Unit 4

I] Internetworking:

Internetworking refers to the practice of connecting multiple computer networks together to create a larger, global network of networks. The primary goal of internetworking is to enable communication and data exchange between devices and users located on different networks.

Internetworking is made possible through the use of various networking technologies and protocols that facilitate data transmission across heterogeneous networks. The Internet is the most prominent example of a global internetwork, which interconnects millions of networks worldwide.

1. How networks differ

Networks can differ in many ways. Some of the differences, such as different modulation techniques or frame formats, are internal to the physical and data link layers. When packets sent by a source on one network must transit one or more foreign networks before reaching the destination network, many problems can occur at the interfaces between networks. To start with, the source needs to be able to address the destination.

- What do we do if the source is on an Ethernet network and the destination is on a WiMAX network?
- If packets on a connection-oriented network transit a connectionless network, they may arrive in a different order than they were sent.
- If one network has strong QoS and the other offers best effort service, it will be impossible to make bandwidth and delay guarantees for real-time traffic end to end.

Item	Some Possibilities
Service offered	Connectionless versus connection oriented
Addressing	Different sizes, flat or hierarchical
Broadcasting	Present or absent (also multicast)
Packet size	Every network has its own maximum
Ordering	Ordered and unordered delivery
Quality of service	Present or absent; many different kinds
Reliability	Different levels of loss
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, packet, byte, or not at all

Figure 5-38. Some of the many ways networks can differ.

2. How networks can be connected

There are two basic choices for connecting different networks: we can build devices that translate or convert packets from each kind of network into packets for each other network. Try to solve the problem by adding a layer of indirection and building a common layer on top of the different networks.

IP provides a universal packet format that all routers recognize and that can be passed through almost every network. IP has extended its reach from computer networks to take over the telephone network. It also runs on sensor networks and other tiny devices that were once presumed too resource-constrained to support it.

An internet comprised of 802.11, MPLS, and Ethernet networks is shown in Fig. 5-39(a). Suppose that the source machine on the 802.11 network wants to send a packet to the destination machine on the Ethernet network. Since these technologies are different, and they are further separated by another kind of network (MPLS), some added processing is needed at the boundaries between the networks. Because different networks may, in general, have different forms of addressing, the packet carries a network layer address that can identify any host across the three networks. The first boundary the packet reaches is when it transitions from an 802.11 network to an MPLS network.

MPLS stands for Multiprotocol Label Switching. It is a protocol used in computer networks to improve the efficiency of data transmission and forwarding. MPLS operates between Layer 2 (Data Link Layer) and Layer 3 (Network Layer) of the OSI model and is commonly used in wide area networks (WANs) and service provider networks.
The term "802.11 network" refers to a set of wireless local area network (WLAN) standards developed by the Institute of Electrical and Electronics Engineers (IEEE). The standards are denoted by the prefix "802.11" followed by a letter or combination of letters. These standards define the specifications for wireless communication between devices, such as computers, smartphones, tablets, and other devices, over radio frequencies. 1. Eg: 802.11a: This standard was one of the earliest introduced and operates in the 5 GHz frequency band. It supports data rates up to 54 Mbps. 2. 802.11b: Introduced around the same time as 802.11a, this standard operates in the 2.4 GHz frequency band and provides data rates up to 11 Mbps. Etc.
MPLS - Multiprotocol Label Switching (2.5 layer protocol) - YouTube What is MPLS? - YouTube What is MPLS (Multiprotocol Label Switching) - How does it work - YouTube

802.11 provides a connectionless service, but MPLS provides a connection-oriented service. This means that a virtual circuit must be set up to cross that network. Once the packet has travelled along the virtual circuit, it will reach the Ethernet network. At this boundary, the packet may be too large to be carried, since 802.11 can work with larger frames than Ethernet. To handle this problem, the packet is divided into fragments, and each fragment is sent separately. When the fragments reach the destination, they are reassembled. Then the packet has completed its journey. The protocol processing for this journey is

shown in Fig. 5-39(b). The source accepts data from the transport layer and generates a packet with the common network layer header, which is IP in this example.

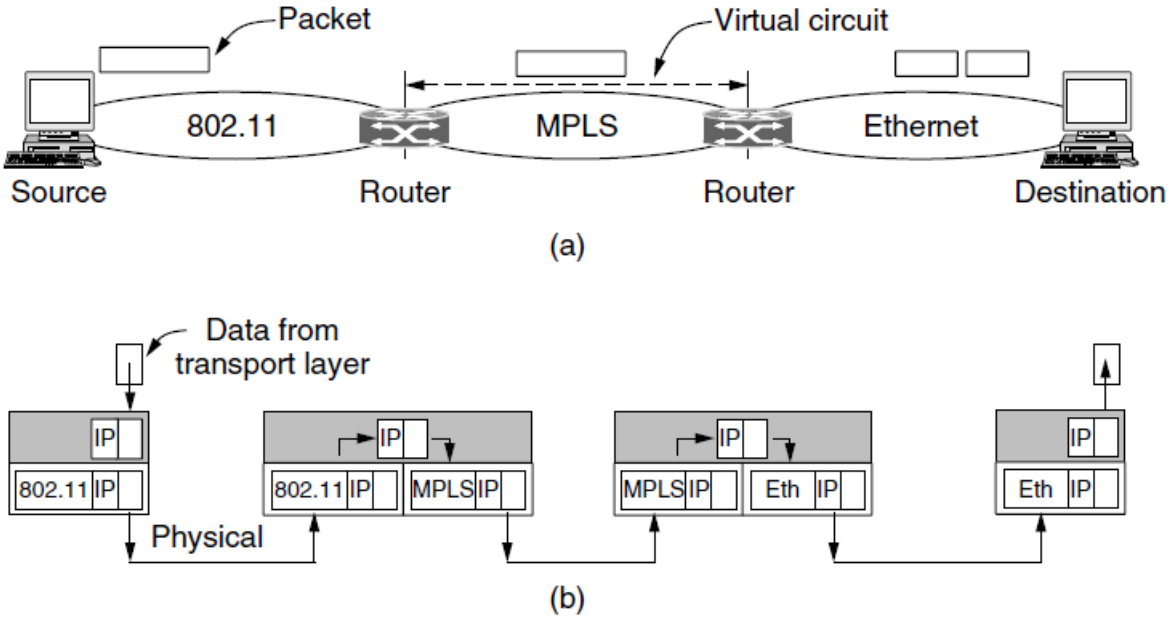


Figure 5-39. (a) A packet crossing different networks. (b) Network and link layer protocol processing.

Internet Vs Ethernet | Difference Between them with Comparison Chart - YouTube

The network header contains the ultimate destination address, which is used to determine that the packet should be sent via the first router. So the packet is encapsulated in an 802.11 frame whose destination is the first router and transmitted. At the router, the packet is removed from the frame's data field and the 802.11 frame header is discarded. The router now examines the IP address in the packet and looks up this address in its routing table. Based on this address, it decides to send the packet to the second router next. For this part of the path, an MPLS virtual circuit must be established to the second router and the packet must be encapsulated with MPLS headers that travel

this circuit. At the far end, the MPLS header is discarded and the network address is again consulted to find the next network layer hop. It is the destination itself. Since the packet is too long to be sent over Ethernet, it is split into two portions. Each of these portions is put into the data field of an Ethernet frame and sent to the Ethernet address of the destination. At the destination, the Ethernet header is stripped from each of the frames, and the contents are reassembled. The packet has finally reached its destination. Observe that there is an essential difference between the routed case and the switched (or bridged) case. With a router, the packet is extracted from the frame and the network address in the packet is used for deciding where to send it. With a switch (or bridge), the entire frame is transported on the basis of its MAC address. Switches do not have to understand the network layer protocol being used

to switch packets. Routers do.

Internetworking has been very successful at building large networks, but it only works when there is a common network layer. There have, in fact, been many network protocols over time. Getting everybody to agree on a single format is difficult when companies perceive it to their commercial advantage to have

a proprietary format that they control. The most relevant example now is probably IPv4 and IPv6. While these are both versions of IP, they are not compatible.

A router that can handle multiple network protocols is called a **multiprotocol router**. It must either translate the protocols, or leave connection for a higher protocol layer.

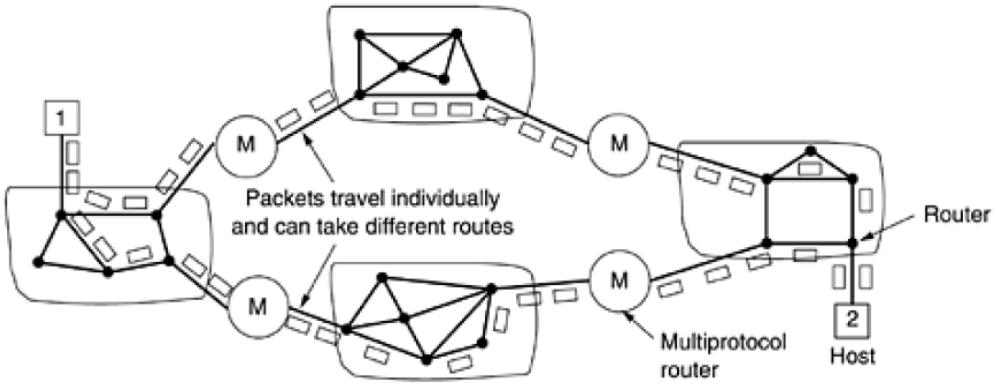
Connecting different networks, also known as **internetworking**, can be challenging due to various problems and issues that arise when attempting to establish seamless communication between diverse network environments. Some of the common problems in connecting different networks include:

1. **Addressing and Routing:** Different networks may use varying addressing schemes, protocols, or subnet configurations. When connecting these networks, it can be complicated to ensure that devices in one network can correctly address and communicate with devices in another network. The routing between networks becomes complex, and misconfigurations can lead to communication failures.
2. **Network Protocols and Standards:** Networks may be built using different protocols and standards, which can create incompatibility issues. For example, one network might use the TCP/IP protocol suite, while another uses IPX/SPX or AppleTalk. These differing protocols may not inherently understand each other, making it difficult for devices in one network to communicate with devices in another.
3. **Security and Access Control:** When connecting different networks, security becomes a significant concern. Different networks may have different security policies, authentication methods, and access control mechanisms. Ensuring consistent and appropriate security measures across all networks can be a complex task, and any misconfigurations or vulnerabilities can lead to potential security breaches.
4. **Network Performance:** Connecting different networks can impact overall network performance. The data transfer rate, latency, and bandwidth might differ significantly between networks, affecting the overall performance when data needs to traverse multiple networks to reach its destination.
5. **Network Management and Monitoring:** When networks are disparate, managing and monitoring them becomes more challenging. Centralized network management and monitoring become critical to ensure proper oversight and control, which can be difficult to achieve when dealing with networks that have different technologies and configurations.
6. **Quality of Service (QoS):** Different networks may have varying capabilities to prioritize and manage network traffic based on QoS requirements. When connecting these networks, maintaining consistent QoS for critical applications across all networks becomes a complex task.
7. **Firewalls and Network Address Translation (NAT):** Firewalls and NAT devices that protect individual networks can also become obstacles to interconnectivity. Proper configuration and rules need to be implemented to allow necessary communication while maintaining security.
8. **Network Stability and Reliability:** Interconnecting different networks introduces additional points of potential failure. If one network experiences instability or downtime, it can affect communication with other interconnected networks.
9. **Protocol Translation:** In cases where networks use entirely different protocols, intermediary devices may be required to translate data between protocols, adding complexity and potential points of failure.

3. Connectionless Internetworking

The alternative internetwork model is the datagram model, shown in Fig. 5-46. In this model, the only service the network layer offers to the transport layer is the ability to inject datagrams into the subnet and hope for the best. There is no notion of a virtual circuit at all in the network layer, let alone a concatenation of them. This model does not require all packets belonging to one connection to traverse the same sequence of gateways. In Fig. 5-46 datagrams from host 1 to host 2 are shown taking different routes through the internetwork. A routing decision is made separately for each packet, possibly depending on the traffic at the moment the packet is sent. This strategy can use multiple routes and thus achieve a higher bandwidth than the concatenated virtual-circuit model. On the other hand, there is no guarantee that the packets arrive at the destination in order, assuming that they arrive at all.

Figure 5-46. A connectionless internet.



The model of Fig. 5-46 is not quite as simple as it looks. For one thing, if each network has its own network layer protocol, it is not possible for a packet from one network to transit another one. One could imagine the multiprotocol routers actually trying to translate from one format to another, but unless the two formats are close relatives with the same information fields, such conversions will always be incomplete and often doomed to failure. For this reason, conversion is rarely attempted.

A second, and more serious, problem is addressing. Imagine a simple case: a host on the Internet is trying to send an IP packet to a host on an adjoining SNA network. The IP and SNA addresses are different.

SNA stands for Systems Network Architecture, which is a proprietary networking architecture developed by IBM in the 1970s. It was primarily designed for connecting IBM mainframe computers and their peripherals in a reliable and efficient manner. SNA served as the networking foundation for IBM's host-centric computing environments.

One would need a mapping between IP and SNA addresses in both directions. Furthermore, the concept of what is addressable is different. In IP, hosts (actually, interface cards) have addresses. In SNA, entities other than hosts (e.g., hardware devices) can also have addresses. At best, someone would have to maintain a database mapping everything to everything to the extent possible, but it would constantly be a source of trouble.

Another idea is to design a universal "internet" packet and have all routers recognize it. This approach is, in fact, what IP is—a packet designed to be carried through many networks. Of course, it may turn out that IPv4 (the current Internet protocol) drives all other formats out of the market, IPv6 (the future Internet protocol) does not catch on, and nothing new is ever invented, but history suggests otherwise. Getting everybody to agree to a single format is difficult when companies perceive it to their commercial advantage to have a proprietary format that they control.

The concatenated virtual-circuit model has essentially the same advantages as using virtual circuits within a single subnet: buffers can be reserved in advance, sequencing can be guaranteed, short headers can be used, and the troubles caused by delayed duplicate packets can be avoided. It also has the same disadvantages: table space required in the routers for each open connection, no alternate routing to avoid congested areas, and vulnerability to router failures along the path. It also has the disadvantage of being difficult, if not impossible, to implement if one of the networks involved is an unreliable datagram network. The properties of the datagram approach to internetworking are pretty much the same as those of datagram subnets: more potential for congestion, but also more potential for adapting to it, robustness in the face of router failures, and longer headers needed. Various adaptive routing algorithms are possible in an internet, just as they are within a single datagram network.

A major advantage of the datagram approach to internetworking is that it can be used over subnets that do not use virtual circuits inside. Many LANs, mobile networks (e.g., aircraft and naval fleets), and even some WANs fall into this category. When an internet includes one of these, serious problems occur if the internetworking strategy is based on virtual circuits.

What is IP address and types of IP address - IPv4 and IPv6 | TechTerms - YouTube

IP Address - IPv4 vs IPv6 Tutorial - YouTube

IPv4 vs IPv6 | Difference Between IPv4 and IPv6 | IP Address Explained | IP Address | Simplilearn - YouTube

Feature	IPv4	IPv6
Address Length	32 bits	128 bits
Address Representation	Dotted decimal (e.g., 192.168.1.1)	Eight groups of four hexadecimal digits separated by colons (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334)
Address Space	Approximately 4.3 billion addresses	Approximately 340 undecillion addresses (3.4 × 10^38)
Address Types	Public and Private addresses	Unique Global Unicast, Link-Local, Site-Local, Multicast, and others
Autoconfiguration	DHCP (Dynamic Host Configuration Protocol) and manual configuration	Stateful and Stateless Address Autoconfiguration (SLAAC)
Network Discovery	ARP (Address Resolution Protocol)	NDP (Neighbor Discovery Protocol)
Fragmentation	Routers and hosts can fragment packets	End-to-end path MTU discovery to avoid intermediate fragmentation
Header Format	Fixed-length header with options	Simplified and more efficient fixed-length header
Security	No inherent security features	IPsec (Internet Protocol Security) is an integral part of the protocol
NAT (Network Address Translation)	Often used due to IPv4 address exhaustion	Less reliance on NAT due to a vast address space
Backward Compatibility	Not directly compatible with IPv6	IPv6 is designed to be backward compatible with IPv4 using transition mechanisms

4. Tunneling

What is TUNNELING , What is TUNNELING in Computer Networks - YouTube

A technique of internetworking called **Tunneling** is used when the source and destination networks of the same type are to be connected through a network of a different type.

Tunneling is **a way to move packets from one network to another**.
Tunneling works via encapsulation: wrapping a packet inside another packet.

Tunneling refers to a technique that allows data to be transmitted securely across an untrusted network, such as the internet. It involves encapsulating data packets from one network protocol within the data

packets of another protocol, effectively creating a "tunnel" through which the data can pass. Tunneling is widely used to ensure data privacy and security when transmitting sensitive information over the internet or connecting remote networks. By encapsulating data and encrypting it, tunneling helps prevent unauthorized access, interception, and tampering of the transmitted data.

Handling the general case of making two different networks interwork is exceedingly difficult. However, there is a common special case that is manageable even for different network protocols. This case is where the source and destination hosts are on the same type of network, but there is a different network in between. As an example, think of an international bank with an IPv6 network in Paris, an IPv6 network in London and connectivity between the offices via the IPv4 Internet. This situation is shown in Fig. 5-40.

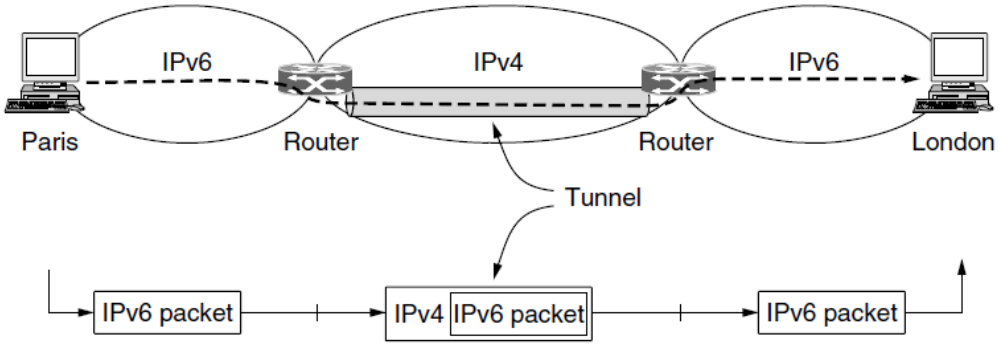


Figure 5-40. Tunneling a packet from Paris to London.

The solution to this problem is a technique called tunneling. To send an IP packet to a host in the London office, a host in the Paris office constructs the packet containing an IPv6 address in London, and sends it to the **multiprotocol router** that connects the Paris IPv6 network to the IPv4 Internet. When this router gets the IPv6 packet, it encapsulates the packet with an IPv4 header addressed to the IPv4 side of the multiprotocol router that connects to the London IPv6 network. That is, the router puts a (IPv6) packet inside a (IPv4) packet. When this wrapped packet arrives, the London router removes the original IPv6 packet and sends it onward to the destination host. The path through the IPv4 Internet can be seen as a big tunnel extending from one multiprotocol router to the other. The IPv6 packet just travels from one end of the tunnel to the other. It does not have to worry about dealing with IPv4 at all. Neither do the hosts in Paris or London. Only the multiprotocol routers have to understand both IPv4 and IPv6 packets. In effect, the entire trip from one multiprotocol router to the other is like a hop over a single link.

- Tunneling is a protocol that allows for the secure movement of data from one network to another.
- A technique of internetworking called **Tunneling** is used when source and destination networks of same type are to be connected through a network of different type.
- For example, let us consider an Ethernet to be connected to another Ethernet through a WAN.
- Tunneling works by encapsulating packets: wrapping packets inside of other packets.

Network Overlay: In computer networking, an overlay network is a virtual network that is built on top of an existing physical network infrastructure. It enables communication between devices or nodes that may not be directly connected in the underlying network. Overlay networks are often used to provide additional features, services, or security, and they can be created using various protocols and technologies.

Advantages of Network Overlay:

1. **Flexibility and Agility:** Overlay networks provide a high degree of flexibility and agility. They can be created and modified without necessarily altering the underlying physical network infrastructure. This makes it easier to adapt to changing requirements, add new services, or accommodate temporary connections.
2. **Virtualization and Multi-tenancy:** Overlay networks allow the creation of virtual networks on top of a shared physical infrastructure. This enables multi-tenancy, where different user groups or organizations can have their isolated virtual networks while sharing the same underlying resources.
3. **Enhanced Security:** Network overlays can offer improved security features, including encryption and tunneling, to protect data during transmission. Virtual Private Networks (VPNs) are a common example of a secure network overlay.
4. **Scalability:** Overlay networks can scale independently of the physical infrastructure. This means that network administrators can expand or shrink the overlay network according to the needs of the users or applications without affecting the underlying network.
5. **Protocol Translation:** Overlay networks can enable communication between devices or networks that use different protocols. By encapsulating data and translating protocols, overlays can bridge the gap between incompatible systems.

6. **Ease of Deployment and Management:** Implementing overlay networks often requires less effort and time compared to reconfiguring the physical network infrastructure. Additionally, management and monitoring of the overlay can be simplified through software-based tools.

Disadvantages of Network Overlay:

1. **Overhead and Complexity:** Overlay networks introduce additional overhead due to encapsulation and processing of data at multiple layers. This can lead to increased latency and reduced network performance.
2. **Potential for Overlapping IP Address Spaces:** When implementing overlay networks, there is a risk of using IP address spaces that overlap with the underlying network or other overlays. This can cause conflicts and communication issues.
3. **Dependency on Underlying Network Reliability:** While overlay networks can offer enhanced features, they still rely on the reliability and performance of the underlying physical network. Any issues in the base network can impact the overlay.
4. **Network Fragmentation:** As more overlays are added, the network can become fragmented, leading to a complex and challenging management environment. This fragmentation may result in inefficient use of resources and increased management complexity.
5. **Compatibility and Interoperability:** Integrating and ensuring compatibility between various overlay technologies and the underlying network can be a complex task. Compatibility issues may arise when overlay networks are deployed across different vendors' equipment.
6. **Performance Variation:** The performance of overlay networks may vary based on factors like network congestion, the distance between nodes, and the available bandwidth. Different applications and services within the overlay may experience varying levels of performance.

5. Internetwork Routing

Internetwork routing, also known as interdomain routing or simply internet routing, is the process of forwarding data packets across multiple interconnected networks or domains on the internet. It involves the exchange of routing information between different autonomous systems (ASes) or networks to determine the best path for data to reach its destination.

The internet is a massive network of networks, and each network is managed by different organizations or Internet Service Providers (ISPs). To enable communication between devices in one network and devices in another network, internetwork routing protocols are used to discover and maintain the paths between them.

Routing through an internet poses the same basic problem as routing within a single network, but with some added complications. To start, the networks may internally use different routing algorithms. For example, one network may use link state routing and another distance vector routing. Since link state algorithms need to know the topology but distance vector algorithms do not, this difference alone would make it unclear how to find the shortest paths across the internet.

Networks run by different operators lead to bigger problems. First, the operators may have different ideas about what is a good path through the network. One operator may want the route with the least delay, while another may want the most inexpensive route. This will lead the operators to use different quantities to set the shortest-path costs (e.g., milliseconds of delay vs. monetary cost). The weights will not be comparable across networks, so shortest paths on the internet will not be well defined.

Worse yet, one operator may not want another operator to even know the details of the paths in its network, perhaps because the weights and paths may reflect sensitive information (such as the monetary cost) that represents a competitive business advantage.

Finally, the internet may be much larger than any of the networks that comprise it. It may therefore require routing algorithms that scale well by using a hierarchy, even if none of the individual networks need to use a hierarchy. All of these considerations lead to a two-level routing algorithm.

A two-level routing algorithm is a routing scheme that divides the routing process into two distinct levels, each responsible for handling specific aspects of the routing process. This approach is often used in large-scale networks to improve scalability and efficiency. The two levels are typically referred to as the "Intra-domain" and "Inter-domain" routing.

1. **Intra-domain Routing**: Intra-domain routing, also known as Interior Gateway Protocol (IGP) routing, deals with routing within a single autonomous system (AS) or administrative domain. An autonomous system is a collection of networks under a common administration and a single routing policy. Intra-domain routing protocols are used to exchange routing information within this autonomous system.

   Common Intra-domain routing protocols include:

   - **OSPF** (Open Shortest Path First): A link-state routing protocol used in IP networks, often employed in large enterprise networks or Internet Service Providers (ISPs).

     [HINDI] OSPF | Animation video | Network Kings - YouTube

   - **RIP** (Routing Information Protocol): A distance-vector routing protocol that uses the hop count as the metric and is mainly used in smaller networks.

2. **Inter-domain Routing:** Inter-domain routing, also known as Exterior Gateway Protocol (EGP) routing, is responsible for exchanging routing information between different autonomous systems (ASes). The goal is to enable communication between different networks, each under separate administrative control.

   The **Border Gateway Protocol (BGP) is** the most widely used inter-domain routing protocol. BGP is a path-vector protocol that allows Autonomous Systems to exchange routing information to determine the best path to reach destinations in other ASes. BGP plays a critical role in enabling internet connectivity and ensuring that packets traverse multiple networks to reach their final destinations.

   What is BGP (Border Gateway Protocol)? An Introduction - YouTube

By dividing the routing process into intra-domain and inter-domain levels, two-level routing algorithms offer several benefits, including better scalability, enhanced control over routing policies within a single AS, and simplified routing management across multiple ASes. This approach helps efficiently manage routing tables and traffic flow in large and complex networks, making it easier to maintain a stable and reliable internet infrastructure.

Feature	Intra-domain Routing	Inter-domain Routing
Scope	Within a single Autonomous System (AS)	Between different Autonomous Systems (ASes)
Purpose	Establish routes within the same network domain	Facilitate communication between different networks
Protocols	OSPF, RIP, IS-IS, EIGRP (for some vendors)	BGP (Border Gateway Protocol)
Metric Calculation	Typically uses link metrics (e.g., bandwidth)	Path attributes (e.g., AS path, preference, MED)
Administrative Control	Managed by a single administrative entity	Involves coordination between multiple administrative entities
Scalability	Suitable for large-scale networks	Handles global internet-scale routing
Convergence Speed	Generally faster convergence due to smaller network size and frequent updates	Slower convergence due to the complexity of inter-domain paths

Policy Flexibility	Administrators have fine-grained control over intra-domain routing policies	Inter-domain policies are often more complex and negotiated
Network Topology	Focuses on the internal topology of an AS	Concerned with the interconnections of multiple ASes
Traffic Engineering	Can be used for traffic engineering within an AS	Essential for managing traffic flow across AS boundaries
Exchange of Routing Information	Within an AS, routing information is shared among routers within the same AS	Between ASes, routing information is exchanged at the boundaries
Security Considerations	Less concern for security as the AS is usually under the same administrative control	More security considerations due to the involvement of multiple ASes
Example Scenario	Routing within a large company network	Routing between different ISPs or corporate networks

6. Fragmentation

Each network or link imposes some maximum size on its packets. These limits have various causes, among them

1. Hardware (e.g., the size of an Ethernet frame).
2. Operating system (e.g., all buffers are 512 bytes).
3. Protocols (e.g., the number of bits in the packet length field).
4. Compliance with some (inter)national standard.
5. Desire to reduce error-induced retransmissions to some level.
6. Desire to prevent one packet from occupying the channel too long.

The result of all these factors is that the network designers are not free to choose any old maximum packet size they wish. Maximum payloads for some common technologies are 1500 bytes for Ethernet and 2272 bytes for 802.11. IP is more generous, allows for packets as big as 65,515 bytes.

Packet fragmentation is a process in computer networking where a large data packet is broken down into smaller fragments to fit within the Maximum Transmission Unit (MTU) size of the network medium. The MTU is the maximum size of a data packet that can be transmitted over a particular network link without being fragmented.

When a data packet is larger than the MTU of the network link it needs to traverse, the router or device responsible for forwarding the packet may fragment it into smaller pieces before transmitting it. Fragmentation occurs at the network layer (Layer 3) of the OSI model.

Network Basics - Maximum Transmission Unit (MTU) - YouTube

The process of packet fragmentation involves the following steps:

1. **Packet Size Check:** When a device receives a data packet for forwarding, it checks the packet's size against the MTU of the outgoing interface. If the packet size exceeds the MTU, it needs to be fragmented.
2. **Fragmentation:** The original packet is divided into smaller fragments, each fitting within the MTU of the network link. The header information is copied to each fragment, ensuring that each fragment has enough information for reassembly at the destination.
3. **Transmission:** The smaller fragments are transmitted independently over the network to their destination. Each fragment follows its own path, and they may arrive at the destination in a different order than they were sent.
4. **Reassembly:** Upon reaching the destination, the receiving device or the final destination host reassembles the fragments back into the original packet using information from the headers of each fragment. The packet is then passed up the network stack for processing.

It's important to note that packet fragmentation can introduce overhead, as additional headers are added to each fragment. Fragmentation can also lead to an increase in network latency and, in some cases, negatively impact network performance. To mitigate these issues, modern network protocols, such as IPv6, encourage the use of Path MTU Discovery (PMTUD) to avoid fragmentation by dynamically determining the optimal MTU size for the path and adjusting packet sizes accordingly. This helps in reducing the reliance on fragmentation and improving overall network efficiency.

1. **Nontransparent Fragmentation:** Nontransparent fragmentation refers to a method where the responsibility for packet fragmentation lies with the sending device or host. When a data packet is larger than the MTU of the outgoing network link, the sending device is responsible for breaking down the packet into smaller fragments that fit within the MTU size.

Each fragment created by the sending device includes its own headers, including the relevant network layer (e.g., IP) and transport layer (e.g., TCP or UDP) headers. When the fragments reach their destination, the receiving device or final destination host must reassemble the fragments back into the original packet. In nontransparent fragmentation, the intermediate network devices, such as routers, are not aware of the fragmentation process. They treat each fragment as an independent packet and forward them to their destination without knowledge of the original packet's structure.

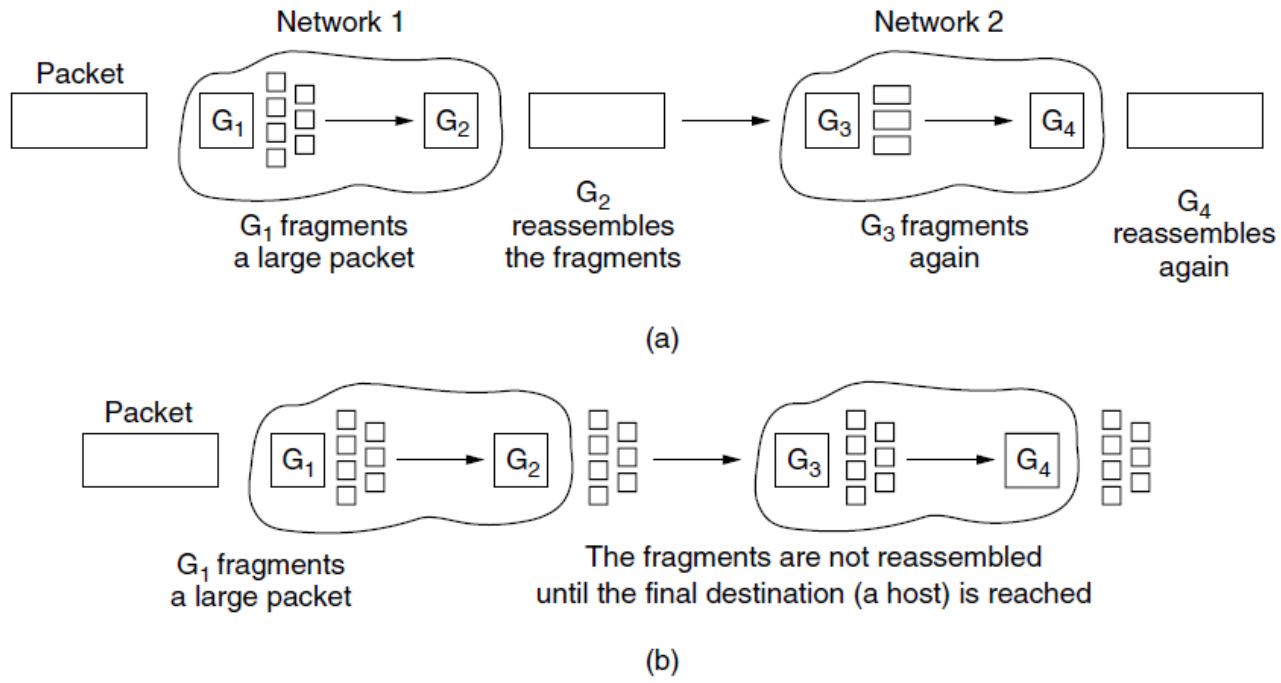


Figure 5-42. (a) Transparent fragmentation. (b) Nontransparent fragmentation.

2. **Transparent Fragmentation:** Transparent fragmentation, on the other hand, refers to a method where the network devices and routers in the path of the packet take responsibility for fragmentation. When a data packet is larger than the MTU of an outgoing network link, the intermediate network devices along the path will detect the oversize packet and fragment it into smaller pieces that fit within the MTU size of the outgoing link.

In transparent fragmentation, the intermediate network devices are actively involved in the process of breaking down and reassembling the packet. They modify the packet headers to account for fragmentation, and the process is entirely transparent to the sender and receiver. The sender sends the original packet, and the receiver receives the reassembled packet, as if no fragmentation occurred.

Transparent fragmentation can help in situations where nontransparent fragmentation might not be feasible or practical, especially when the sender or receiver devices do not support or cannot handle fragmentation. Transparent fragmentation ensures that packets can traverse the network without issues related to packet size limitations.

Path MTU Discovery (PMTUD) is a technique used in computer networking to dynamically discover the Maximum Transmission Unit (MTU) size of a network path between two devices. The MTU is the maximum size of a data packet that can be transmitted over a specific network link without fragmentation.

Path MTU Discovery - YouTube

PMTUD is especially important in scenarios where the network path includes links with different MTU sizes or where firewalls and routers along the path might block or discard fragmented packets. By determining the path's MTU, PMTUD ensures that data packets are sent with an appropriate size that fits within the MTU of each network link, avoiding fragmentation and potential issues with packet loss or delay.

The PMTUD process typically works as follows:

1. **Initial Packet:** When a device wants to send a data packet to a destination, it starts by sending an initial packet with a relatively large size, commonly the standard IPv6 minimum MTU (1280 bytes) or the IPv4 default MTU (1500 bytes).
2. **Fragmentation Check:** If any router along the path detects that the packet is too large for its outgoing link's MTU, it will not fragment the packet (as in transparent fragmentation) but instead will drop it and send an Internet Control Message Protocol (ICMP) "Destination Unreachable - Fragmentation Needed" message back to the sender.
3. **Packet Size Reduction:** Upon receiving the "Fragmentation Needed" message, the sender reduces the packet size and retransmits the data packet with a smaller MTU value. This process continues iteratively until the sender determines the path's optimal MTU.
4. **MTU Discovery:** The sender uses the smallest MTU value that successfully reaches the destination without being fragmented. This discovered MTU value is then used for subsequent data packets sent to the same destination.

PMTUD is more commonly used with IPv6 networks, as IPv6 does not support network layer fragmentation like IPv4. In IPv4, network layer fragmentation (nontransparent fragmentation) is more prevalent, but PMTUD can still be beneficial to avoid fragmentation and improve network efficiency.

Path MTU Discovery is an essential mechanism for maintaining smooth data transmission in modern networks, ensuring that packets are appropriately sized and able to traverse various network links without encountering fragmentation-related issues.

Advantages of Path MTU Discovery:

1. **Reduced Fragmentation:** PMTUD helps prevent packet fragmentation by dynamically determining the optimal MTU size for the path between the sender and receiver. This ensures that data packets are sent with an appropriate size that fits within the MTU of each network link, avoiding fragmentation and the potential issues related to fragmented packets.
2. **Improved Performance:** By avoiding fragmentation, PMTUD reduces the processing overhead on intermediate network devices. It ensures that data packets can be transmitted without

additional fragmentation and reassembly operations, leading to improved network performance and reduced latency.

3. **Efficient Data Transmission:** With PMTUD, data packets are sent with the largest MTU size possible for the path, maximizing the payload size of each packet. This results in more efficient data transmission as fewer packets need to be sent to transfer the same amount of data.

Disadvantages and Considerations of Path MTU Discovery:

1. **ICMP Filtering**: Some networks or firewalls may block or filter ICMP packets, including the "Fragmentation Needed" messages used in PMTUD. When these messages are blocked, PMTUD may not function properly, and data packets may encounter issues with fragmentation.
2. **Incomplete or Misconfigured PMTUD:** In some cases, PMTUD might not work correctly due to misconfigurations, software issues, or incomplete implementations. If PMTUD is not functioning as intended, data packets may not be properly sized for the path, potentially leading to fragmentation-related problems.
3. **Vulnerable to PMTUD Black Hole:** In rare cases, a PMTUD black hole may occur. This situation arises when an ICMP "Fragmentation Needed" message is lost or blocked by an intermediate device, and the sender assumes that the MTU is larger than it actually is. Consequently, the sender may continue to send large packets, which will then be dropped, causing performance issues.
4. **Additional Overhead:** The PMTUD process requires the exchange of additional packets (ICMP "Fragmentation Needed" messages) between the sender and intermediate devices, which may slightly increase the overhead of the data transmission process.

II] The Network Layer in the Internet:

1. The IP Protocol, IP Addresses

The IP (Internet Protocol) is a fundamental protocol in computer networking that facilitates communication and data exchange between devices on a network, especially on the internet. It provides a standardized way for data packets to be routed from their source to their destination across different networks and devices. IP is an essential building block of the internet and is used in conjunction with other protocols to enable the transmission of data.

There are two main versions of the Internet Protocol: IPv4, IPv6

IP operates at the Network Layer (Layer 3) of the OSI (Open Systems Interconnection) model. It provides the basic functions of addressing and routing, enabling devices to find each other on a network and ensuring that data packets are delivered to the correct destination.

IP is connectionless and best-effort, which means that it does not establish a dedicated connection between devices before sending data. Instead, it breaks data into packets and sends them independently. This approach allows for greater flexibility and efficiency in network communication but does not guarantee delivery or reliability.

To facilitate reliable communication, higher-level protocols (such as TCP, UDP, and ICMP) often work in conjunction with IP. TCP (Transmission Control Protocol) provides reliable, connection-oriented communication, while UDP (User Datagram Protocol) offers connectionless, lightweight communication. ICMP (Internet Control Message Protocol) is used for error reporting, diagnostics, and network management.

IP Addresses

A defining feature of IPv4 is its 32-bit addresses. Every host and router on the Internet has an IP address that can be used in the Source address and Destination address fields of IP packets. It is important to note that an IP address does not actually refer to a host. It really refers to a network interface, so if a host is on two networks, it must have two IP addresses. However, in practice, most hosts are on one network and thus have one IP address. In contrast, routers have multiple interfaces and thus multiple IP addresses.

There are two main versions of the Internet Protocol:

1. **IPv4 (Internet Protocol version 4)**: This is the most widely used version of IP and is characterized by a 32-bit address space, which allows for approximately 4.3 billion unique addresses. An IPv4 address is typically represented in dotted-decimal notation, such as "192.168.1.1". However, due to the increasing number of devices connected to the internet, the availability of IPv4 addresses has become limited.
2. **IPv6 (Internet Protocol version 6)**: IPv6 was introduced to address the limitations of IPv4 by using a 128-bit address space, allowing for an astronomically larger number of unique addresses (about 340 undecillion). IPv6 addresses are usually represented using hexadecimal notation with colons, such as "2001:0db8:85a3:0000:0000:8a2e:0370:7334". IPv6 adoption has been growing to accommodate the growing number of devices and the expanding internet landscape.

2. Internet Control Protocols

In addition to IP, which is used for data transfer, the Internet has several companion control protocols that are used in the network layer. They include ICMP, ARP, and DHCP.

IMCP—The Internet Control Message Protocol

How Internet Control Message Protocol (ICMP) Works? - YouTube

The operation of the Internet is monitored closely by the routers. When something unexpected occurs during packet processing at a router, the event is reported to the sender by the **ICMP** (**Internet Control Message Protocol**). ICMP is also used to test the Internet. About a dozen types of ICMP messages are defined. Each ICMP message type is carried encapsulated in an IP packet. The most important ones are listed in Fig. 5-60.

The DESTINATION UNREACHABLE message is used when the router cannot locate the destination or when a packet with the *DF* bit cannot be delivered because a ‘‘small-packet’’ network stands in the way.

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo and echo reply	Check if a machine is alive
Timestamp request/reply	Same as Echo, but with timestamp
Router advertisement/solicitation	Find a nearby router

Figure 5-60. The principal ICMP message types.

The TIME EXCEEDED message is sent when a packet is dropped because its *TtL (Time to live)* counter has reached zero. This event is a symptom that packets are looping, or that the counter values are being set too low.

The PARAMETER PROBLEM message indicates that an illegal value has been detected in a header field. This problem indicates a bug in the sending host’s IP software or possibly in the software of a router transited.

The SOURCE QUENCH message was long ago used to throttle hosts that were sending too many packets. When a host received this message, it was expected to slow down. It is rarely used anymore because when congestion occurs, these packets tend to add more fuel to the fire and it is unclear how to respond to them. Congestion control in the Internet is now done largely by taking action in the transport layer, using packet losses as a congestion signal;

<

The REDIRECT message is used when a router notices that a packet seems to be routed incorrectly. It is used by the router to tell the sending host to update to a better route.

The ECHO and ECHO REPLY messages are sent by hosts to see if a given destination is reachable and currently alive. Upon receiving the ECHO message, the destination is expected to send back an ECHO REPLY message. These messages are used in the **ping** utility that checks if a host is up and on the Internet.

The TIMESTAMP REQUEST and TIMESTAMP REPLY messages are similar, except that the arrival time of the message and the departure time of the reply are recorded in the reply. This facility can be used to measure network performance.

The ROUTER ADVERTISEMENT and ROUTER SOLICITATION messages are used to let hosts find nearby routers. A host needs to learn the IP address of at least one router to be able to send packets off the local network.

The Internet Control Message Protocol (ICMP) is a network protocol used in the Internet Protocol (IP) suite. ICMP is primarily used for diagnostic and error reporting purposes within IP networks. It allows network devices to communicate error messages, operational information, and other control messages related to the functioning of the network. Some of the common message types in ICMP include:

1. **Echo Request and Echo Reply (Ping):** ICMP Echo Request (Type 8) is used to request an "echo" from a target host, often referred to as "pinging." The target host responds with an ICMP Echo Reply (Type 0), indicating its availability and responsiveness.
2. **Destination Unreachable (Type 3):** This message type is used to indicate that a destination host or network is unreachable for various reasons, such as network congestion, unreachable host, or protocol unreachable.
3. **Time Exceeded (Type 11):** This message type is used to indicate that a packet has exceeded its time-to-live (TTL) value while traversing through routers. It is often used to detect routing loops or network issues.
4. **Redirect Message (Type 5):** A router can send an ICMP Redirect message to inform a host that a better route is available for a specific destination.
5. **Router Advertisement and Router Solicitation (Type 9 and Type 10):** These messages are used in the context of IPv6 to facilitate the autoconfiguration of network interfaces and to discover routers on the local network.
6. **Parameter Problem (Type 12):** This message is used to indicate that a problem has been detected with the IP header, such as an unrecognized option or an incorrect length.
7. **Timestamp Request and Timestamp Reply (Type 13 and Type 14):** These messages are used to request and respond with timestamps for diagnostic and timing purposes.
8. **Address Mask Request and Address Mask Reply (Type 17 and Type 18):** These messages are used to determine the subnet mask of a network, particularly in older versions of ICMP.
9. **Source Quench (Type 4):** This message type is used to indicate to a sender that its traffic is causing congestion and should slow down.

ARP—The Address Resolution Protocol

Although every machine on the Internet has one or more IP addresses, these addresses are not sufficient for sending packets. Data link layer NICs (Network Interface Cards) such as Ethernet cards do not understand Internet addresses. In the case of Ethernet, every NIC ever manufactured comes equipped with a unique 48-bit Ethernet address. Manufacturers of Ethernet NICs request a block of Ethernet addresses from IEEE to ensure that no two NICs have the same address (to avoid conflicts should the two NICs ever appear on the same LAN). The NICs send and receive frames based on 48-bit Ethernet addresses. They know nothing at all about 32-bit IP addresses.

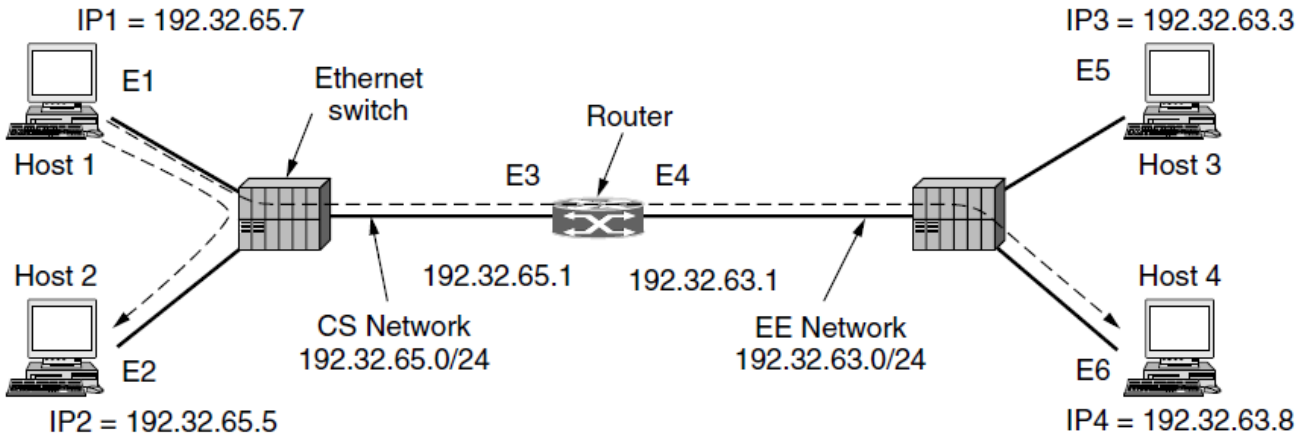
The question now arises, how do IP addresses get mapped onto data link layer addresses, such as Ethernet? To explain how this works, let us use the example of Fig. 5-61, in which a small university with two /24 networks is illustrated. One network (CS) is a switched Ethernet in the Computer Science Dept. It has the prefix 192.32.65.0/24. The other LAN (EE), also switched Ethernet, is in Electrical Engineering and has the prefix 192.32.63.0/24. The two LANs are connected by an IP router. Each machine on an Ethernet and each interface on the router has a unique Ethernet address, labeled *E1* through *E6*, and a unique IP address on the CS or EE network.

Let us start out by seeing how a user on host 1 sends a packet to a user on host 2 on the CS network. Let us assume the sender knows the name of the intended receiver, possibly something like *eagle.cs.uni.edu*. The first step is to find the IP address for host 2. This lookup is performed by DNS. Assume that DNS returns the IP address for host 2 (192.32.65.5).

The upper layer software on host 1 now builds a packet with 192.32.65.5 in the *Destination address* field and gives it to the IP software to transmit. The IP software can look at the address and see that the destination is on the CS network, (i.e., its own network). However, it still needs some way to find the destination's Ethernet address to send the frame. One solution is to have a configuration file somewhere in the system that maps IP addresses onto Ethernet addresses.

While this solution is certainly possible, for organizations with thousands of machines keeping all these files up to date is an error-prone, time-consuming job.

A better solution is for host 1 to output a broadcast packet onto the Ethernet asking who owns IP address 192.32.65.5. The broadcast will arrive at every machine on the CS Ethernet, and each one will check its IP address. Host 2 alone will respond with its Ethernet address (*E2*). In this way host 1 learns that IP address 192.32.65.5 is on the host with Ethernet address *E2*. The protocol used for asking this question and getting the reply is called **ARP (Address Resolution Protocol)**. Almost every machine on the Internet runs it.



Frame	Source IP	Source Eth.	Destination IP	Destination Eth.
Host 1 to 2, on CS net	IP1	E1	IP2	E2
Host 1 to 4, on CS net	IP1	E1	IP4	E3
Host 1 to 4, on EE net	IP1	E4	IP4	E6

Figure 5-61. Two switched Ethernet LANs joined by a router.

The advantage of using ARP over configuration files is the simplicity. The system manager does not have to do much except assign each machine an IP address and decide about subnet masks. ARP does the rest. At this point, the IP software on host 1 builds an Ethernet frame addressed to *E2*, puts the IP packet (addressed to 192.32.65.5) in the payload field, and dumps it onto the Ethernet. The IP and Ethernet addresses of this packet are given in Fig. 5-61. The Ethernet NIC of host 2 detects this frame, recognizes it as a frame for itself, scoops it up, and causes an interrupt. The Ethernet driver extracts the IP packet from the payload and passes it to the IP software, which sees that it is correctly addressed and processes it.

Various optimizations are possible to make ARP work more efficiently. To start with, once a machine has run ARP, it caches the result in case it needs to contact the same machine shortly. Next time it will find the mapping in its own cache, thus eliminating the need for a second broadcast.

A Gratuitous ARP (Address Resolution Protocol) is an ARP packet sent by a network device, typically a computer or a router, in order to announce its own IP-to-MAC address mapping to the local network. Unlike a regular ARP request that asks for the MAC address corresponding to a specific IP address, a gratuitous ARP is not a request but rather a self-initiated broadcast message.

Now let us look at Fig. 5-61 again, only this time assume that host 1 wants to send a packet to host 4 (192.32.63.8) on the EE network. Host 1 will see that the destination IP address is not on the CS network. It knows to send all such off-network traffic to the router, which is also known as the **default gateway**. By convention, the default gateway is the lowest address on the network (198.31.65.1). To send a frame to the router, host 1 must still know the Ethernet address of the router interface on the CS network.

It discovers this by sending an ARP broadcast for 198.31.65.1, from which it learns *E3*. It then sends the frame. The same lookup mechanisms are used to send a packet from one router to the next over a sequence of routers in an Internet path. When the Ethernet NIC of the router gets this frame, it gives the packet to the IP software. It knows from the network masks that the packet should be sent onto the EE network where it will reach host 4. If the router does not know the Ethernet address for host 4, then it will use ARP again. The table in Fig. 5-61 lists the source and destination Ethernet and IP addresses that are present in the frames as observed on the CS and EE networks. Observe that the Ethernet addresses change with the frame on each network while the IP addresses remain constant (because they indicate the endpoints across all of the interconnected networks). It is also possible to send a packet from host 1 to host 4 without host 1 knowing that host 4 is on a different network. The solution is to have the router answer ARPs on the CS network for host 4 and give its Ethernet address, *E3*, as the response. It is not possible to have host 4 reply directly because it will not see the ARP request (as routers do not forward Ethernet-level broadcasts). The router will then receive frames sent to 192.32.63.8 and forward them onto the EE network. This solution is called **proxy ARP**. It is used in special cases in which a host wants to appear on a network even though it actually resides on another network. A common situation, for example, is a mobile computer that wants some other node to pick up packets for it when it is not on its home network.

DHCP—The Dynamic Host Configuration Protocol

ARP (as well as other Internet protocols) makes the assumption that hosts are configured with some basic information, such as their own IP addresses. How do hosts get this information? It is possible to manually configure each computer, but that is tedious and error-prone. There is a better way, and it is called **DHCP** (**Dynamic Host Configuration Protocol**). With DHCP, every network must have a DHCP server that is responsible for configuration. When a computer is started, it has a built-in Ethernet or other link layer address embedded in the NIC, but no IP address. Much like ARP, the computer broadcasts a request for an IP address on its network. It does this by using a DHCP DISCOVER packet. This packet must reach the DHCP server. If that server is not directly attached to the network, the router will be configured to receive DHCP broadcasts and relay them to the DHCP server, wherever it is located. When the server receives the request, it allocates a free IP address and sends it to the host in a DHCP OFFER packet (which again may be relayed via the router). To be able to do this work even when hosts do not have IP addresses, the server identifies a host using its Ethernet address (which is carried in the DHCP DISCOVER packet). An issue that arises with automatic assignment of IP addresses from a pool is for how long an IP address should be allocated. If a host leaves the network and does not return its IP address to the DHCP server, that address will be permanently lost. After a period of time, many addresses may be lost. To prevent that from happening, IP address assignment may be for a fixed period of time, a technique called **leasing**. Just before the lease expires, the host must ask for a DHCP renewal.

If it fails to make a request or the request is denied, the host may no longer use the IP address it was given earlier.

DHCP is described in RFCs 2131 and 2132. It is widely used in the Internet to configure all sorts of parameters in addition to providing hosts with IP addresses. As well as in business and home networks, DHCP is used by ISPs to set the parameters of devices over the Internet access link, so that customers do not need to phone their ISPs to get this information. Common examples of the information that is configured include the network mask, the IP address of the default gateway, and the IP addresses of DNS and time servers. DHCP has largely replaced earlier protocols (called RARP and BOOTP) with more limited functionality.

The Dynamic Host Configuration Protocol, commonly known as DHCP, is a network management protocol used to automatically assign and manage IP addresses, subnet masks, gateway addresses, and other network configuration parameters to devices in a TCP/IP network. DHCP simplifies the process of configuring network settings for devices and helps to reduce administrative overhead.

DHCP offers several advantages, including:

1. **Simplified Network Management**: DHCP eliminates the need for manual IP address configuration on each device, making network administration more efficient and reducing the risk of configuration errors.
2. **IP Address Pooling**: DHCP servers maintain a pool of available IP addresses that can be dynamically assigned to devices as they connect to the network.
3. **Centralized Configuration**: Network settings and parameters are centrally managed on the DHCP server, allowing for easier changes and updates.
4. **Address Conservation**: DHCP helps prevent IP address exhaustion by efficiently managing IP address allocation and reuse.
5. **Dynamic Network Changes**: DHCP accommodates changes in network topology and device connectivity without requiring manual configuration adjustments.

DHCP is commonly used in home networks, small businesses, and large enterprise environments to streamline the process of assigning and managing IP addresses. It's an essential component of modern network infrastructure that plays a crucial role in making IP-based communication and networking easier to manage and scale.

3. OSPF- Interior Gateway Routing Protocol

The Open Shortest Path First (OSPF) algorithm is a widely used routing protocol in computer networks. It is designed to determine the best paths for routing data packets between routers within an Autonomous System (AS), which is a collection of IP networks and routers under the control of a single organization. The Internet is made up of a large number of independent networks or **ASes (Autonomous Systems)** that are operated by different organizations, usually a company, university, or ISP. Inside of its own network, an organization can use its own algorithm for internal routing, or **intradomain routing**, as it is more commonly known. An intradomain routing protocol is also called an **interior gateway protocol**.

Early intradomain routing protocols used a distance vector design, based on the distributed Bellman-Ford algorithm inherited from the ARPANET. RIP (Routing Information Protocol) is the main example that is used to this day. It works well in small systems, but less well as networks get larger. It also suffers from the count-to-infinity problem and generally slow convergence. The ARPANET switched over to a link state protocol in May 1979 because of these problems, and in 1988 IETF began work on a link state protocol for intradomain routing. That protocol, called **OSPF** (**Open Shortest Path First**), became a standard in 1990. It drew on a protocol called **IS-IS** (**Intermediate-System to Intermediate-System**), which became an ISO standard.

OSPF supports both point-to-point links (e.g., SONET) and broadcast networks (e.g., most LANs). Actually, it is able to support networks with multiple routers, each of which can communicate directly with the others (called **multiaccess networks**) even if they do not have broadcast capability.
An example of an autonomous system network is given in Fig. 5-64(a). Hosts are omitted because they do not generally play a role in OSPF, while routers and networks (which may contain hosts) do. Most of the routers in Fig. 5-64(a) are connected to other routers by point-to-point links, and to networks to reach the hosts on those networks. However, routers *R3*, *R4*, and *R5* are connected by a broadcast LAN such as switched Ethernet.

OSPF operates by abstracting the collection of actual networks, routers, and links into a directed graph in which each arc is assigned a weight (distance, delay, etc.). A point-to-point connection between two routers is represented by a pair of arcs, one in each direction. Their weights may be different. A broadcast network is represented by a node for the network itself, plus a node for each router. The arcs from that network node to the routers have weight 0. They are important nonetheless, as without them there is no path through the network. Other networks, which have only hosts, have only an arc reaching them and not one returning. This structure gives routes to hosts, but not through them.

Figure 5-64(b) shows the graph representation of the network of Fig. 5-64(a). What OSPF fundamentally does is represent the actual network as a graph like this and then use the link state method to have every router compute the shortest path from itself to all other nodes. Multiple paths may be found that are equally short. In this case, OSPF remembers the set of shortest paths and during packet forwarding, traffic is split across them. This helps to balance load. It is called **ECMP** (**Equal Cost MultiPath**).

Many of the ASes in the Internet are themselves large and nontrivial to manage. To work at this scale, OSPF allows an AS to be divided into numbered **areas**, where an area is a network or a set of contiguous networks. Areas do not overlap but need not be exhaustive, that is, some routers may belong to no area.

Routers that lie wholly within an area are called **internal routers**. An area is a generalization of an individual network. Outside an area, its destinations are visible but not its topology. This characteristic helps routing to scale.

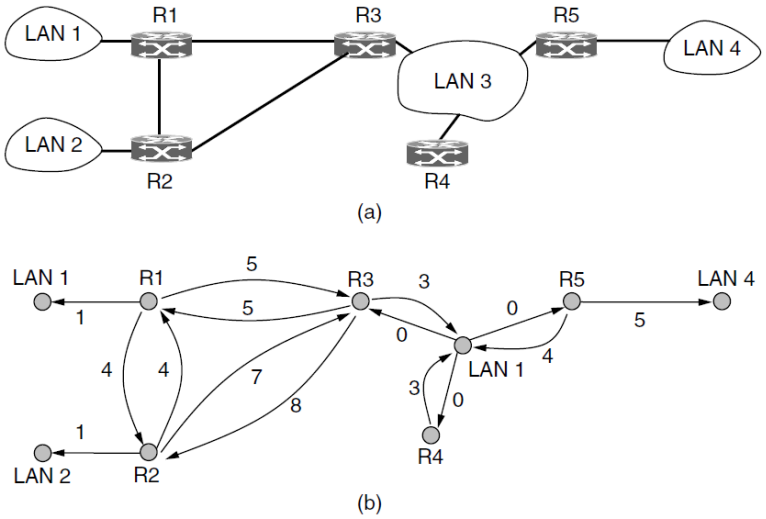


Figure 5-64. (a) An autonomous system. (b) A graph representation of (a).

Every AS has a **backbone area**, called area 0. The routers in this area are called **backbone routers**. All areas are connected to the backbone, possibly by tunnels, so it is possible to go from any area in the AS to any other area in the AS via the backbone. A tunnel is represented in the graph as just another arc with a cost. As with other areas, the topology of the backbone is not visible outside the backbone.

Each router that is connected to two or more areas is called an **area border router**. It must also be part of the backbone. The job of an area border router is to summarize the destinations in one area and to inject this summary into the other areas to which it is connected. This summary includes cost information but not all the details of the topology within an area. Passing cost information allows hosts in other areas to find the best area border router to use to enter an area. Not passing topology information reduces traffic and simplifies the shortest-path computations of routers in other areas. However, if there is only one border router out of an area, even the summary does not need to be passed. Routes to destinations out of the area always start with the instruction ‘‘Go to the border router.’’ This kind of area is called a **stub area**.

The last kind of router is the **AS boundary router**. It injects routes to external destinations on other ASes into the area. The external routes then appear as destinations that can be reached via the AS boundary router with some cost. An external route can be injected at one or more AS boundary routers. The relationship between ASes, areas, and the various kinds of routers is shown in Fig. 5-65. One router may play multiple roles, for example, a border router is also a backbone router.

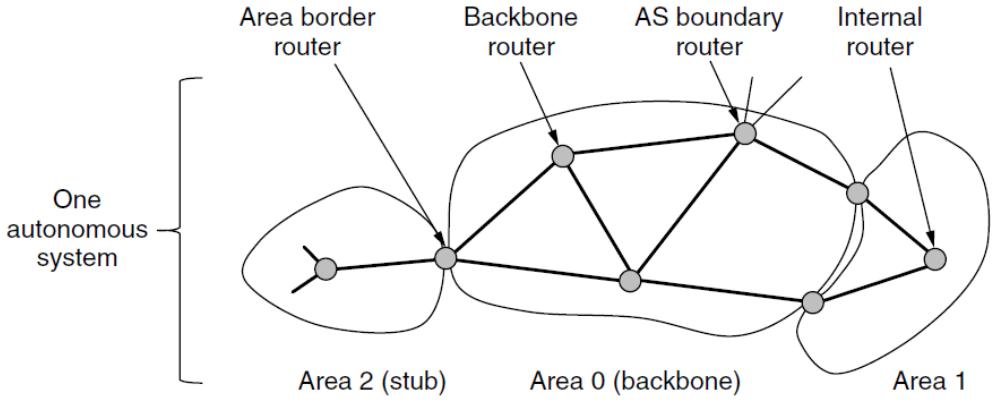


Figure 5-65. The relation between ASes, backbones, and areas in OSPF.

During normal operation, each router within an area has the same link state database and runs the same shortest path algorithm. Its main job is to calculate the shortest path from itself to every other router and network in the entire AS. An area border router needs the databases for all the areas to which it is connected and must run the shortest path algorithm for each area separately.

For a source and destination in the same area, the best intra-area route (that lies wholly within the area) is chosen. For a source and destination in different areas, the inter-area route must go from the source to the backbone, across the backbone to the destination area, and then to the destination. This algorithm forces a star configuration on OSPF, with the backbone being the hub and the other areas being spokes. Because the route with the lowest cost is chosen, routers in different parts of the network may use different area border routers to enter the backbone and destination area. Packets are routed from source to destination ‘‘as is.’’ They are not encapsulated or tunneled (unless going to an area whose only connection to the backbone is a tunnel). Also, routes to external destinations may include the external cost from the AS boundary router over the external path, if desired, or just the cost internal to the AS. When a router boots, it sends HELLO messages on all of its point-to-point lines and multicasts them on LANs to the group consisting of all the other routers.

From the responses, each router learns who its neighbours are. Routers on the same LAN are all neighbours. OSPF works by exchanging information between adjacent routers, which is not the same as between neighbouring routers. In particular, it is inefficient to have every router on a LAN talk to every other router on the LAN. To avoid this situation, one router is elected as the **designated router**. It is said to be **adjacent** to all the other routers on its LAN, and exchanges information with them. In effect, it is acting as the single node that represents the LAN. Neighbouring routers that are not adjacent do not exchange information with each other. A backup designated router is always kept up to date to ease the transition should the primary designated router crash and need to be replaced immediately.

During normal operation, each router periodically floods LINK STATE UPDATE messages to each of its adjacent routers. These messages gives its state and provide the costs used in the topological database. The flooding messages are acknowledged, to make them reliable. Each message has a sequence number,

so a router can see whether an incoming LINK STATE UPDATE is older or newer than what it currently has. Routers also send these messages when a link goes up or down or its cost changes.

DATABASE DESCRIPTION messages give the sequence numbers of all the link state entries currently held by the sender. By comparing its own values with those of the sender, the receiver can determine who has the most recent values. These messages are used when a link is brought up. Either partner can request link state information from the other one by using LINK STATE REQUEST messages. The result of this algorithm is that each pair of adjacent routers checks to see who has the most recent data, and new information is spread throughout the area this way. All these messages are sent directly in IP packets. The five kinds of messages are summarized in Fig. 5-66.

Message type	Description
Hello	Used to discover who the neighbors are
Link state update	Provides the sender's costs to its neighbors
Link state ack	Acknowledges link state update
Database description	Announces which updates the sender has
Link state request	Requests information from the partner

Figure 5-66. The five types of OSPF messages.

Using flooding, each router informs all the other routers in its area of its links to other routers and networks and the cost of these links. This information allows each router to construct the graph for its area(s) and compute the shortest paths. The backbone area does this work, too. In addition, the backbone routers accept information from the area border routers in order to compute the best route from each backbone router to every other router. This information is propagated back to the area border routers, which advertise it within their areas. Using this information, internal routers can select the best route to a destination outside their area, including the best exit router to the backbone.

main types of OSPF messages:

1. **Hello:** OSPF routers use Hello packets to discover and establish OSPF neighbor relationships. Hello packets contain information about the router's OSPF interface, such as the router's ID, area ID, and authentication type.
2. **Database Description (DBD):** DBD packets are used to exchange information about the OSPF link-state database. Each DBD packet contains a list of Link State Advertisements (LSAs) that the sending router has in its database. The receiving router can then compare this list with its own database to determine which LSAs it needs to request.

Link State Advertisements (LSAs) are packets of information used in link-state routing protocols to exchange and maintain information about network topology.

3. **Link State Request (LSR):** If a router determines that it is missing certain LSAs(Link State Advertisements) from its database based on the received DBD packets, it sends Link State Request packets to its neighbors requesting the missing LSAs.
4. **Link State Update (LSU):** In response to a Link State Request, routers send Link State Update packets containing the requested LSAs. These packets carry the actual LSAs that the requesting router needs to complete its OSPF database.

5. **Link State Acknowledgment (LSAck):** When a router receives Link State Update packets, it sends Link State Acknowledgment packets to confirm the receipt of the LSAs. These acknowledgments ensure reliable data transmission and help maintain database consistency.

These OSPF message types work together to establish and maintain accurate routing information among OSPF routers in a network. OSPF operates as a link-state routing protocol, which means that routers exchange information about the state of their links (interfaces) and use this information to calculate the best paths for routing packets through the network.

4. BGP- Exterior Gateway Routing Protocol

Within a single AS, OSPF and IS-IS (Intermediate System to Intermediate System) are the protocols that are commonly used. Between ASes, a different protocol, called BGP (Border Gateway Protocol), is used. A different protocol is needed because the goals of an intradomain protocol and an interdomain protocol are not the same. All an intradomain protocol has to do is move packets as efficiently as possible from the source to the destination. It does not have to worry about politics.

In contrast, interdomain routing protocols have to worry about politics a great deal (Metz, 2001). For example, a corporate AS might want the ability to send packets to any Internet site and receive packets from any Internet site. However, it might be unwilling to carry transit packets originating in a foreign AS and ending in a different foreign AS, even if its own AS is on the shortest path between the two foreign ASes (‘‘That’s their problem, not ours’’). On the other hand, it might be willing to carry transit traffic for its neighbors, or even for specific other ASes that paid it for this service. Telephone companies, for example, might be happy to act as carriers for their customers, but not for others. Exterior gateway protocols in general, and BGP in particular, have been designed to allow many kinds of routing policies to be enforced in the interAS traffic.

Typical policies involve political, security, or economic considerations. A few examples of possible routing constraints are:

1. Do not carry commercial traffic on the educational network.
2. Never send traffic from the Pentagon on a route through Iraq.
3. Use TeliaSonera instead of Verizon because it is cheaper.
4. Don’t use AT&T in Australia because performance is poor.
5. Traffic starting or ending at Apple should not transit Google.

Aspect	Interdomain Routing Protocols	Intradomain Routing Protocols
Scope	Operate between different Autonomous Systems (AS)	Operate within a single Autonomous System (AS)
Main Protocols	BGP (Border Gateway Protocol)	OSPF (Open Shortest Path First), IS-IS, RIP, etc.
Objective	Exchange routing information between ASes	Establish and maintain routing within an AS
Policy	Emphasize policy-based routing decisions	Focus on efficient path selection and metrics
Path Selection	Based on complex policies, preferences, and AS path	Based on shortest path algorithms and metrics
Metrics	AS path length, attributes, and policies	Link cost, bandwidth, delay, etc.
Convergence Speed	Typically slower due to complex decision processes	Generally faster within a single AS
Topology Knowledge	Often limited to neighboring ASes	Comprehensive knowledge of internal topology
Addressing	Often uses Classless Inter-Domain Routing (CIDR)	Uses IP addresses within the same address space
Examples	BGP-4, BGP-5, BGP-LS, BGP-MP, BGP-LS	OSPFv2, OSPFv3, IS-IS, RIP, EIGRP, etc.
Security Emphasis	Focuses on secure inter-AS communication	Security is important but within the AS
Typical Deployments	Internet backbone and ISPs	Corporate networks, data centers, campuses
Scale	Must handle Internet-scale routing tables	Typically smaller-scale, AS-specific routing
Typical Issues	Policy conflicts, route hijacking, security risks	Scalability, convergence time, routing loops

A routing policy is implemented by deciding what traffic can flow over which of the links between ASes. One common policy is that a customer ISP pays another provider ISP to deliver packets to any other destination on the Internet and receive packets sent from any other destination. The customer ISP is said to buy **transit service** from the provider ISP. This is just like a customer at home buying Internet access service from an ISP. To make it work, the provider should advertise routes to all destinations on the Internet to the customer over the link that connects them. In this way, the customer will have a route to use to send packets anywhere. Conversely, the customer should advertise routes only to the destinations

on its network to the provider. This will let the provider send traffic to the customer only for those addresses; the customer does not want to handle traffic intended for other destinations.

Internet Exchange Points (IXPs) are crucial elements in the global internet infrastructure that facilitate the exchange of internet traffic between different Internet Service Providers (ISPs) and networks. IXPs play a vital role in improving the efficiency of internet traffic exchange and reducing the need for data to travel long distances through external networks.

Internet Service Providers (ISPs) are companies that provide internet access and related services to individuals, businesses, and other organizations. ISPs play a crucial role in connecting users to the global internet infrastructure.eg: Airtel, Vodafone etc

Example of transit service in Fig. 5-67. There are four ASes that are connected. The connection is often made with a link at **IXPs** (**Internet eXchange Points**), facilities to which many ISPs have a link for the purpose of connecting with other ISPs. *AS2*, *AS3*, and *AS4* are customers of *AS1*. They buy transit service from it. Thus, when source *A* sends to destination *C*, the packets travel from *AS2* to *AS1* and finally to *AS4*. The **routing advertisements travel in the opposite direction to the packets**. *AS4* advertises *C* as a destination to its transit provider, *AS1*, to let sources reach *C* via *AS1*. Later, *AS1* advertises a route to *C* to its other customers, including *AS2*, to let the customers know that they can send traffic to *C* via *AS1*.

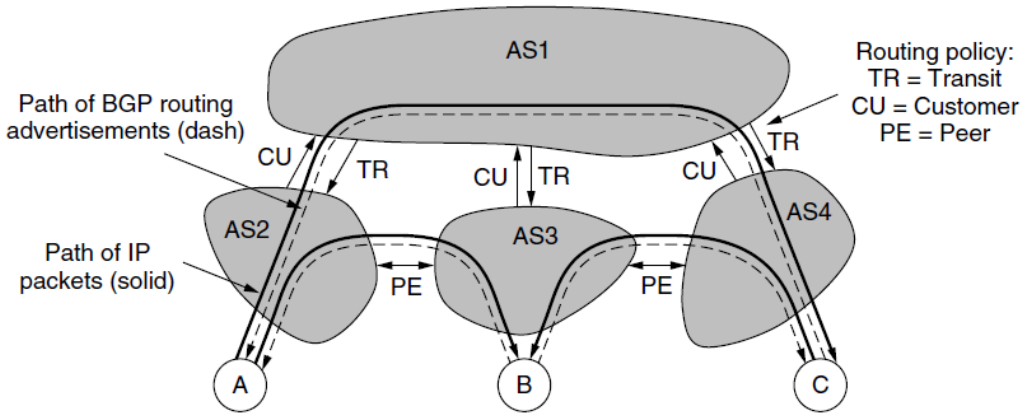


Figure 5-67. Routing policies between four autonomous systems.

In Fig. 5-67, all of the other ASes buy transit service from *AS1*. This provides them with connectivity so they can interact with any host on the Internet. However, they have to pay for this privilege. Suppose that *AS2* and *AS3* exchange a lot of traffic. Given that their networks are connected already, if they want to, they can use a different policy—they can send traffic directly to each other for free. This will reduce the amount of traffic they must have *AS1* deliver on their behalf, and hopefully it will reduce their bills. This policy is called **peering.** [Peering in the context of Border Gateway Protocol (BGP) refers to the establishment of direct interconnections between two or more Autonomous Systems (ASes) for the purpose of exchanging routing information and internet traffic. Peering agreements allow networks to exchange data more efficiently and directly, reducing the

need to route traffic through third-party ISPs]. To implement peering, two ASes send routing advertisements to each other for the addresses that reside in their networks. Doing so makes it possible for *AS2* to send *AS3* packets from *A* destined to *B* and vice versa. However, note that peering is not transitive. In Fig. 5-67, *AS3* and *AS4* also peer with each other. This peering allows traffic from *C* destined for *B* to be sent directly to *AS4*. What happens if *C* sends a packet to *A*? *AS3* is only advertising a route to *B* to *AS4*. It is not advertising a route to *A*. The consequence is that traffic will not pass from *AS4* to *AS3* to *AS2*, even though a physical path exists. This restriction is exactly what *AS3* wants. It peers with *AS4* to exchange traffic, but does not want to carry traffic from *AS4* to other parts of the Internet since it is not being paid to so do. Instead, *AS4* gets transit service from *AS1*. Thus, it is *AS1* who will carry the packet from *C* to *A*.

Transit service refers to the provision of network connectivity and routing services by an Internet Service Provider (ISP) to other networks, organizations, or individuals that do not have direct peering relationships with all other networks on the Internet. Transit service enables those networks to access the entire global Internet and exchange data with networks that they do not have a direct connection to.

Now that we know about transit and peering, we can also see that *A*, *B*, and *C* have transit arrangements. For example, *A* must buy Internet access from *AS2*. *A* might be a single home computer or a company network with many LANs. However, it does not need to run BGP because it is a **stub network** that is connected to the rest of the Internet by only one link. So the only place for it to send packets destined outside of the network is over the link to *AS2*. There is nowhere else to go. This path can be arranged simply by setting up a default route. For this reason, we have not shown *A*, *B*, and *C* as ASes that participate in interdomain routing. On the other hand, some company networks are connected to multiple ISPs.

This technique is used to improve reliability, since if the path through one ISP fails, the company can use the path via the other ISP. This technique is called **multihoming**. In this case, the company network is likely to run an interdomain routing protocol (e.g., BGP) to tell other ASes which addresses should be reached via which ISP links.

Many variations on these transit and peering policies are possible, but they already illustrate how business relationships and control over where route advertisements go can implement different kinds of policies. Now we will consider in more detail how routers running BGP advertise routes to each other and select paths over which to forward packets.

BGP is a form of distance vector protocol, but it is quite unlike intradomain distance vector protocols such as RIP. We have already seen that policy, instead of minimum distance, is used to pick which routes to use. Another large difference is that instead of maintaining just the cost of the route to each destination, each BGP router keeps track of the path used. This approach is called a **path vector protocol**. The path consists of the next hop router (which may be on the other side of the ISP, not adjacent) and the sequence of ASes, or **AS path**, that the route has followed (given in reverse order). Finally, pairs of BGP routers

communicate with each other by establishing TCP connections. Operating this way provides reliable communication and also hides all the details of the network being passed through. An example of how BGP routes are advertised is shown in Fig. 5-68. There are three ASes and the middle one is providing transit to the left and right ISPs. A route advertisement to prefix *C* starts in *AS3*. When it is propagated across the link to *R2c* at the top of the figure, it has the AS path of simply *AS3* and the next hop router of *R3a*. At the bottom, it has the same AS path but a different next hop because it came across a different link. This advertisement continues to propagate and crosses the boundary into *AS1*. At router *R1a*, at the top of the figure, the AS path is *AS2*, *AS3* and the next hop is *R2a*.

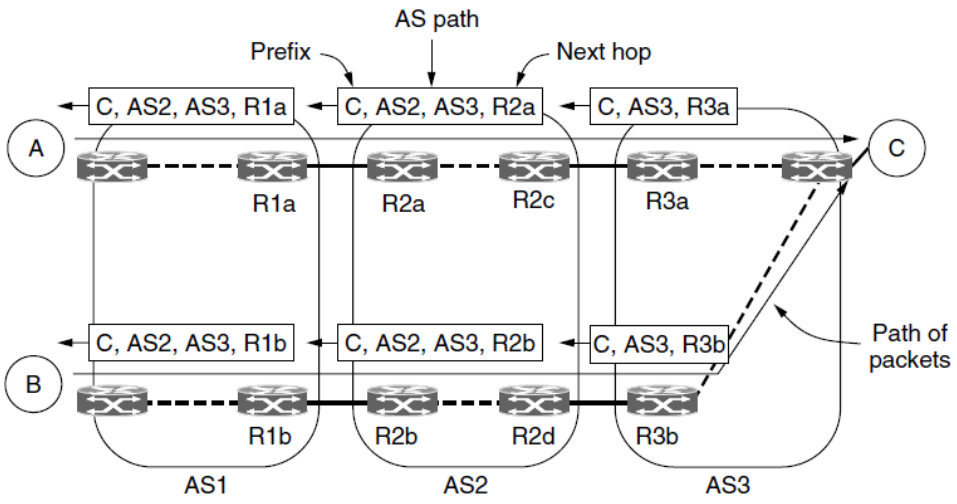


Figure 5-68. Propagation of BGP route advertisements.

Carrying the complete path with the route makes it easy for the receiving router to detect and break routing loops. The rule is that each router that sends a route outside of the AS prepends its own AS number to the route. (This is why the list is in reverse order.) When a router receives a route, it checks to see if its own AS number is already in the AS path. If it is, a loop has been detected and the advertisement is discarded. However, and somewhat ironically, it was realized in the late 1990s that despite this precaution BGP suffers from a version of the count-to-infinity problem (Labovitz et al., 2001). There are no long-lived loops, but routes can sometimes be slow to converge and have transient loops.

Giving a list of ASes is a very coarse way to specify a path. An AS might be a small company, or an international backbone network. There is no way of telling from the route. BGP does not even try because different ASes may use different intradomain protocols whose costs cannot be compared. Even if they could be compared, an AS may not want to reveal its internal metrics. This is one of the ways that interdomain routing protocols differ from intradomain protocols. We still need some way to propagate BGP routes from one side of the ISP to the other, so they can be sent on to the next ISP. This task could be handled by the intradomain protocol, but because BGP is very good at scaling to large networks, a variant of BGP is often used. It is called **iBGP** (**internal BGP**) to distinguish it from the regular use of BGP as **eBGP** (**external BGP**). The rule for propagating routes inside an ISP is that every router at the boundary of the ISP learns of all the routes seen by all the other boundary routers, for

consistency. If one boundary router on the ISP learns of a prefix to IP 128.208.0.0/16, all the other routers will learn of this prefix. The prefix will then be reachable from all parts of the ISP, no matter how packets enter the ISP from other ASes.

Each BGP router may learn a route for a given destination from the router it is connected to in the next ISP and from all of the other boundary routers (which have heard different routes from the routers they are connected to in other ISPs). Each router must decide which route in this set of routes is the best one to use. Ultimately the answer is that it is up to the ISP to write some policy to pick the preferred route. However, this explanation is very general and not at all satisfying, so we can at least describe some common strategies. The first strategy is that routes via peered networks are chosen in preference to routes via transit providers. The former are free; the latter cost money. A similar strategy is that customer routes are given the highest preference. It is only good business to send traffic directly to the paying customers. A different kind of strategy is the default rule that shorter AS paths are better.

This is debatable given that an AS could be a network of any size, so a path through three small ASes could actually be shorter than a path through one big AS. However, shorter tends to be better on average, and this rule is a common tiebreaker.

The final strategy is to prefer the route that has the lowest cost within the ISP. This is the strategy implemented in Fig. 5-68. Packets sent from *A* to *C* exit *AS1* at the top router, *R1a*. Packets sent from *B* exit via the bottom router, *R1b*. The reason is that both *A* and *B* are taking the lowest-cost path or quickest route out of *AS1*. Because they are located in different parts of the ISP, the quickest exit for each one is different. The same thing happens as the packets pass through *AS2*. On the last leg, *AS3* has to carry the packet from *B* through its own network. This strategy is known as **early exit** or **hot-potato routing**. It has the curious side effect of tending to make routes asymmetric. For example, consider the path taken when *C* sends a packet back to *B*. The packet will exit *AS3* quickly, at the top router, to avoid wasting its resources. Similarly, it will stay at the top when *AS2* passes it to *AS1* as quickly as possible. Then the packet will have a longer journey in *AS1*. This is a mirror image of the path taken from *B* to *C*.

The above discussion should make clear that each BGP router chooses its own best route from the known possibilities. It is not the case, as might naively be expected, that BGP chooses a path to follow at the AS level and OSPF chooses paths within each of the ASes. BGP and the interior gateway protocol are integrated much more deeply. This means that, for example, BGP can find the best exit point from one ISP to the next and this point will vary across the ISP, as in the case of the hot-potato policy. It also means that BGP routers in different parts of one AS may choose different AS paths to reach the same destination. Care must be exercised by the ISP to configure all of the BGP routers to make compatible choices given all of this freedom, but this can be done in practice.

5. IPv6

IPv6 (Internet Protocol version 6) is the next generation of the Internet Protocol, designed to replace the widely used IPv4 due to its limitations in address space and other technical constraints. IPv6 introduces several improvements and enhancements over IPv4.

IPv6 is a 128-bit addressing scheme that was developed to address the address exhaustion problem of IPv4, which uses 32-bit addresses. With 128 bits, IPv6 allows for an enormous number of unique addresses, which is necessary to accommodate the ever-growing number of devices connected to the internet. The increased address space is not the only improvement; IPv6 also includes features to enhance security, simplify network configuration, and improve overall network efficiency.

Advantages of IPv6:

1. **Vast Address Space**: The most significant advantage of IPv6 is its massive address space, which ensures that there are enough unique IP addresses for all devices, even as the number of connected devices continues to grow.
2. **Improved Network Efficiency**: IPv6's simplified header structure and more efficient routing enable faster and more streamlined data transmission across networks.
3. **Enhanced Security**: The inclusion of IPsec as a standard feature in IPv6 provides improved security for communications, making it more challenging for unauthorized parties to intercept or tamper with data.
4. **Autoconfiguration**: IPv6's autoconfiguration feature simplifies network setup and reduces the need for manual configuration or reliance on DHCP servers.
5. **Support for Emerging Technologies**: IPv6's design takes into account the evolving technology landscape, making it well-suited for IoT devices, mobile networks, and other emerging technologies.
6. **Multicast Efficiency**: IPv6 improves multicast support, enabling efficient content distribution to multiple recipients.

Disadvantages of IPv6:

1. **Transition Complexity**: Transitioning from IPv4 to IPv6 can be complex and challenging, requiring updates to network infrastructure, devices, and software.
2. **Compatibility Issues**: While IPv6 is designed for backward compatibility, some older devices, applications, and network equipment might not fully support IPv6, potentially causing interoperability issues.

3. **Lack of Immediate Incentive**: As IPv4 addresses are still in use and available through techniques like NAT (Network Address Translation), some organizations might not see an immediate need to transition to IPv6.
4. **Learning Curve**: Network administrators and IT professionals may need to learn new concepts and practices associated with IPv6, which could involve a learning curve.
5. **Security Challenges**: While IPv6 includes enhanced security features, its adoption could introduce new security challenges and vulnerabilities that need to be properly managed.

IPv6 offers substantial benefits in terms of addressing space, network efficiency, security, and support for emerging technologies. However, its adoption comes with challenges related to transitioning, compatibility, and network management. As the internet continues to grow and evolve, IPv6 is becoming increasingly essential to ensure the continued expansion of the digital landscape.

The Main IPv6 Header

IPv4 Header vs IPv6 Header Explained - YouTube

The IPv6 header is shown in Fig. 5-56.

Version (4-bits): indicates the version of the IP protocol being used. For IPv6, this field is set to the value 6 (0110), which distinguishes it from its predecessor.

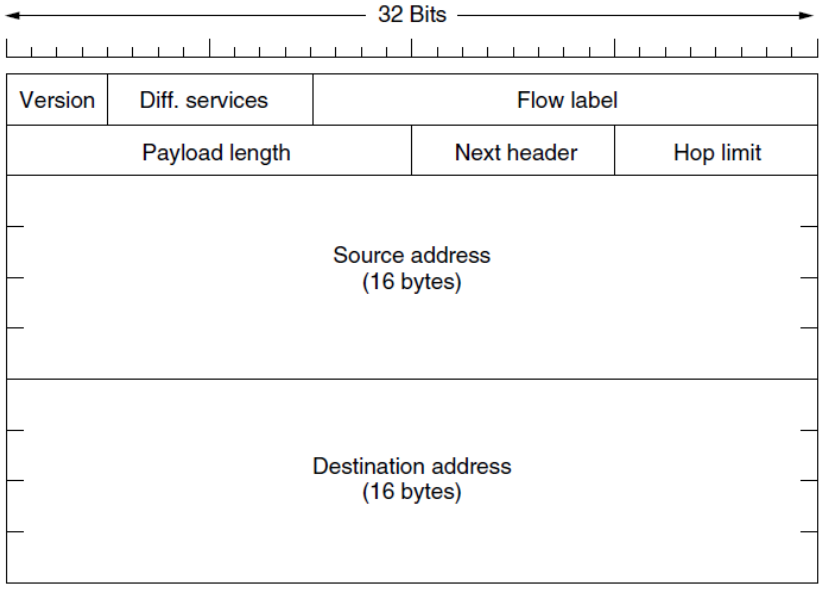


Figure 5-56. The IPv6 fixed header (required).

The Differentiated services field / Traffic class (8-bits): indicates class or priority of IPv6 packet which is similar to *Service Field* in IPv4 packet. It helps routers to handle the traffic based on the priority of the packet. If congestion occurs on the router then packets with the least priority will be discarded. As of now, only 4-bits are being used (and the remaining bits are under research), in which 0 to 7 are assigned to Congestion controlled traffic and 8 to 15 are assigned to Uncontrolled traffic.

Uncontrolled data traffic is mainly used for Audio/Video data. So we give higher priority to Uncontrolled data traffic. The source node is allowed to set the priorities but on the way, routers can change it. Therefore, the destination should not expect the same priority which was set by the source node.

Priority assignment of Congestion controlled traffic :

Priority	Meaning
0	No Specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Attended bulk data traffic
5	Reserved
6	Interactive traffic
7	Control traffic

Flow Label (20-bits): Flow Label field is used by a source to label the packets belonging to the same flow in order to request special handling by intermediate IPv6 routers, such as non-default quality of service or real-time service. In order to distinguish the flow, an intermediate router can use the source address, a destination address, and flow label of the packets. Between a source and destination, multiple flows may exist because many processes might be running at the same time. Routers or Host that does not support the functionality of flow label field and for default router handling, flow label field is set to 0. While setting up the flow label, the source is also supposed to specify the lifetime of the flow.

Flow Label provides support for flow labeling, which is a mechanism designed to help with the identification, labeling, and handling of packets belonging to the same flow or traffic stream. A flow is typically defined as a sequence of packets that share certain characteristics, such as source and destination addresses, transport protocol, and port numbers.

The main purpose of the Flow Label field is to assist routers and switches in implementing Quality of Service (QoS) features and traffic management, where specific treatment can be given to packets belonging to the same flow. The Flow Label value is chosen by the source node and remains consistent for the duration of the flow. Routers and intermediate devices can use this label to make forwarding decisions, prioritize traffic, and ensure that packets within the same flow receive consistent treatment throughout their journey across the network.

Payload Length (16-bits): It indicates the total size of the payload which tells routers about the amount of information a particular packet contains in its payload. The payload Length field includes extension headers(if any) and an upper-layer packet. In case the length of the payload is greater than 65,535 bytes (payload up to 65,535 bytes can be indicated with 16-bits), then the payload length field will be set to 0 and the jumbo payload option is used in the Hop-by-Hop options extension header.

Next Header (8-bits): It tells the type of Extension Header which are additionally used with base header to send more data or information's. The "Next Header" field is crucial for proper packet processing, as it tells the receiving node how to interpret and process the rest of the packet. It can point to standard transport layer protocols like TCP (6), UDP (17), ICMPv6 (58), or to various IPv6 extension headers such as Routing Header (43), Fragment Header (44), and others. Some extension headers are given below.

Extension Header	Next Header Vaule	Description
Hop-by-Hop Options header	0	Read by all devices in transit network
Routing Header	43	Contains method to support making routing decision
Fragment Header	44	contains parameters of datagram fragmentations
Destination Options Header	60	Read by destination Device
Authentication Header	51	Information Regarding Security
Encapsulating security payload Header	50	encryption informations

Next Header indicates the type of extension header(if present) immediately following the IPv6 header. Whereas in some cases it indicates the protocols contained within upper-layer packets, such as TCP, UDP. If extension header is used along with payload, then corresponding bits value is presented in this filed. It mean if Routing header is used as extension header then 43(in bits) is represented in Next header (8bit) field.

Note: Extension headers are optional, and are used if needed.

1. **Hop-by-Hop Options Header:** This header carries options that must be examined by every router along the path of the packet. It is often used for features such as router alert, multicast listener discovery, and flow label.
2. **Routing Header:** This header is used to specify the route that the packet should take through the network. It can have multiple types, including Type 0 for the strict source route and Type 2 for the loose source route.
3. **Fragmentation Header:** Unlike IPv4, where fragmentation can be done by routers, IPv6 requires the sender to fragment packets before transmission if the Maximum Transmission Unit (MTU) of the next hop is smaller. The Fragmentation Header is used to carry information about how to reassemble the original packet.
4. **Authentication Header (AH):** This header provides data integrity and authentication to the packet. It ensures that the contents of the packet have not been modified in transit and that the sender of the packet is genuine.
5. **Encapsulating Security Payload (ESP):** This header is used to provide encryption, confidentiality, and authentication to the packet's payload. It protects the actual data being transmitted.
6. **Destination Options Header:** Similar to the Hop-by-Hop Options Header, this header carries options, but these options are meant to be examined only by the destination node.

7. **Mobility Headers:** These headers are used for Mobile IPv6, a protocol that allows mobile devices to move between different networks while maintaining their IP connectivity.

Hop Limit (8-bits): This field is used to stop packet to loop in the network infinitely. This is same as TTL in IPv4. The Hop-Limit field value is decremented by 1 as it passes a link (i.e. router). When the value of Hop-limit field reaches 0 the packet is discarded. This field also sets an upper threshold on the maximum number of links between two nodes of the IPv6 protocol. It allows a maximum of 255 hops between the nodes, and anything afterward will be discarded.

Source Address (128-bits): Source Address is the 128-bit IPv6 address of the original source of the packet.

Destination Address (128-bits): The destination Address field indicates the IPv6 address of the final destination(in most cases). All the intermediate nodes can use this information in order to correctly route the packet.

Extension Headers: Some of the missing IPv4 fields are occasionally still needed, so IPv6 introduces the concept of (optional) **extension headers**. These headers can be supplied to provide extra information, but encoded in an efficient way. Six kinds of extension headers are defined at present, as listed in Fig. 5-57. Each one is optional, but if more than one is present they must appear directly after the fixed header, and preferably in the order listed. The hop-by-hop extension header for large datagrams are called jumbograms.

Extension header	Description
Hop-by-hop options	Miscellaneous information for routers
Destination options	Additional information for the destination
Routing	Loose list of routers to visit
Fragmentation	Management of datagram fragments
Authentication	Verification of the sender's identity
Encrypted security payload	Information about the encrypted contents

Figure 5-57. IPv6 extension headers.

Conventions :

1. Any extension header can appear at most once except Destination Header because Destination Header is present two times in the above list itself.
2. If Destination Header is present before Routing Header then it will be examined by all intermediate nodes specified in the routing header.
3. If Destination Header is present just above the Upper layer then it will be examined only by the Destination node.


IPv4 Header (20-Byte)
Version
IHL
Type of Service
Total Length
Identification
Flags
Fragment Offset
Time to Live
Protocol
Header Checksum
Source Address
Destination Address
Options
Padding
Legend
Field's Name Kept from IPv4 to IPv6
Fields Not Kept in IPv6
Name and Position Changed in IPv6
New Field in IPv6
IPv6 Header (40-Byte)
Version
Traffic Class
Flow Label
Payload Length
Next Header
Hop Limit
Source Address
Destination Address