receives the Ethernet frame and will:

- Check the FCS
- Find its own MAC address as the destination MAC address.
- De-encapsulates the IP packet from the frame.
- Finds its own IP address as the destination in the IP packet.

H2 then looks for the protocol field to figure out what transport layer protocol we are dealing with, what happens next depends on the *transport layer protocol* that is used.



Previous

Next

You have completed 0% of the lesson

0%

Jump to...

Network Basics

[Dashboard](#) / [Courses](#) / [Miscellaneous](#) / [Network Basics](#) / [IP Routing](#)
/ [Static and Dynamic Routing](#)

Static and Dynamic Routing

Static and Dynamic Routing

How does a router send packets to remote networks when the only way it can send them is by looking at the routing table to find out how to get to the remote networks? And what happens when a router receives a packet for a network that isn't listed in the routing table? It doesn't send a broadcast looking for the remote network—the router just discards the packet.

There are several ways to configure the routing tables to include all the networks so that packets will be forwarded. Understand that what's best for one network isn't necessarily what's best for another. Knowing about and being able to recognize the different types of routing will really help you come up with the best solution for your specific environment and business requirements.

We can configure a router with either **static** or **dynamic routing**. If we choose **static routing**, then we have to go to each router and type in each network and the path that IP will use to send packets. However, **static routing** does not scale well in large networks, but **dynamic routing** does because network routes are automatically added to the routing table via the routing protocol.

Dynamic routing protocols break up into many different categories or types of protocols. The first split in the dynamic protocol branch is the division of **interior gateway protocols (IGPs)** and **exterior gateway protocols (EGPs)** (The difference between IGP and EGP is interior or exterior routing of an **autonomous system (AS)**).

An **autonomous system** is a collection of networks or subnets that are in the same administrative domain. You control and set the policy for what happens in the network or autonomous system.

The most popular protocol for an **EGP** is **Border Gateway Protocol (BGP)**, which is typically used by ISPs or really large corporations. As an administrator of a small to medium network, you'll probably never use BGP. (BGP will be discussed in the following section)

What kinds of great things the dynamic routing protocols provide for us. Well, the thing that comes to mind first is the **amount of time** and **energy** we save configuring routers. We won't have to go to every single router and define for it, with a static route, what and where every destination network is. We still have to know what the routing protocols are going to do and how they will do it, but the protocols will take care of most of the updating and sending information to each other.

The IGP splits into two primary categories: **distance vector (DV)** and **link-state (LS)** routing protocols. But in the distance vector category, for example, we have RIP and Interior Gateway Routing Protocol (IGRP). Under the link-state category are the **nonproprietary OSPF** and **Intermediate System-to-Intermediate System (IS-IS)** that was designed to work in larger internetworks.

[Previous](#)

[Next](#)

You have completed 0% of the lesson

0%

[Jump to...](#)