Module – 3

The Network layer

4.3 What's Inside a Router?

Forwarding function—the actual transfer of packets from a router's incoming links to the appropriate outgoing links at that router. In Computer networking, *forwarding* and *switching* are often used interchangeably by researchers and practitioners;

A high-level view of a generic router architecture is shown in Figure 4.6. Four router components can be identified:

- ***Input ports***. An input port performs several key functions. It performs the physical layer function of terminating an incoming physical link at a router; this is shown in the leftmost box of the input port and the rightmost box of the output port in Figure 4.6.
  - An input port also performs link-layer functions needed to interoperate with the link layer at the other side of the incoming link; this is represented by the middle boxes in the input and output ports.
  - Perhaps most crucially, the lookup function is also performed at the input port; this will occur in the rightmost box of the input port.
  - It is here that the forwarding table is consulted to determine the router output port to which an arriving packet will be forwarded via the switching fabric.
  - Control packets (for example, packets carrying routing protocol information) are forwarded from an input port to the routing processor. Note that the term *port* here— referring to the physical input and output router interfaces—is distinctly different from the software ports associated with network applications.

- ***Switching fabric.*** The switching fabric connects the router's input ports to its output ports. This switching fabric is completely contained within the router— a network inside of a network router!

- ***Output ports.*** Output port stores packets received from the switching fabric and transmit these packets on the outgoing link by performing the necessary link-layer and physical-layer functions.
  - When a link is bidirectional (that is carries traffic in both directions), an output port will typically be paired with the input port for that link on the same line card (a printed circuit board containing one or more input ports, which is connected to the switching fabric).

- ***Routing processor.***

- The routing processor executes the routing protocols, maintains routing tables and attached link state information, and computes the forwarding table for the router.
- It also performs the network management functions.
- A router's input ports, output ports, and switching fabric together implement the forwarding function and are almost always implemented in hardware, as shown in Figure 4.6.
- These forwarding functions are sometimes collectively referred to as the **router forwarding plane**.

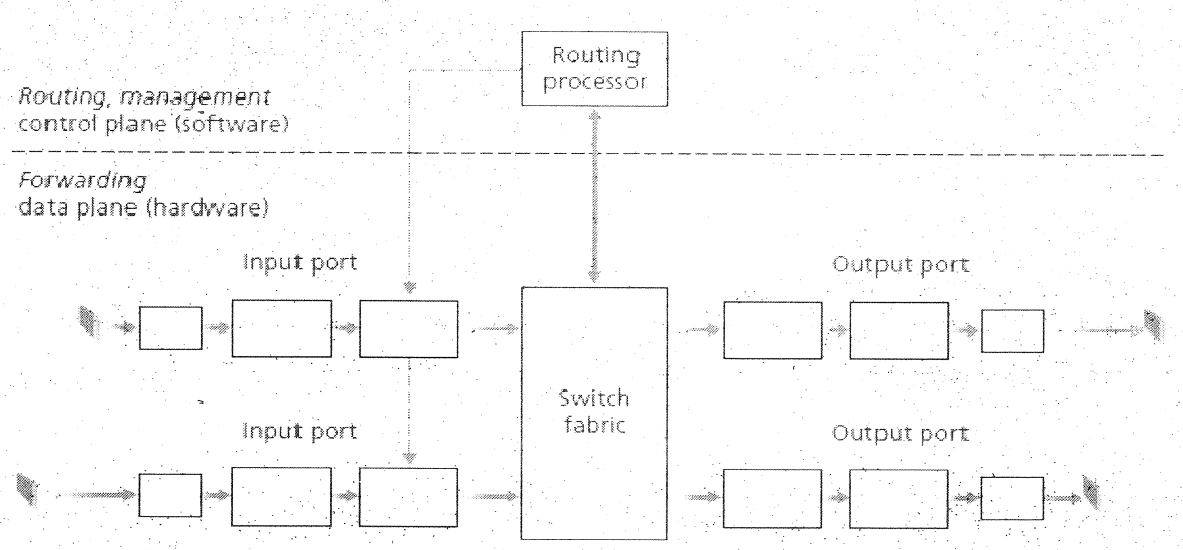


Figure 4.6 ♦ Router architecture

Input port processing:

- A more detailed view of input processing is given in Figure 4.7. As discussed above, the input port's line termination function and link-layer processing implement the physical and link layers for that individual input link.
- The lookup performed in the input port is central to the router's operation—it is here that the router uses the forwarding table to look up the output port to which an arriving packet will be forwarded via the switching fabric.

- The forwarding table is computed and updated by the routing processor, with a shadow copy typically stored at each input port. The forwarding table is copied from the routing processor to the line cards over a separate bus (e.g., a PCI bus) indicated by the dashed line from the routing processor to the input line cards in Figure 4.6.
- With a shadow copy, forwarding decisions can be made locally, at each input port, without invoking the centralized routing processor on a per-packet basis and thus avoiding a centralized processing bottleneck.
- Given the existence of a forwarding table, lookup is conceptually simple—we just search through the forwarding table looking for the longest prefix match, as described.

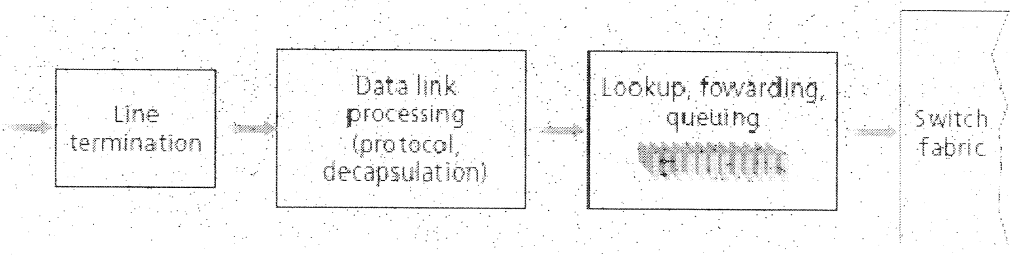


Figure 4.7 ♦ Input port processing

- Once a packet's output port has been determined via the lookup, the packet can be sent into the switching fabric.
- In some designs, a packet may be temporarily blocked from entering the switching fabric if packets from other input ports are currently using the fabric.
- A blocked packet will be queued at the input port and then scheduled to cross the fabric at a later point in time.
- We'll take a closer look at the blocking, queuing, and scheduling of packets (at both input ports and output ports) in Section 4.3.4.
- Although "lookup" is arguably the most important action in input port processing, many other actions must be taken: (1) physical- and link-layer processing must occur, as discussed above; (2) the packet's version number, checksum and time-to-live field and the latter two fields rewritten; and (3) counters used for network management (such as the number of IP datagrams received) must be updated.

4.3.2 Switching

The switching fabric is at the very heart of a router, as it is through this fabric that the packets are actually switched (that is, forwarded) from an input port to an output port. Switching can be accomplished in a number of ways, as shown in Figure 4.8:

• ***Switching via memory***. The simplest, earliest routers were traditional computers, with switching between input and output ports being done under direct control of the CPU (routing processor). Input and output ports functioned as traditional I/O devices in a traditional operating system.

- An input port with an arriving packet first signaled the routing processor via an interrupt. The packet was then copied from the input port into processor memory. The routing processor then extracted the destination address from the header, looked up the appropriate output port in the forwarding table, and copied the packet to the output port's buffers.

- In this scenario, if the memory bandwidth is such that *B* packets per second can be written into, or read from, memory, then the overall forwarding throughput (the total rate at which packets are transferred from input ports to output ports) must be less than *B*/2. Note also that two packets cannot be forwarded at the same time, even if they have different destination ports, since only one memory read/write over the shared system bus can be done at a time.

Switching via a bus. In this approach, an input port transfers a packet directly to the output port over a shared bus, without intervention by the routing processor. This is typically done by having the input port pre-pend a switch-internal label (header) to the packet indicating the local output port to which this packet is being transferred and transmitting the packet onto the bus.

- The packet is received by all output ports, but only the port that matches the label will keep the packet.
- The label is then removed at the output port, as this label is only used within the switch to cross the bus.
- If multiple packets arrive to the router at the same time, each at a different input port, all but one must wait since only one packet can cross the bus at a time.
- Because every packet must cross the single bus, the switching speed of the router is limited to the bus speed; in our roundabout analogy, this is as if the roundabout could only contain one car at a time.

Switching via an interconnection network. One way to overcome the bandwidth limitation of a single, shared bus is to use a more sophisticated interconnection network, such as those that have been used in the past to interconnect processors in multiprocessor computer architecture.

- A crossbar switch is an interconnection network consisting of 2*N* buses that connect *N* input ports to *N* output ports, as shown in Figure 4.8. Each vertical bus intersects each horizontal bus at a crosspoint, which can be opened or closed at any time by the switch fabric controller (whose logic is part of the switching fabric itself).
- When a packet arrives from port A and needs to be forwarded to port Y, the switch controller closes the crosspoint at the intersection of busses Aand Y, and port Athen sends the packet onto its bus, which is picked up (only) by bus Y. Note that a packet from port B can be forwarded to port X at the same time, since the A-to-Y and B-to-X packets use different input and output busses. Thus, unlike the previous two switching approaches, crossbar networks are capable of forwarding multiple packets in parallel.
- However, if two packets from two different input ports are destined to the same output port, then one will have to wait at the input, since only one packet can be sent over any given bus at a time.

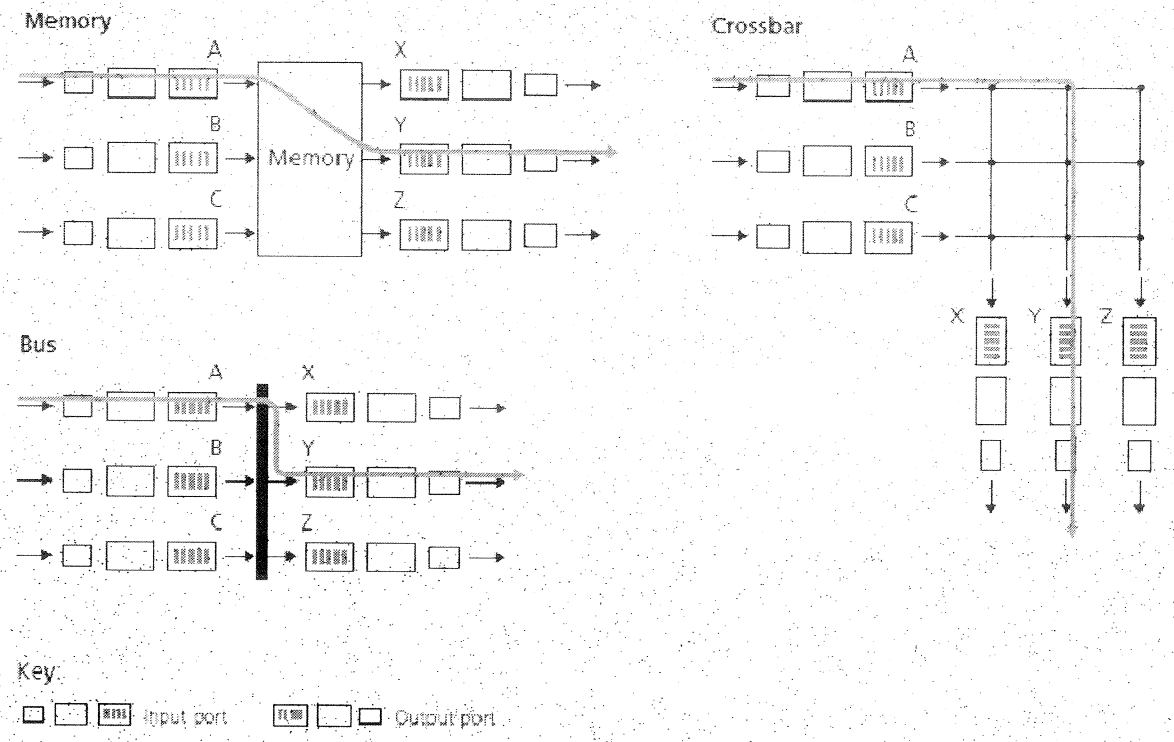


Figure 4.8 ♦ Three switching techniques

4.3.3 Output Processing

Output port processing, shown in Figure 4.9, takes packets that have been stored in the output port's memory and transmits them over the output link. This includes selecting and de-queueing packets for transmission, and performing the needed link layer and physical-layer transmission functions.

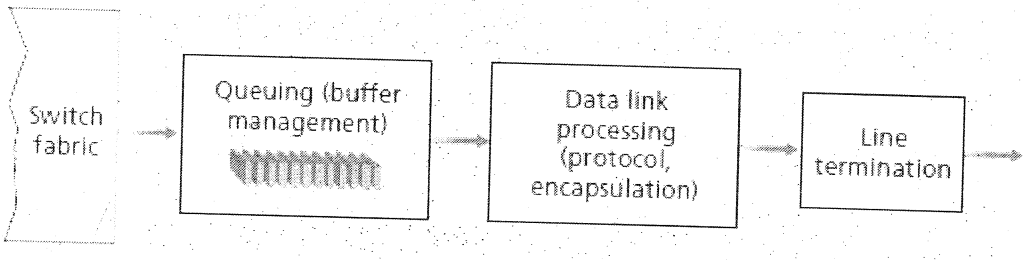


Figure 4.9 ♦ Output port processing

4.3.4 Where Does Queueing Occur?

- packet queues may form at both the input ports *and* the output ports.
- The location and extent of queuing (either at the input port queues or the output port queues) will depend on the traffic load, the relative speed of the switching fabric, and the line speed.
- Suppose that the input and output line speeds (transmission rates) all have an identical transmission rate of R_{line} packets per second, and that there are N input ports and N output ports.
- let's assume that all packets have the same fixed length, and the packets arrive to input ports in a synchronous manner.
- That is, the time to send a packet on any link is equal to the time to receive a packet on any link, and during such an interval of time, either zero or one packet can arrive on an input link.
- Define the switching fabric transfer rate R_{switch} as the rate at which packets can be moved from input port to output port.
- If R_{switch} is N times faster than R_{line}, then only negligible queuing will occur at the input ports. This is because even in the worst case, where all N input lines are receiving packets, and all packets are to be forwarded to the same output port, each batch of N packets (one packet per input port) can be cleared through the switch fabric before the next batch arrives.
- Output port queuing is illustrated in Figure 4.10. At time t, a packet has arrived at each of the incoming input ports, each destined for the uppermost outgoing port. Assuming identical line speeds and a switch operating at three times the line speed, one time unit later (that is, in the time needed to receive or send a packet), all three original packets have been transferred to the outgoing port and are queued awaiting transmission. In the next time unit, one of these three packets will have been transmitted over the outgoing link. In our example, two *new* packets have

arrived at the incoming side of the switch; one of these packets is destined for this uppermost output port.

- Packet scheduling plays a crucial role in providing **quality-of-service guarantees**. Similarly, if there is not enough memory to buffer an incoming packet, a decision must be made to either drop the arriving packet (a policy known as **drop-tail**) or remove one or more already-queued packets to make room for the newly arrived packet.
- In some cases, it may be advantageous to drop (or mark the header of) a packet *before* the buffer is full in order to provide a congestion signal to the sender. A number of packet-dropping and -marking policies are known as **active queue management** (**AQM**) algorithms.

- One of the most widely studied and implemented AQM algorithms are the **Random Early Detection** (**RED**) algorithm. Under RED, a weighted average is maintained for the length of the output queue.
- If the average queue length is less than a minimum threshold, min_{th}, when a packet arrives, the packet is admitted to the queue. Conversely, if the queue is full or the average queue length is greater than a maximum threshold, max_{th}, when a packet arrives, the packet is marked or dropped.
- Finally, if the packet arrives to find an average queue length in the interval [min_{th}, max_{th}], the packet is marked or dropped with a probability that is typically some function of the average queue length, min_{th}, and max_{th}.

If the switch fabric is not fast enough (relative to the input line speeds) to transfer *all* arriving packets through the fabric without delay, then packet queuing can also occur at the input ports, as packets must join input port queues to wait their turn to be transferred through the switching fabric to the output port. To illustrate an important consequence of this queuing, consider a crossbar switching fabric and suppose that(1) all link speeds are identical, (2) that one packet can be transferred from any one input port to a given output port in the same amount of time it takes for a packet to be received on an input link, and (3) packets are moved from a given input queue to their desired output queue in an FCFS manner. Multiple packets can be transferred in parallel, as long as their output ports are different. However, if two packets at the front of two input queues are destined for the same output queue, then one of the packets will be blocked and must wait at the input queue—the switching fabric can transfer only one packet to a given output port at a time.

Figure 4.11 shows an example in which two packets (darkly shaded) at the front of their input queues are destined for the same upper-right output port. Suppose that the switch fabric chooses to transfer the

packet from the front of the upper-left queue. In this case, the darkly shaded packet in the lower-left queue must wait. But not only must this darkly shaded packet wait, so too must the lightly shaded packet that is queued behind that packet in the lower-left queue, even though there is *no* contention for the middle-right output port (the destination for the lightly shaded packet). This phenomenon is known as **head-of-the-line (HOL) blocking** in an input-queued switch—a queued packet in an input queue must wait for transfer through the fabric (even though its output port is free) because it is blocked by another packet at the head of the line.

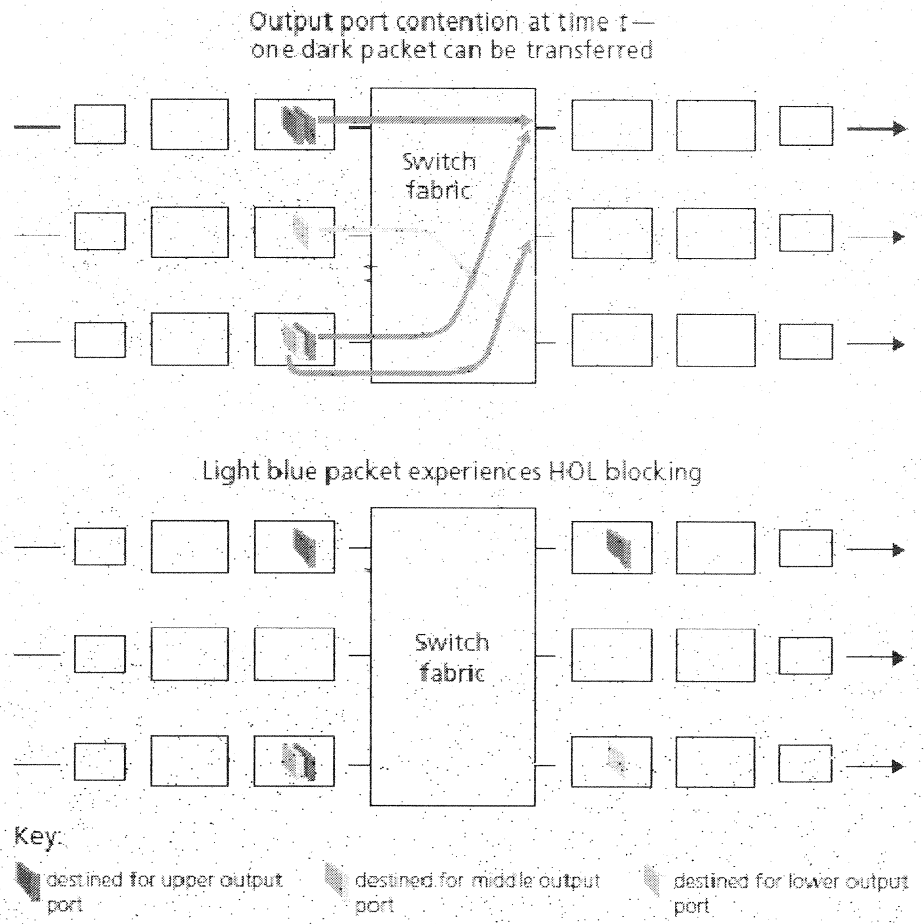


Figure 4.11 ♦ HOL blocking at an input queued switch

4.3.5 The Routing Control Plane

The routing control plane fully resides and executes in a routing processor within the router. The network-wide routing control plane is thus decentralized—with different pieces (e.g., of a routing algorithm) executing at different routers and interacting by sending control messages to each other. Additionally, router and switch vendors bundle their hardware data plane and software control plane together into closed (but inter-operable) platforms in a vertically integrated product.

4.4.4 IPv6

In the early 1990s, the Internet Engineering Task Force began an effort to develop a successor to the IPv4 protocol. A prime motivation for this effort was the realization that the 32-bit IP address space was beginning to be used up, with new subnets an IP nodes being attached to the Internet (and being allocated unique IP addresses) at a breathtaking rate. To respond to this need for a large IP address space, a new IP protocol, IPv6, was developed. The designers of IPv6 also took this opportunity to tweak and augment other aspects of IPv4, based on the accumulated operational experience with IPv4.

IPv6 Datagram Format

The format of the IPv6 datagram is shown in Figure 4.24.

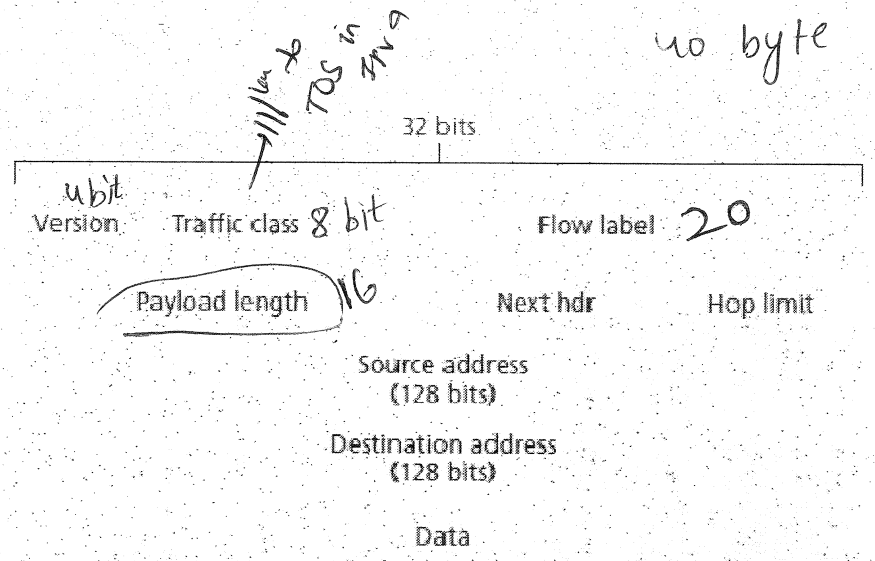


Figure 4.24 ♦ IPv6 datagram format

The most important changes introduced in IPv6 are evident in the datagram format:

• *Expanded addressing capabilities*. IPv6 increases the size of the IP address from 32 to 128 bits. This ensures that the world won't run out of IP addresses. Now, every grain of sand on the planet can be IP-addressable. In addition to unicast and multicast addresses, IPv6 has introduced a new type of address, called an **anycast address**, which allows a datagram to be delivered to any one of a group of hosts. (This feature could be used, for example, to send an HTTP GET to the nearest of a number of mirror sites that contain a given document.)

• *A streamlined 40-byte header.* As discussed below, a number of IPv4 fields have been dropped or made optional. The resulting 40-byte fixed-length header allows for faster processing of the IP datagram. A new encoding of options allows for more flexible options processing.

• *Flow labeling and priority.* IPv6 has an elusive definition of a **flow**. RFC 1752 and RFC 2460 state that this allows "labeling of packets belonging to particular flows for which the sender requests special handling, such as a nondefault quality of service or real-time service." For example, audio and video transmission might likely be treated as a flow.

On the other hand, the more traditional applications, such as file transfer and e-mail, might not be treated as flows. It is possible that the traffic carried by a high-priority user (for example, someone paying for better service for their traffic) might also be treated as a flow.

What is clear, however, is that the designers of IPv6 foresee the eventual need to be able to differentiate among the flows, even if the exact meaning of a flow has not yet been determined. The IPv6 header also has an 8-bit traffic class field. This field, iike the TOS field in IPv4, can be used to give priority to certain datagrams within a flow, or it can be used to give priority to datagrams from certain applications (for example, ICMP) over datagrams from other applications (for example, network news). As noted above, a comparison of Figure 4.24 with Figure 4.13 reveals the simpler, more streamlined structure of the IPv6 datagram.

The following fields are defined in IPv6:

• ***Version***. This 4-bit field identifies the IP version number. Not surprisingly, IPv6 carries a value of 6 in this field. Note that putting a 4 in this field does not create a valid IPv4 datagram.

• ***Traffic class***. This 8-bit field is similar in spirit to the TOS field we saw in IPv4.

• ***Flow label***. As discussed above, this 20-bit field is used to identify a flow of Datagrams.

• ***Payload length.*** This 16-bit value is treated as an unsigned integer giving the number of bytes in the IPv6 datagram following the fixed-length, 40-byte datagram header.

• ***Next header.*** This field identifies the protocol to which the contents (data field) of this datagram will be delivered (for example, to TCP or UDP). The field uses the same values as the protocol field in the IPv4 header.

TTL

• ***Hop limit.*** The contents of this field are decremented by one by each router that forwards the datagram. If the hop limit count reaches zero, the datagram is discarded.

• ***Source and destination addresses***. The various formats of the IPv6 128-bit address are described in RFC 4291.

• ***Data.*** This is the payload portion of the IPv6 datagram. When the datagram reaches its destination, the payload will be removed from the IP datagram and passed on to the protocol specified in the next header field.

The discussion above identified the purpose of the fields that are included in the IPv6 datagram. Comparing the IPv6 datagram format in Figure 4.24 with the IPv4 datagram format that we saw in Figure 4.13, we notice that several fields appearing in the IPv4 datagram are no longer present in the IPv6 datagram:

• *Fragmentation/Reassembly*. IPv6 does not allow for fragmentation and reassembly at intermediate routers; these operations can be performed only by the source and destination. If an IPv6 datagram received by a router is too large to be forwarded over the outgoing link, the router simply drops the datagram and sends a "Packet Too Big" ICMP error message (see below) back to the sender.

• *Header checksum.* Because the transport-layer (for example, TCP and UDP) and link-layer (for example, Ethernet) protocols in the Internet layers perform checksumming, the designers of IP probably felt that this functionality was sufficiently redundant in the network layer that it could be removed.

- *Options.* An options field is no longer a part of the standard IP header. However, it has not gone away. Instead, the options field is one of the possible next headers pointed to from within the IPv6 header. That is, just as TCP or UDP protocol headers can be the next header within an IP packet, so too can an options field. The removal of the options field results in a fixed-length, 40- byte IP header.

Transitioning from IPv4 to IPv6

- How will the public Internet, which is based on IPv4, be transitioned to IPv6? The problem is that while new IPv6-capable systems can be made backward compatible, that is, can send, route, and receive IPv4 datagrams, already deployed IPv4-capable systems are not capable of handling IPv6 datagrams.

- Probably the most straightforward way to introduce IPv6-capable nodes is a **dual-stack** approach, where IPv6 nodes also have a complete IPv4 implementation. Such a node, referred to as an IPv6/IPv4 node in RFC 4213, has the ability to send and receive both IPv4 and IPv6 datagrams.
- When interoperating with an IPv4 node, an IPv6/IPv4 node can use IPv4 datagrams; when interoperating with an IPv6 node, it can speak IPv6. IPv6/IPv4 nodes must have both IPv6 and IPv4 addresses. They must furthermore be able to determine whether another node is IPv6-capable or IPv4-only.

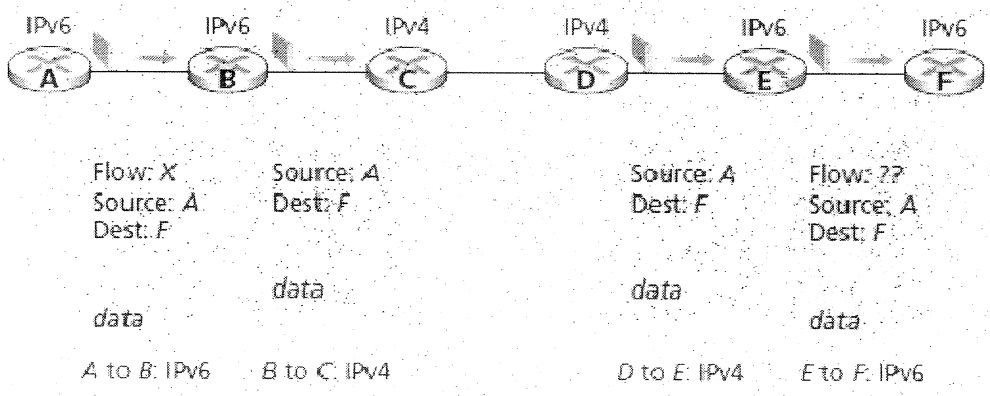


Figure 4.25 ♦ A dual-stack approach

Tunneling

An alternative to the dual-stack approach, also discussed in RFC 4213, is known as **tunneling**.

- Suppose two IPv6 nodes (for example, B and E in Figure 4.25)
  - want to interoperate using IPv6 datagrams but are connected to each other by intervening IPv4 routers.
  - intervening set of IPv4 routers between two IPv6 routers as a **tunnel**, as illustrated in Figure 4.26.
- With tunneling, the IPv6 node on the sending side of the tunnel (for example, B) takes the *entire* IPv6 datagram and puts it in the data (payload) field of an IPv4 datagram.
- This IPv4 datagram is then addressed to the IPv6 node on the receiving side of the tunnel (for example, E) and sent to the first node in the tunnel (for example, C).
- The intervening IPv4 routers in the tunnel route this IPv4 datagram among themselves, just as they would any other datagram, blissfully unaware that the IPv4 datagram itself contains a complete IPv6 datagram. The IPv6 node on the receiving side of the tunnel eventually receives the IPv4 datagram (it is the destination of the IPv4 datagram!), determines that the IPv4 datagram contains an IPv6 datagram, extracts the IPv6 datagram, and then routes the IPv6 datagram exactly as it would if it had received the IPv6 datagram from a directly connected IPv6 neighbor

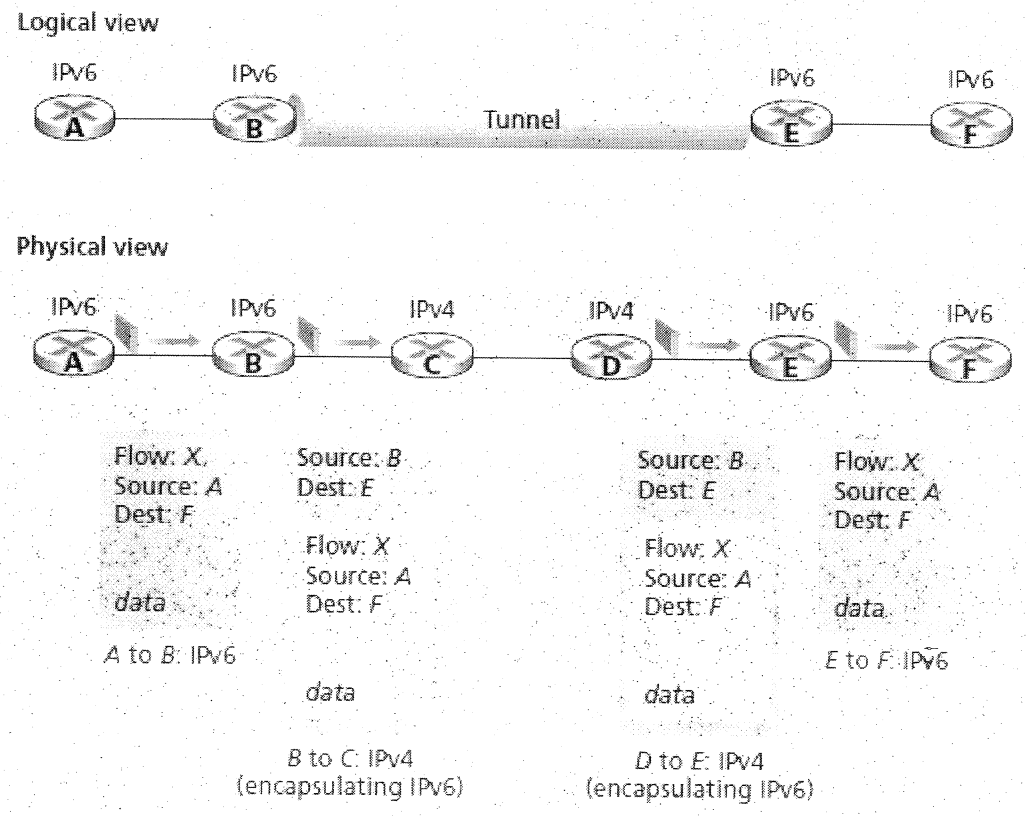


Figure 4.26 ♦ Tunneling

4.4.5 A Brief Foray into IP Security

The services provided by an IPsec session include:

• ***Cryptographic agreement.*** Mechanisms that allow the two communicating hosts to agree on cryptographic algorithms and keys.

• ***Encryption of IP datagram payloads***. When the sending host receives a segment from the transport layer, IPsec encrypts the payload. The payload can only be decrypted by IPsec in the receiving host.

• ***Data integrity***. IPsec allows the receiving host to verify that the datagram's header fields and encrypted payload were not modified while the datagram was en route from source to destination.

• ***Origin authentication.*** When a host receives an IPsec datagram from a trusted source (with a trusted key—see Chapter 8), the host is assured that the source IP address in the datagram is the actual source of the datagram. When two hosts have an IPsec session established between them, all TCP and UDP

segments sent between them will be encrypted and authenticated. IPsec therefore provides blanket coverage, securing all communication between the two hosts for all network applications.

4.5.1 The Link-State (LS) Routing Algorithm

Recall that in a link-state algorithm, the network topology and all link costs areknown, that is, available as input to the LS algorithm. In practice this is accomplished by having each node broadcast link-state packets to *all* other nodes in the network, with each link-state packet containing the identities and costs of its attached links. In practice (for example, with the Internet's OSPF routing protocol, discussed in Section 4.6.1) this is often accomplished by a **link-state broadcast** then run the LS algorithm and compute the same set of least-cost paths as every other node.

The link-state routing algorithm we present below is known as *Dijkstra's algorithm*.

. Let us define the following notation:

• $D(v)$: cost of the least-cost path from the source node to destination v as of this iteration of the algorithm.

• $p(v)$: previous node (neighbor of v) along the current least-cost path from the source to v.

• N': subset of nodes; v is in N' if the least-cost path from the source to v is definitively known.

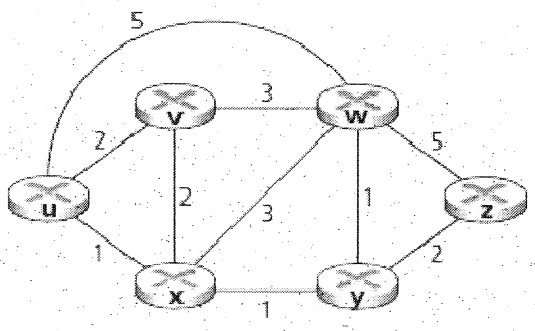
The global routing algorithm consists of an initialization step followed by a loop. The number of times the loop is executed is equal to the number of nodes in the network. Upon termination, the algorithm will have calculated the shortest paths from the source node u to every other node in the network.

Link-State (LS) Algorithm for Source Node *u*

```
1  Initialization:
2     N' = {u}
3     for all nodes v
4       if v is a neighbor of u
5         then D(v) = c(u,v)
6       else D(v) = ∞
7
8  Loop
9     find w not in N' such that D(w) is a minimum
10    add w to N'
11    update D(v) for each neighbor v of w and not in N':
12          D(v) = min( D(v), D(w) + c(w,v) )
13    /* new cost to v is either old cost to v or known
14     least path cost to w plus cost from w to v */
15 until N'= N
```

As an example, let's consider the network in Figure 4.27

u-v x w y z
1 2 5 ∞ ∞



and compute the least-cost paths from *u* to all possible destinations. A tabular summary of the algorithm's computation is shown in Table 4.3, where each line in the table gives the values of the algorithm's variables at the end of the iteration. Let's consider the few first steps in detail.

• In the initialization step, the currently known least-cost paths from *u* to its directly attached neighbors, *v*, *x*, and *w*, are initialized to 2, 1, and 5, respectively. Note in particular that the cost to *w* is set to 5 (even though we will soon see that a lesser-cost path does indeed exist) since this is the cost of the direct (one hop) link from *u* to *w*. The costs to *y* and *z* are set to infinity because they are not directly connected to *u*.

• In the first iteration, we look among those nodes not yet added to the set *N*_ and find that node with the least cost as of the end of the previous iteration. That node is *x*, with a cost of 1, and thus *x* is added to the set *N*_. Line 12 of the LS algorithm is then performed to update *D*(*v*) for all nodes *v*, yielding the results shown in the second line (Step 1) in Table 4.3. The cost of the path to *v* is unchanged. The cost of the path to *w* (which was 5 at the end of the initialization) through node *x* is found to have a cost of 4. Hence this lower-cost path is selected and *w*'s predecessor along the shortest path from *u* is set to *x*. Similarly, the cost to *y* (through *x*) is computed to be 2, and the table is updated accordingly.

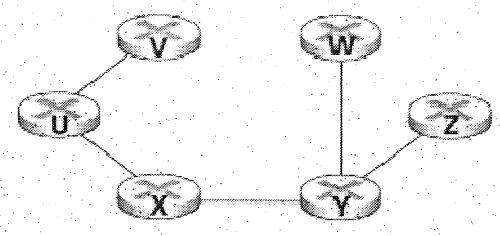
• In the second iteration, nodes *v* and *y* are found to have the least-cost paths (2), and we break the tie arbitrarily and add *y* to the set *N*_ so that *N*_ now contains *u*, *x*, and *y*. The cost to the remaining nodes not yet in *N*_, that is, nodes *v*, *w*, and *z*, are updated via line 12 of the LS algorithm, yielding the results shown in the third row in the Table 4.3.

• And so on. . . .

step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

Table 4.3 ♦ Running the link-state algorithm on the network in Figure 4.27

When the LS algorithm terminates, we have, for each node, its predecessor along the least-cost path from the source node.



Destination	Link
v	(u, v)
w	(u, x)
x	(u, x)
y	(u, x)
z	(u, x)

Figure 4.28 ♦ Least cost path and forwarding table for nodule u

4.5.2 The Distance-Vector (DV) Routing Algorithm

Whereas the LS algorithm is an algorithm using global information, the **distancevector** (**DV**) algorithm is iterative, asynchronous, and distributed. It is *distributed* in that each node receives some information from one or more of its *directly attached* neighbors, performs a calculation, and then distributes the results of its calculation back to its neighbors. It is *iterative* in that this process continues on until no more information is exchanged between neighbors. (Interestingly, the algorithm is also self-terminating—there is no signal that the computation should stop; it just stops.) The algorithm is *asynchronous* in that it does not require all ofthe nodes to operate in lockstep with each other.

Before we present the DV algorithm, it will prove beneficial to discuss an important relationship that exists among the costs of the least-cost paths. Let $d_x(y)$ be the cost of the least-cost path from node x to node y. Then the least costs are related by the celebrated Bellman-Ford equation, namely,

$d_x(y) = \min_v\{c(x,v) + d_v(y)\}$, (4.1)

where the $\min_v$ in the equation is taken over all of x's neighbors.

Distance-Vector (DV) Algorithm

At each node, x:

```
1  Initialization:
2     for all destinations y in N:
3        Dx(y) = c(x,y)    /* if y is not a neighbor then c(x,y) = ∞ */
4     for each neighbor w
5        Dw(y) = ? for all destinations y in N
6     for each neighbor w
7        send distance vector Dx = [Dx(y): y in N] to w
8
9  loop
10    wait (until I see a link cost change to some neighbor w or
11          until I receive a distance vector from some neighbor w)
12
13    for each y in N:
14       Dx(y) = minv{c(x,v) + Dv(y)}
15
16    if Dx(y) changed for any destination y
17       send distance vector Dx = [Dx(y): y in N] to all neighbors
18
19 forever
```

Hierarchical Routing

Homogenous set of routers all executing the same routing algorithm is a bit simplistic for atleast two important reasons:

Scale: As the number of routers becomes large, the Overhead involved in computing, storing, and communicating routing information becomes prohibitive. Today's public Internet consists of millions of hosts. Storing routing information at each of these hosts would clearly require enormous amounts of memory.

* LS (Link-State) updates among all of the routers in the Internet would leave no bandwidth left for sending packets.

* A distance vector algorithm that iterated among such a large number of routers would surely never converge.

* Clearly, something must be done to reduce the complexity of route computation in networks as large as the public internet.

Administrative autonomy:

An Organizations should be able to run and administer its network as it wishes, while still being able to connect its network to other outside networks.

Both of these problems can be solved by Organi-zing routers into autonomous systems (ASs), with each AS consisting of a group of routers that are typically under the same administrative control (e.g. Operated by the same ISP or belonging to the same company network)

* Routers within the same AS all run the same routing algorithm (for ex. an LS or DV algorithm).

* Routing algorithm running within an autonomous system is called an intra autonomous routing protocol.

* One or more routers in an AS will have the added task of being responsible for forwarding packets to destinations outside the AS; these routers are called gateway routers.

Figure 4.32 provides a simple example with three ASs: AS1, AS2, and AS3.

→ In this figure, the heavy lines represent direct link connections between pairs of routers.

→ the thinner lines hanging from the routers represent the subnets that are directly connected to the routers.

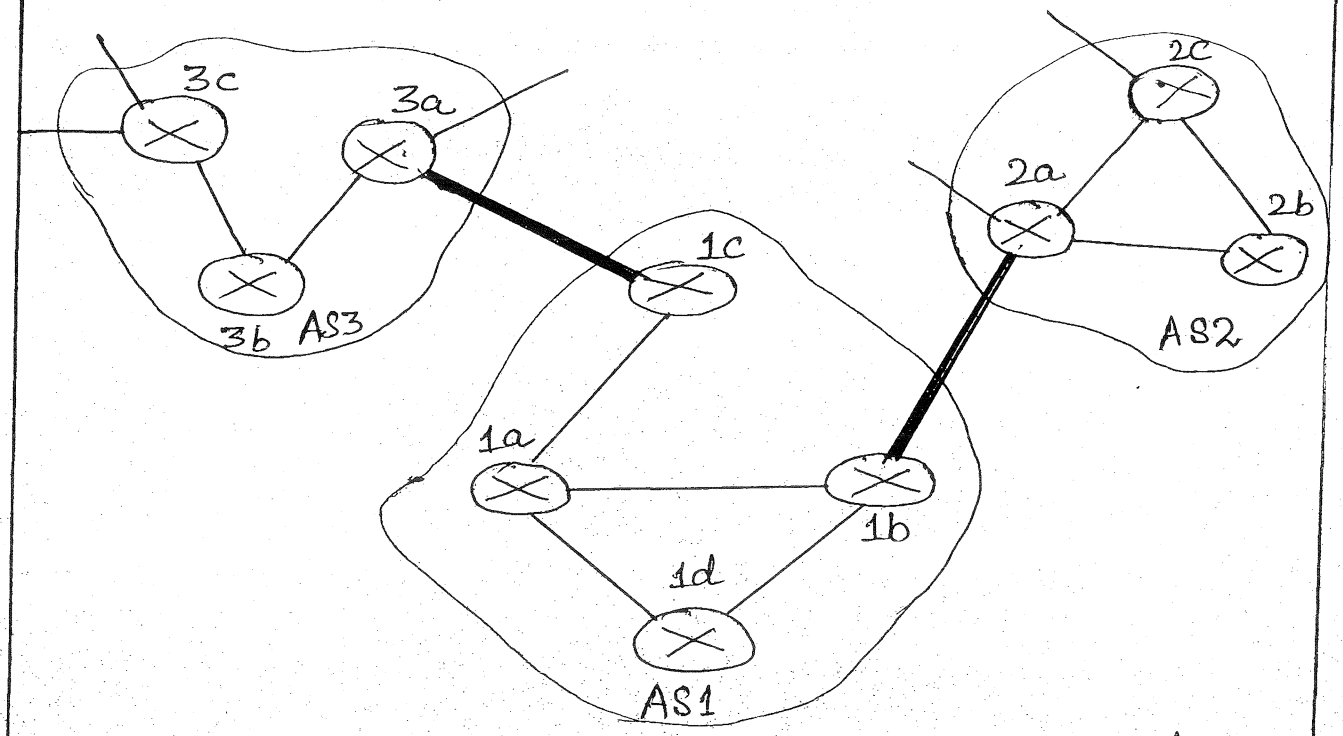


Figure 4.32 An Example of interconnected autonomous systems.

AS1 has four routers - 1a, 1b, 1c, 1d which run the intra-AS routing protocol used within AS1.

Similarly, AS2 and AS3 each have three routers.

* Note that the intra-AS routing protocols running in AS1, AS2, and AS3 need not be the same.

* Also note that the routers 1b, 1c, 2a and 3a are all gateway routers.

* How does a router, within some AS, know how to route a packet to a destination that is outside the AS?

* If the AS has only one gateway router that connects to only one other AS.

* Gateway Routers are responsible for forwarding packets to destinations outside the AS.

Two main tasks of inter-AS routing Protocol

1) obtaining Reachability information from neighboring AS

2) Propagating the reachability information to all routers internal to AS.

The communicating ASs must run the same inter AS routing protocol. For. ex. BGP.

Figure

One approach, which is often employed in Practice, is to use hot-potato routing. In hot potato routing, the AS gets rid of the packet (the hot potato) as quickly as possible (as inexpensively as possible). This is done by having a router send the packet to the gateway router that has the smallest router to gateway cost among all gateways with a path to the destination. Figure 4.33 summarizes the steps in adding an outside AS destination in a router's forwarding table.

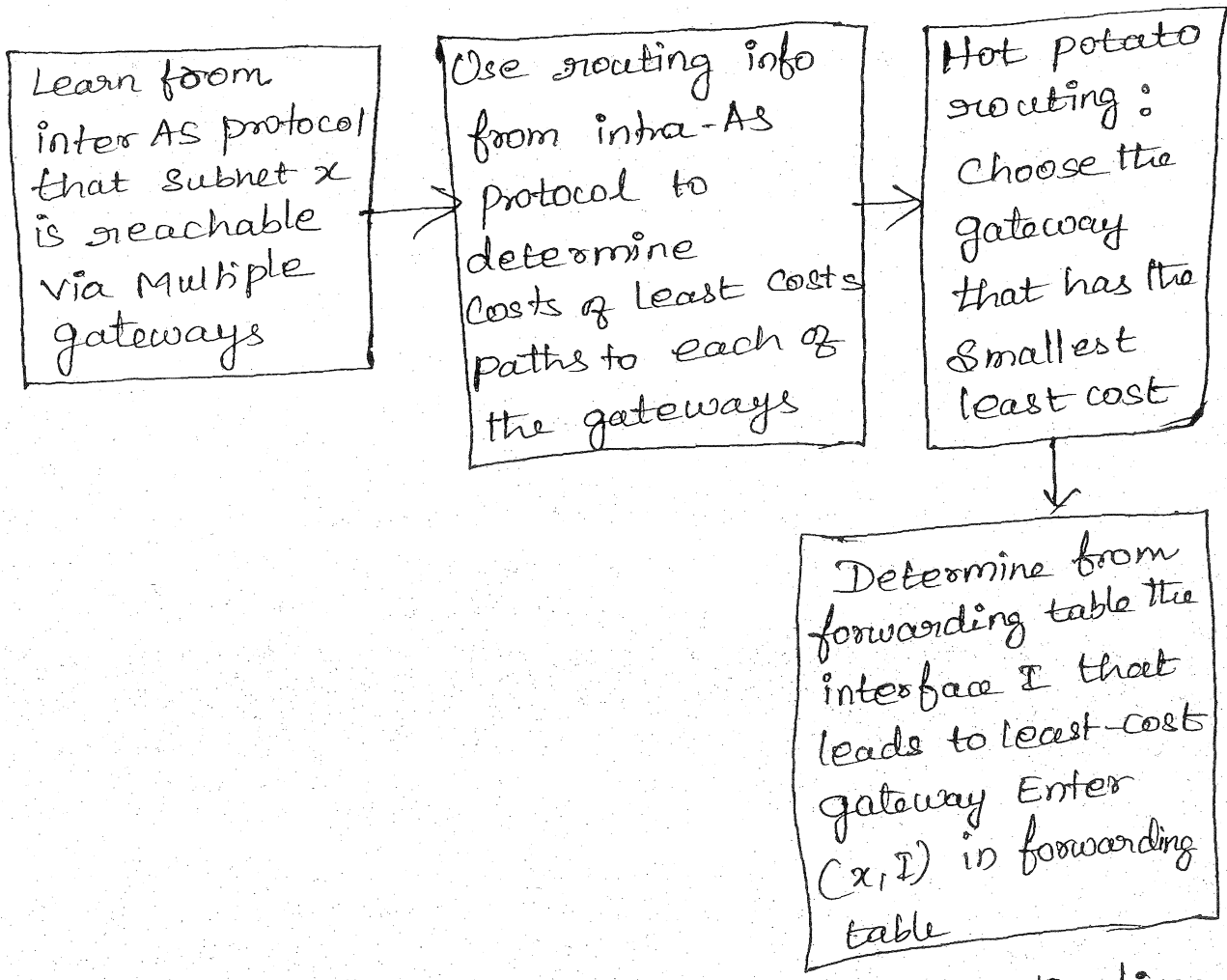


figure 4.33
Steps in adding an Outside AS destination in a router's forwarding table.

4.6 Routing in the Internet

4.6.1. Intra-AS Routing in the Internet : RIP

→ An Intra-AS routing Protocol is Used to determine how routing is performed within an autonomous system.

→ Intra-AS routing protocol are also known as interior gateway protocols.

Two routing protocols within an autonomous system
- RIP (Route information Protocol)
- OSPF (open shortest path first)

CN (10)

4.6. Routing in the Internet

Internet's routing protocols job is to determine the path taken by a datagram between source and destination.

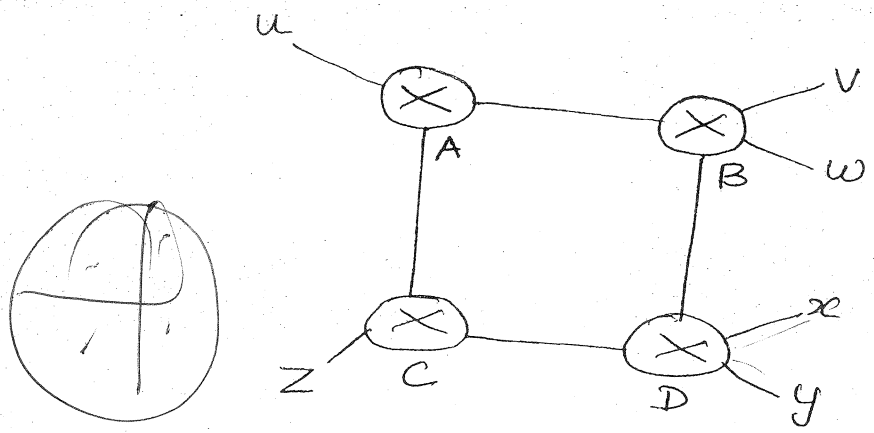
Intra-As Routing in the Internet : RIP

An intra-AS routing protocol is used to determine how routing is performed within an autonomous system (AS). Intra-AS routing Protocols are also known as interior gateway protocols. There are two types protocols

- Routing Information Protocol (RIP)
- Open shortest path first (OSPF)

RIP

RIP is a distance vector protocol that operates in a manner very close to the idealized DV Protocol. The Version of RIP specified in RFC 1058 uses hop count as a cost metric, that is each link has a cost of 1. In RIP, costs are actually from source router to a destination subnet. RIP uses the term hop, which is the number of subnets traversed along the shortest path from source router to destination subnet, including the destination subnet. Figure 4.34 illustrates an AS with six leaf subnets. The table to the figure indicates the number of hops from the source A to each of the leaf subnets.



Destination	Hops
u	1
v	2
w	2
x	3
y	3
z	2

Figure 4.34 Number of hops from source router A to various subnets.

* The Maximum cost of a path is limited to 15, thus limiting the use of RIP to autonomous systems that are fewer than 15 hops in diameter.

* In RIP, routing updates are exchanged between neighbors approximately every 30 seconds using a RIP response message.

* The response message sent by a router or host contains a list of up to 25 destination subnets within the AS, as well as the sender's distance to each of those subnets. Response message are also known as RIP advertisements.

Example :

In this figure, lines connecting the routers denote subnets. Only selected routers (A, B, C and D) and subnets (w, x, y and z) are labeled.

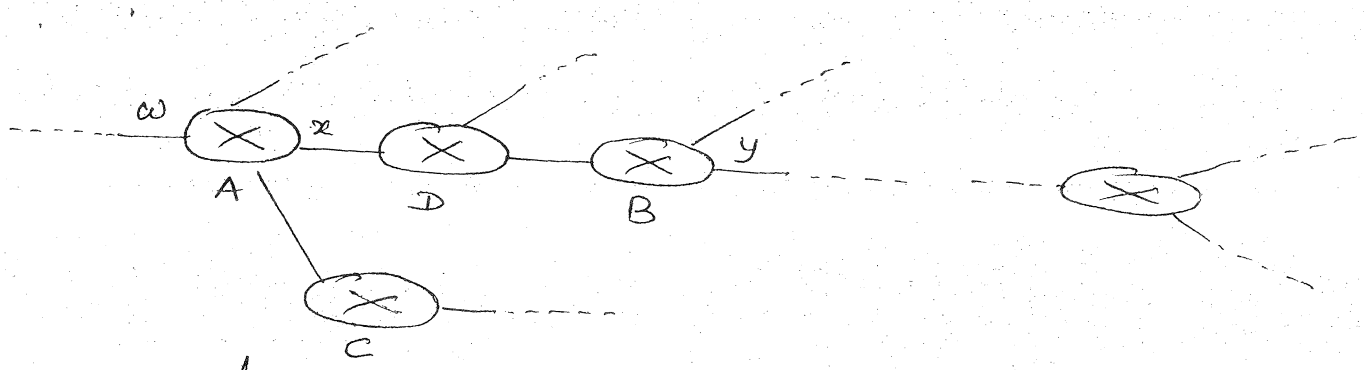


figure 4.35 A portion of an autonomous system

Each router maintains a RIP table known as a routing table. A router's routing table includes both the router's distance vector and the router's forwarding table. figure 4.36 shows the routing table for router D.

* Note that routing tables has three columns. The first column is for the destination Subnet, the Second column indicates the identity of the next router along the shortest path to the destination Subnet, and the third column indicates the number of hops to get to the destination subnet along the shortest path. For this Example, the table indicates that to Send a datagram from router D to destination Subnet w, the datagram should first be forwarded to neighboring router A. the table also indicates that destination subnet w is two hops away along the shortest path. to the destination subnet, and the third column indicates the number of hops (that is, the number of subnets that have to be traversed, including the destination Subnet) to get to the destination Subnet along the shortest path.

For this example, the table indicates that to send a datagram from router D to destination subnet w, the datagram should first be forwarded to neighboring router A, the table also indicates that destination subnet w is two hops away along the shortest path. Similarly, the table indicates that subnet z is seven hops away via router B.

Destination Subnet	Next Router	No. of hops to Destination
w	A	2
y	B	2
z	B	7
x	—	1
.....	.....	.....

Now suppose that 30 seconds later, router D receives from router A the advertisement shown in figure 4.37. This information indicates, in particular, that subnet z is only four hops away from router A. Router D, upon receiving this advertisement, merges the advertisement, with the routing table. In particular, router D learns that there is now a path through router A to subnet z that is shorter than the path through router B. Thus router D updates its routing table to account for the shortest path as figure 4.38.

Destination Subnet	Next Router	No. of hops to Destination
z	C	4
w	—	1
x	—	1
...	...	...

Figure 4.37 Advertisement from router A.

Destination Subnet	Next Router	No. of hops to Destination
w	A	2
y	B	2
z	A	5
...	...	...

Figure 4.38 Routing table in router D after receiving advertisement from router A.

Implementation of RIP

RIP routers exchange advertisements approximately every 30 seconds. If a router does not hear from its neighbor at least once every 180 seconds, that neighbor is considered to be no longer reachable, that is either the neighbor has died or the connecting link has gone down. When this happens, RIP modifies the local routing table and then propagates this information by sending advertisements to its neighboring routers.

neighboring routers. Because RIP is implemented as an application layer process manipulate the routing tables within the UNIX kernel), it can send and receive messages over a standard socket and use a standard transport protocol.

4.6.2. Intra-AS Routing in the Internet : OSPF

OSPF routing is widely used for intra-AS routing in the internet. OSPF and its closely related IS-IS, are typically deployed in upper-tier ISP's whereas RIP is deployed in lower-tier ISP's and enterprise networks.

* OSPF is a link state protocol that uses flooding of link state information and a Dijkstra's least cost path algorithm.

* With OSPF, a router constructs a complete topological Map of the entire autonomous system.

* The router then locally run Dijkstra's Shortest path algorithm to determine a shortest path tree to all subnets, with itself as the root node.

* Individual link costs are configured by the network administrator. The administrator might choose to set all link costs to 1, thus achieving minimum hop routing, or might choose to set ~~all~~ link ~~costs to~~ weights to be inversely proportional to link capacity in order to discourage traffic from using low-bandwidth links.

A router can also request information about its neighbor's cost to a given destination using RIP's request message.

* Routers send RIP request and response messages to each other over UDP using port number 520.

* The UDP segment is carried between routers in a standard IP datagram. The fact that RIP uses a transport layer Protocol (UDP) on top of a network layer Protocol (IP) to implement network layer functionality.

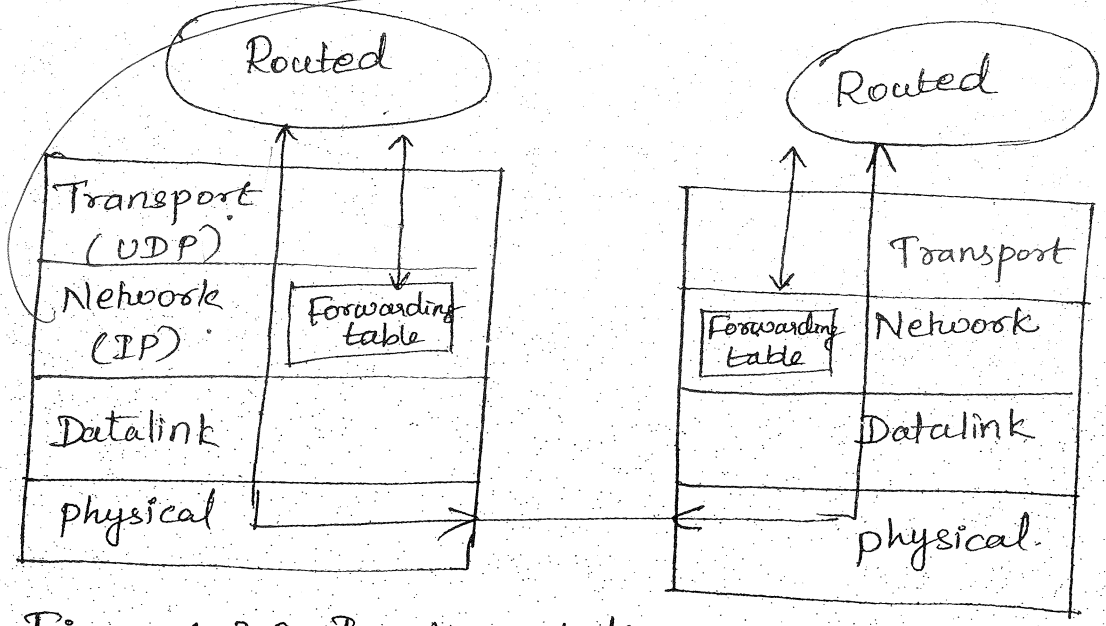


Figure 4.39 Implementation of RIP as the routed daemon.

RIP is typically implemented in a UNIX system, for example, a UNIX workstation serving as a router. A process called routed (pronounced "route dee") executes RIP, that is maintains routing information and exchanges messages with routed processes running in

With OSPF, a router broadcasts routing information to all other routers in the autonomous system, not just to its neighboring routers. A router broadcasts link state information whenever there is a change in a link state

* "This periodic updating of link state advertisements adds robustness to the link state algorithm."

* OSPF advertisements are contained in OSPF messages that are carried by IP, upper layer protocol of 89 for OSPF.

* The OSPF protocol also checks that links are operational (via a HELLO message that is sent to an attached neighbor) and allows an OSPF router to obtain a neighboring router's database of network wide link state.

Security: Exchanges between OSPF routers (for example link updates) can be authenticated, within AS, thus preventing malicious intruders from injecting incorrect information into router tables.

→ Simple
→ MD5
} Two Types of authentication

Simple: The same password is configured on each router. When a router sends an OSPF packet, it includes the password in plaintext. It is not secure.

MD5: * It is based on the shared keys that are configured in all the routers.

* Sequence numbers are also used with MD5 authentication to protect against replay attacks

Multiple Same-Cost paths : When multiple paths to a destination have the same cost, OSPF allows multiple paths to be used.

Integrated Support for unicast and Multicast routing

Multicast OSPF (MOSPF) provides simple extensions to OSPF to provide for multicast routing.

Support for hierarchy within a single routing domain

* Ability to structure an autonomous system hierarchically.

4.6.3. Inter-AS Routing : BGP - Exterior

Border Gateway Protocol version 4, determines path between source & destination pairs that span multiple ASs.

* It is a example of exterior gateway protocol. [RFC 4271]

BGP provides each AS a means to

1) obtain subnet reachability information from neighboring ASs.

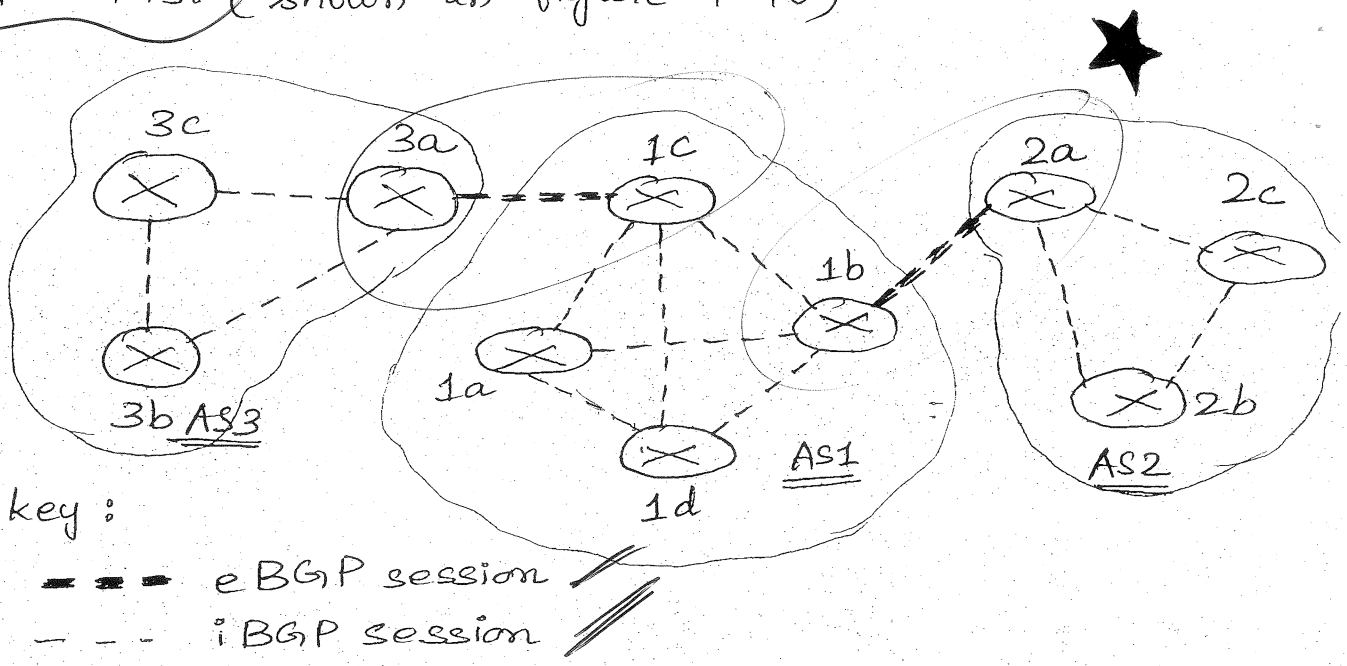
2) Propagate the reachability information to all routers internal to the AS.

3) Determine "good" routes to subnets based on the reachability information and on AS policy.

* BGP allows each subnet to advertise its existence to the rest of the Internet.

BGP Basics

In BGP, pairs of routers exchange routing information over semipermanent TCP connections using ~~Port~~ 179. (shown in figure 4.40)



There is typically one such BGP TCP connection for each link that directly connects two routers in two different ASs; In this figure there is a TCP connection between gateway routers 3a and 1c and another TCP connection between gateway routers 2a and 1b. There are semipermanent BGP TCP connections between routers within AS.

For each TCP connection, the two routers at the end of the connection are called BGP peers, and the TCP connection along with all the BGP messages sent over the connection is called a BGP session.

a BGP session that spans two AS's is called an external BGP (eBGP) session, and a BGP session between routers in the same AS is called an internal BGP (iBGP) session.

BGP allows each AS to learn which destinations are reachable via its neighboring ASs. In BGP destinations are not hosts but instead are CIDRized prefixes, with each prefix representing a subnet or a collection of subnets.

BGP, distribute prefix reachability information over the BGP sessions shown in figure. Using the eBGP session between the gateway routers 3a and 1c, AS3 sends AS1 the list of prefixes that are reachable from AS3. and AS1 sends AS3 list of prefixes that are reachable from AS1. Similarly, AS1 and AS2 exchange prefix reachability information through their gateway routers 1b and 2a.

Path Attributes and BGP Routes

In BGP, an autonomous system is identified by its globally unique autonomous system number (ASN). ASN

When a Router advertises prefix across a BGP session, it includes with a prefix a number of BGP attributes

- AS-PATH
- NEXT-HOP

AS-PATH

This attribute contains the ASs through which the advertisements for the prefix has passed. When a prefix is passed into an AS, the AS adds its ASN to the AS-PATH attribute.

NEXT-HOP

Providing the critical link between the inter AS and intra AS routing protocols, the NEXT-HOP attribute has a subtle but important use. The NEXT-HOP is the router interface that begins the AS-PATH.

gateway router 3a in AS3 advertises a router to gateway router 1c in AS1 using eBGP. The route includes the advertised prefix, x, and an AS-PATH to the Prefix. This advertisement also includes the NEXT-HOP, which is the IP address of the router 3a interface that leads to 1c.

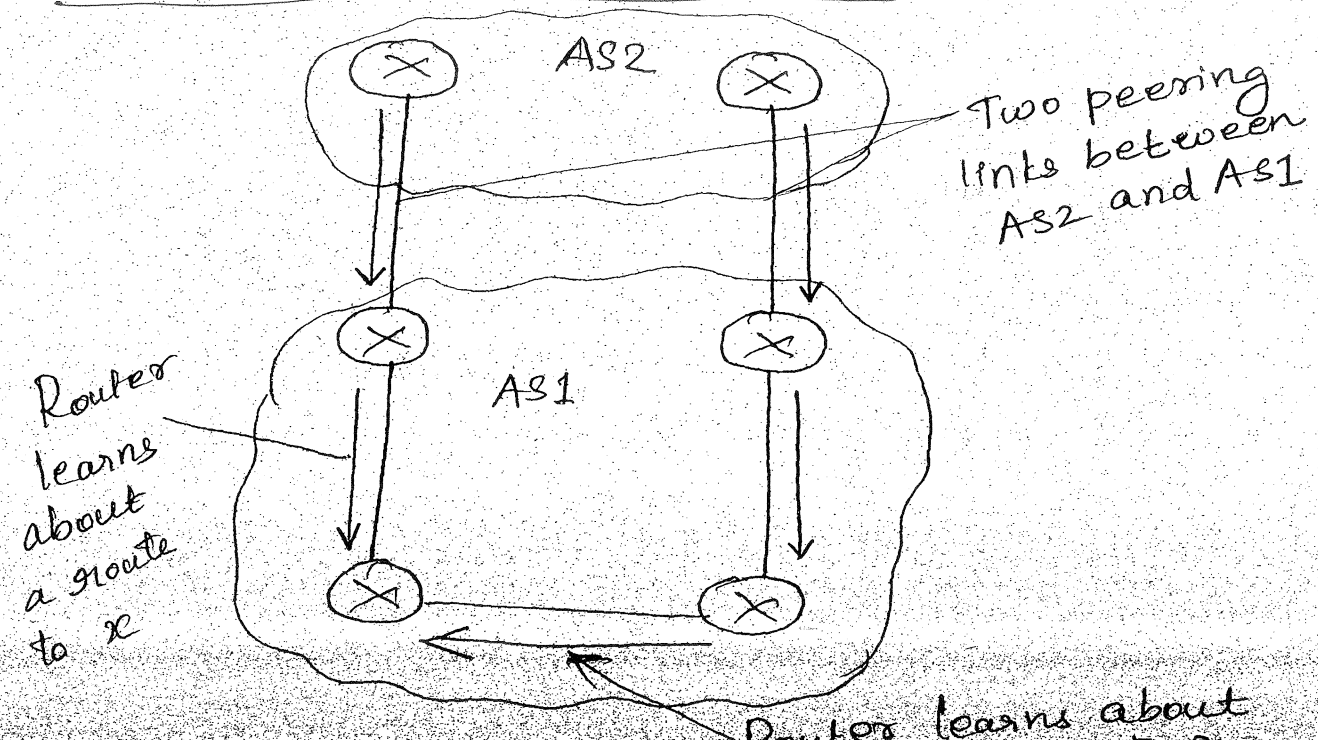


Figure 4.41. NEXT-HOP

Figure 4.41 illustrates another situation where the NEXT-HOP is needed. In this figure AS1 & AS2 are connected by two peering links. A router in AS1 could learn about two different routes to the same prefix x. These two routes could have the same AS-PATH to x. but could have different NEXT-HOP values corresponding to the different Peering links. Using the NEXT-HOP values and the intra AS routing algorithm, the router can determine the cost of the path to each peering link, and then apply hot potato routing to determine the appropriate interface.

When a gateway router receives a route advertisement, it uses its import policy to decide whether to accept or filter the route and whether to set certain attributes such as the router Preferences metrics.

BGP Route Selection

BGP uses eBGP and iBGP to distribute routes to all the routers within ASs. From this distribution, a router may learn about more than one route to any one prefix, in which case the router must select one of the possible routes. The input into this route selection process is the set of all routes that have been learned and accepted by the router. If there are two or more routes to the same prefix then BGP sequentially invokes the following elimination rules

until one route remains

1) Routers are assigned a local Preference Value as one of their attributes. The local preference of a route could have been set by the router or could have been learned by another router in the same AS. This is a policy decision that is left up to the AS's network administrator. The routes with the highest local preference values are selected.

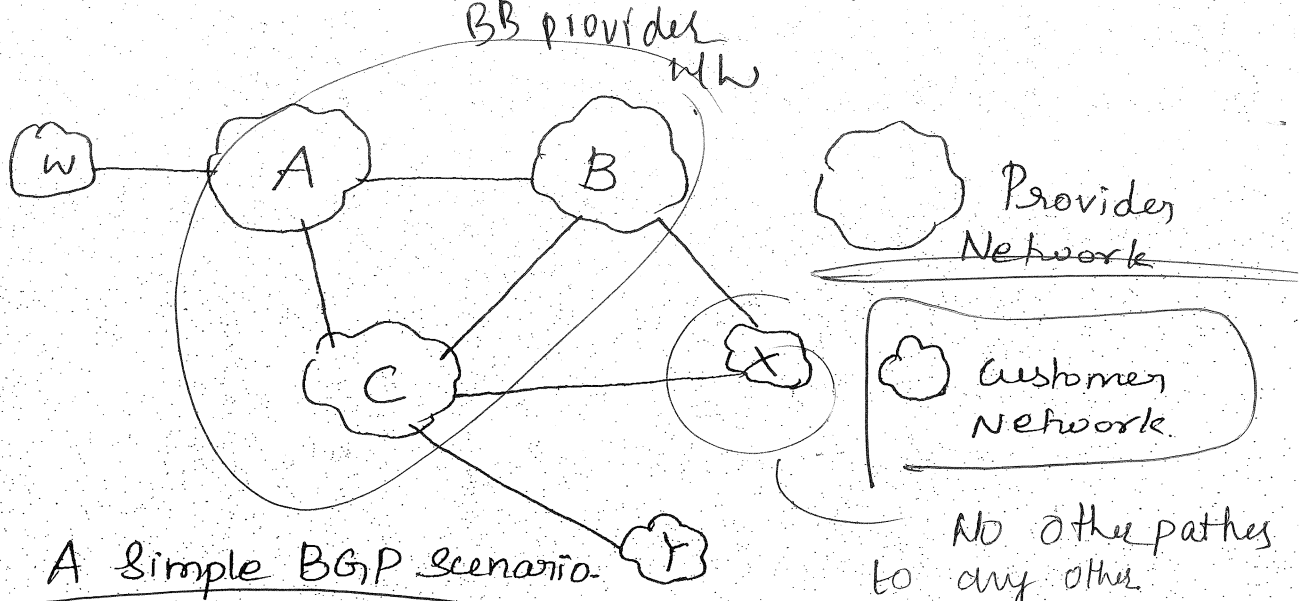
2) From the remaining routes, the route with shortest AS-PATH is selected. If this rule were only rule for route selection, then BGP would be using a DV algorithm for path determination, where the distance metric uses the number of AS hops rather than the number of router hops

3) From the remaining routes, the route with the closest NEXT-HOP router is selected. Here closest means the router for which the cost of the least cost path, determined by the intra AS algorithm is the smallest. This process is called hot potato routing.

Routing Policy

Figure shows six interconnected autonomous systems: A, B, C, W, X and Y. It is important to Note that A, B, C, W, X and Y are ASs, not routers Let's assume that autonomous systems W, X, Y are stub networks and that A, B and C are backbone provider networks. Also assume that

A, B and C, all peer with each other, and provide full BGP information to their customer networks. All traffic entering a stub network must be destined for that network, and all traffic leaving a stub network must have originated in that network. W and Y are clearly stub networks. X is a multihomed stub network since it is connected to the rest of the network via two different providers. However like W and Y, X itself must be the source, destination of all traffic leaving/entering X. But how will this stub network behavior be implemented and enforced? How will X be prevented from forwarding traffic between B and C? This can easily be



A Simple BGP Scenario

accomplished by controlling the manner in which BGP routes are advertised. In particular, X will function as a stub network if it advertises (to its neighbors B and C) that it has no paths to any other destination except itself.

Even though X may know of a path, say XCY that reaches network Y, it will not advertise this path to B. Since B is unaware that X has a path to Y, B would never forward traffic destined to Y (or C) via X. This simple example illustrates how a selective route advertisement policy can be used to implement customer/provider routing relationships.

Let us next focus on a provider network, say AS B. Suppose that B has learned (from A) that A has path AW to W. B can thus install the route BAW into its routing information base. Clearly, B also wants to advertise the path BAW to its customer X, so that X knows that it can route to W via B. But should B advertise the path BAW to C? If it does so, then C could route traffic to W via CBAW. If A, B, and C are all backbone providers, then B might rightly feel that it should not have to shoulder the burden of carrying transit traffic between A and C.

4.7.2 Multicast

* In Multicast Service, a multicast packet is delivered to only a subset of network nodes.

* A Number of emerging network applications require the delivery of packets from one or more senders to a group of receivers.

* Applications include bulk data transfer (for example, the transfer of a software upgrade from the Software developer to users needing upgrade), streaming continuous media (for example, the transfer of the audio, video, and text of a live lecture to a set of distributed lecture participants), Shared data applications (for example, a white board or teleconferencing application that is Shared among many distributed participants).

Problems - Multicast communication

1. How to identify the receivers of a multicast packet.
2. How to address a packet sent to these receivers.

In case of unicast communication,

-the IP address of the receiver (destination) is carried in each IP unicast datagram and identifies the single recipient.

In case of broadcast,

all nodes need to receive the broadcast packet, so no destination addresses are needed.

In case of Multicast

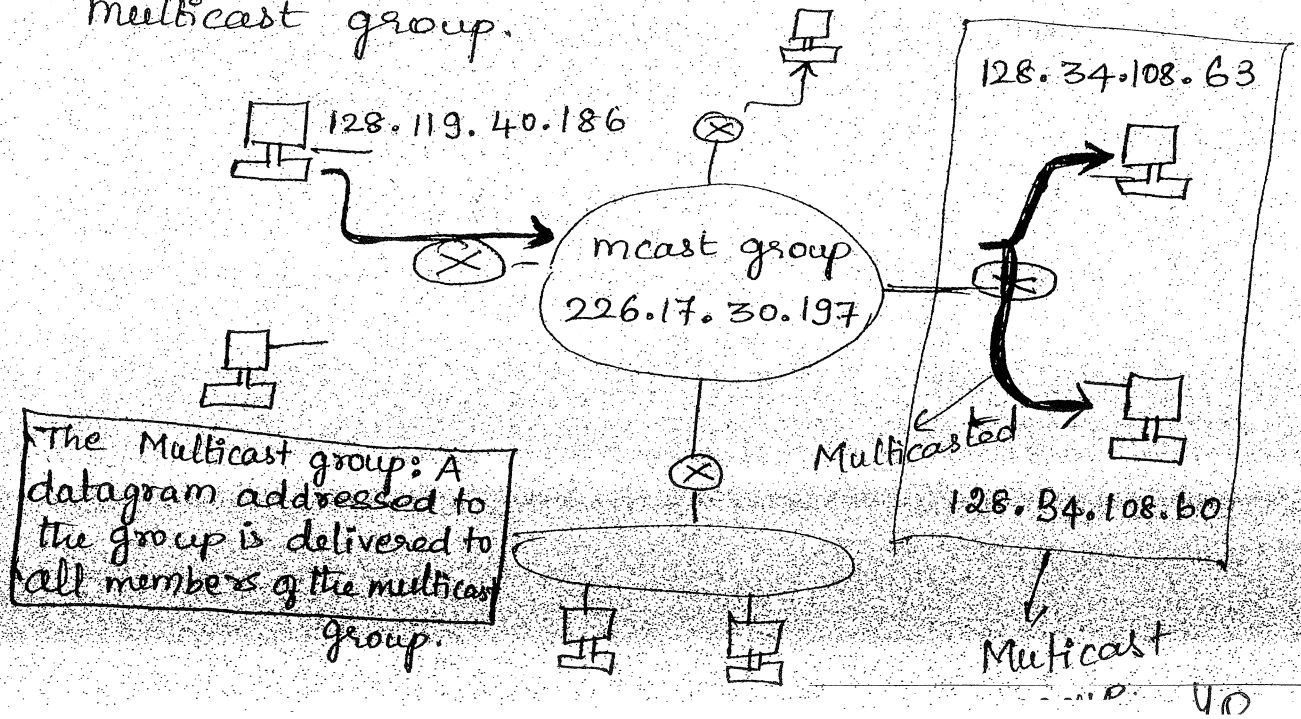
Does it make sense for each multicast packet to carry the IP addresses of all of the multiple recipients?

→ while this approach might be workable with a small number of recipients, it would not scale well to the case of hundreds or thousands of receivers.

* For these reasons, in the Internet architecture, a multicast packet is addressed using address indirection.

* In the Internet, the single identifier that represents a group of receivers is a class D multicast IP address.

* The group of receivers associated with a class D address is referred to as multicast group.



Internet Group Management Protocol

The IGMP Protocol Version 3 operates between a host and its directly attached router.

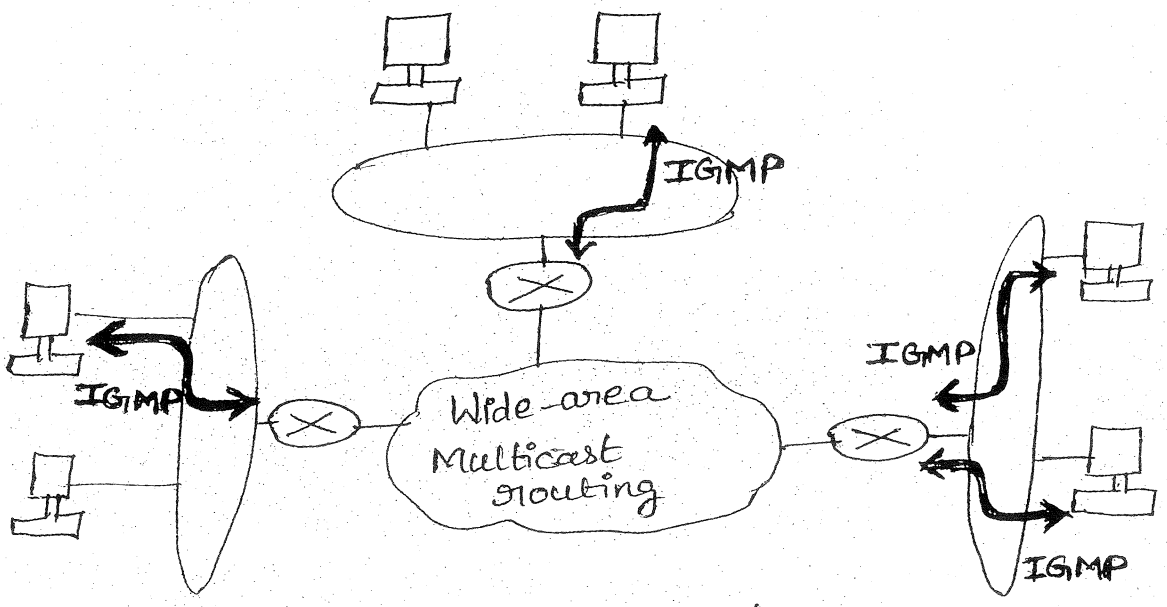


Figure 4.48 the two Components of network layer multicast in the Internet : IGMP and Multicast routing protocols

* IGMP Provides the means for a host to Inform its attached router that an application running on the host wants to join a specific multicast group.

* IGMP interaction is limited to a host and its attached router, another protocol is clearly required to Co-ordinate the multicast routers throughout the Internet, so that multicast datagrams are routed to their final destinations.

IGMP has only three Message types. Like ICMP, IGMP messages are carried (encapsulated) within an IP datagram, with an IP Protocol number of 2.

1. membership_query Message is sent by a router to all hosts on an attached interface to determine the set of all multicast groups that have been joined by the hosts on that interface.

2. membership_report messages can also be generated by a host when an application first joins a multicast group without waiting for a membership_query message from router.

3. Leave_group message is optional.

Sometimes the router infers that a host is no longer in the multicast group if it no longer responds to a membership_query message with the given group address.

———— x ————

Multicast Routing algorithms

The Multicast routing problem is illustrated in figure 4.49. Host joined to the multicast group and immediately attached router too. As shown in figure 4.49, only a subset of routers (those with attached hosts that are joined to the multicast group)

actually needs to receive the multicast traffic.

In figure 4.49, only routers A, B, E and F need to receive the multicast traffic. Since none of the hosts attached to router D are joined to the multicast group and since router C has no attached hosts, neither C nor D needs to receive the multicast group traffic.

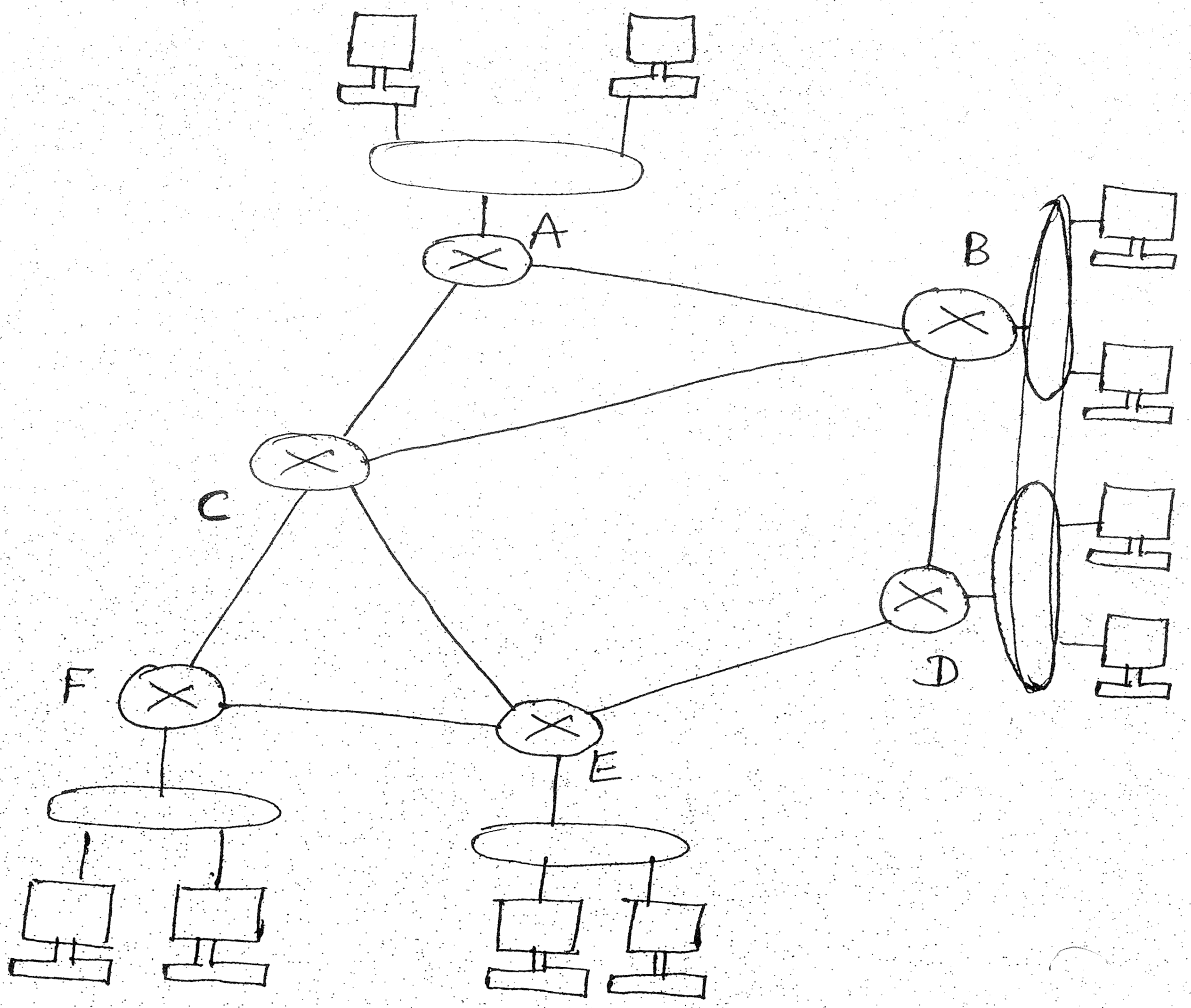


Figure 4.49. Multicast hosts, their attached routers, and other routers.

* The goal of multicast routing then is to find a tree of links that connects all of the

routers that have attached hosts belonging to the Multicast group. Multicast packets will then be routed along this tree from the sender to all of the hosts belonging to the Multicast tree.

Two approaches

→ Single group shared tree
→ Source specific routing tree

Multicasting using a group-shared tree

As in the case of Spanning tree broadcast, Multicast routing over a group shared tree is based on building a tree that includes all edge routers with attached hosts belonging to the multicast group.

- In Practice, a center-based approach is used to construct the multicast-routing tree.
- Edge-routers send join messages addressed to the center-node.

Here is how it works:

1) A center node (rendezvous point or a core) is defined.
2) Then, the edge-routers send unicast-tree join messages to the center-node.
3) Finally, a tree-join message is forwarded toward the center until it either

→ arrives at a node that already belongs to the multicast tree or

→ arrives at the center.

[2] Multicast Routing using a Source Based Tree

- A Source-Specific routing tree is constructed for each individual sender in the group.
- In practice, an RPF algorithm is used to construct a multicast forwarding tree.
- The solution to the problem of receiving unwanted multicast-packets under RPF is known as pruning.
- A Multicast-router that has no attached-hosts will send a prune message to its upstream router.

Multicasting (Multicast Routing) in the Internet

Multicast routing Protocols are:

1) Distance Vector Multicast Routing Protocol (DVMRP)
2) Protocol Independent Multicast (PIM) and
3) Source Specific Multicast (SSM).

1) DVMRP

- DVMRP was the first multicast routing Protocol used in the Internet.
- DVMRP uses an RPF algorithm with Pruning (Reverse path forwarding).

2) PIM

- PIM divides multicast routing into sparse and dense mode.

(i) Dense Mode

> Group members are densely located.
> Most of the routers in the area need to be involved in routing the data
> It is a flood-and-prune reverse path forwarding technique.

ii) Sparse Mode

* No. of. routers with attached group-members is small with respect to total no. of. routers.

* Group-members are widely dispersed.

* This uses rendezvous points to set up the multicast distribution tree.

3) SSM

* only a single sender is allowed to send traffic into the multicast tree. This simplifies tree Construction & maintanence

—x—

— End —

Broadcast and Multicast Routing.

In broadcast routing, the network layer Provides a service of delivering a packet sent from a source node to all other nodes in the network.

Multicast routing enables a single source node to send a copy of a packet to a subset of other network nodes.

Broadcast Routing algorithm

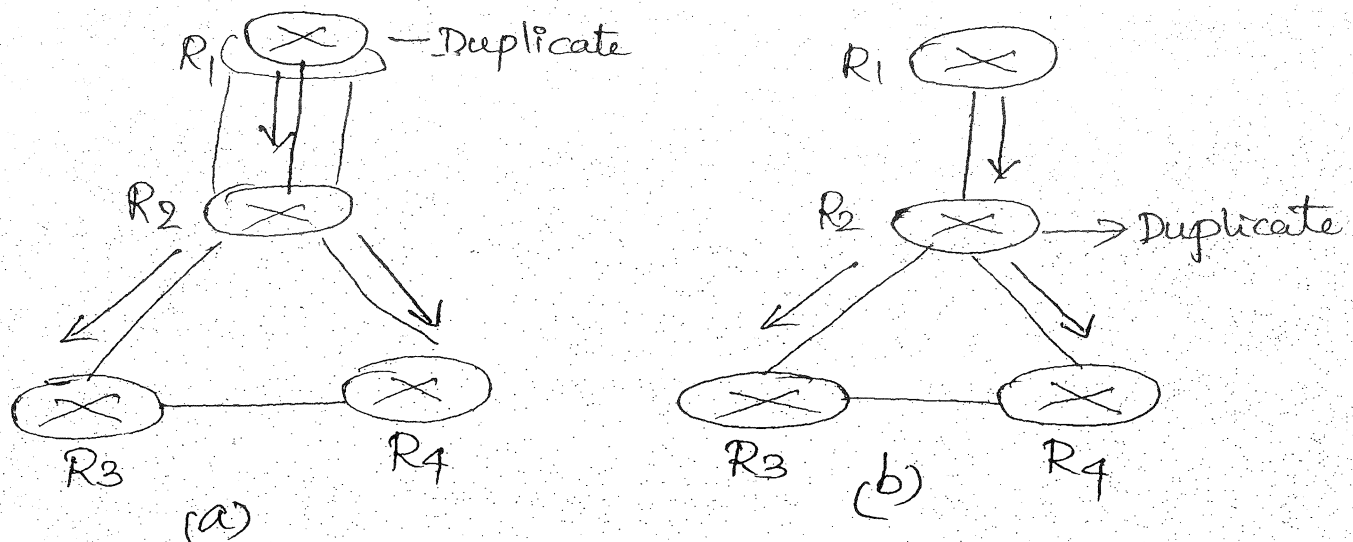
Sending node to send a separate copy of the packet to each destination as shown in figure 4.43(a). Given N destination nodes, the source node simply makes N copies of the Packet, addresses each copy to different destination, and then transmits the N copies to the N destination Using unicast routing. This N-way unicast approach to broadcasting is simple — no new network layer routing Protocol, packet duplication, or forwarding functionality is needed.

Drawback

1) InEfficiency: If the source node is Connected to the rest of the network via a single link, then N separate copies of the

Copies of the (same) packet will traverse this single link. It would clearly be more efficient to send only a single copy of a packet over this first hop and then have the node at the other end of the first hop make and forward any additional needed copies. That is, it would be more efficient for the network nodes themselves to create duplicate copies of a packet.

For Ex, in figure 4.43(b) only a single copy of a packet traverses the R1-R2 link. That packet is then duplicated at R2, with a single copy being sent over links R2-R3 and R2-R4.



Source duplication Vs in-network duplication.

② An implicit assumption of N-way unicast is that broadcast recipients, and their addresses, are known to the sender. But how is this information obtained?

Additional protocol Mechanisms (such as broadcast membership or destination-registration protocol) would be required. This would add more overhead and importantly additional Complexity to a Protocol that had initially seemed quite simple.

(3) A final drawback of N-way unicast relates to the purposes for which broadcast is to be used. Link-State routing protocols use broadcast to disseminate the link-state information that is used to compute unicast routes. Clearly, in situations where broadcasts is used to create & update unicast routes, it would be (unwise) to rely on the unicast routing infrastructure to achieve broadcast.

Given the several drawbacks of N-way unicast broadcast, approaches in which the network nodes themselves play an active role in packet duplication, packet forwarding, and computation of the broadcast routes are clearly of interest.

Uncontrolled Flooding.

The Most obvious technique for achieving broadcast is a flooding approach in which the source node sends a copy of the packet to all of its neighbors.

When a node receives a broadcast packet, it duplicates the packet and forwards it to all of its neighbors. (except the neighbor from which it received the packet).
If the graph is connected, this scheme will eventually deliver a copy of the broadcast packet to all nodes in graph.
Although this scheme is simple & elegant, it has a fatal flaw. (If the graph has cycles, then one or more copies of each broadcast packet will cycle indefinitely. For Ex. R_2 will flood to R_3, R_3 will flood to R_4. R_4 will flood to R_2, and R_2 will flood (again!) to R_3, and so on. This simple scenario results in the endless cycling of two broadcast packets, one clockwise, and one counterclockwise. But there can be even more calamitous fatal flaw: when a node is connected to more than two other nodes, it will create and forward multiple copies of the broadcast packet, each of which will create multiple copies of itself. (at other nodes with more than two neighbors), and so on. This broadcast storm, resulting from endless multiplication of packets, would eventually result in so many broadcast packets being created

so

that the network would be rendered useless.

Controlled flooding.

The key to avoiding a broadcast storm is for a node to judiciously choose when to flood a packet and when not to flood a packet.

(1) In sequence-number controlled flooding, a source node put its address as well as a broadcast sequence number into a broadcast packet, then sends the packet to all of its neighbors. Each node maintains a list of the source address and sequence number of each broadcast packet it has already received, duplicated, and forwarded. When a node receives a broadcast packet, it first checks whether the packet is in this list. If so, the packet is dropped; if not the packet is duplicated and forwarded to all the node's neighbors (except the node from which the packet has just been received).

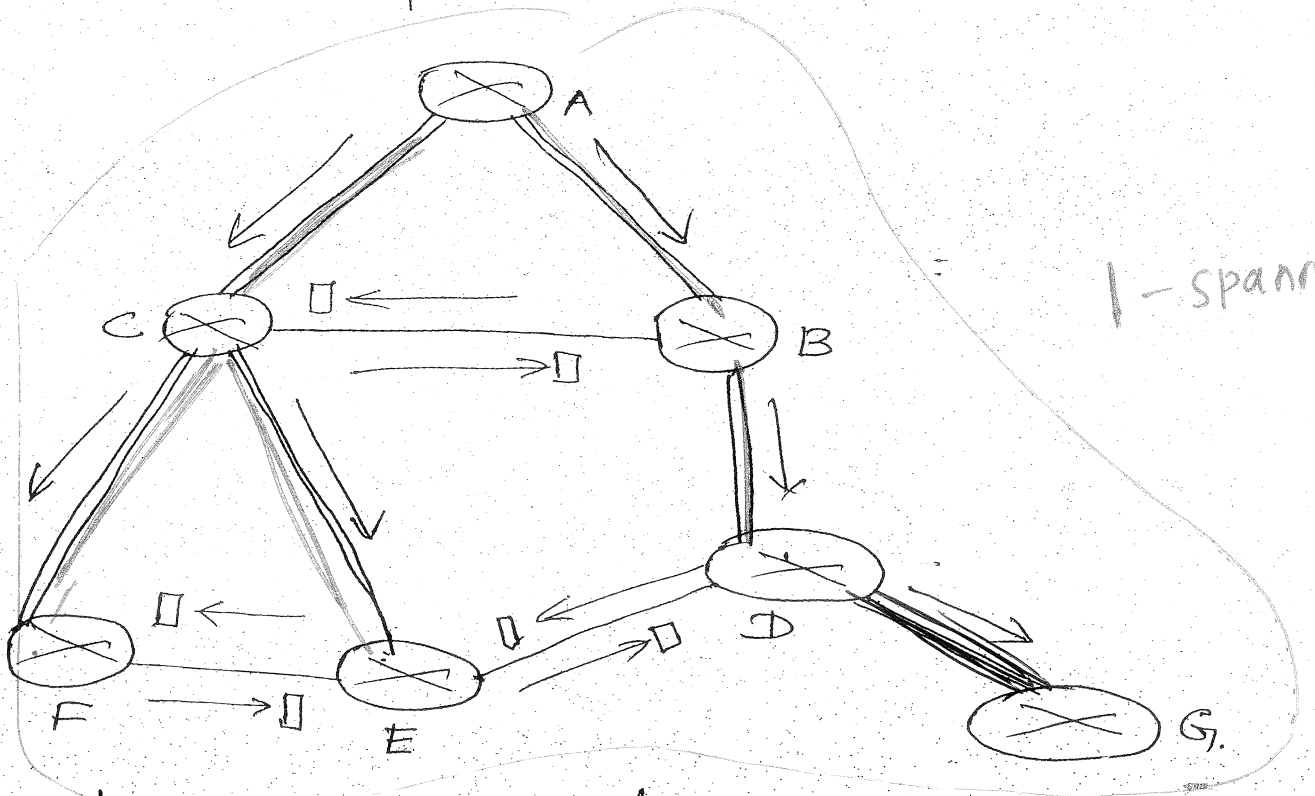
Gnutella Protocol, use sequence number controlled flooding to broadcast queries in its Overlay network.

A second approach to controlled flooding is known as reverse path forwarding (RPF) also sometimes referred to as reverse path broadcast (RPB).

When a router receives a broadcast (RPB) packet with a given source address, it transmits the packet on all of its outgoing links (except the one on which it was received) only if the packet arrived on the link that is on its own shortest unicast path back to the source. Otherwise, the router simply discards the incoming packet without forwarding it on any of its outgoing links. Such a packet can be dropped because the router knows it either will receive or has already received a copy of this packet on the link that is on its own shortest path back to the sender.

Figure 4.44 illustrates RPF. Suppose that the links drawn with thick lines represent the least-cost paths from the receivers to the source (A). Node A initially broadcasts a source A packet to nodes C and B. Node B will forward the source A packet it has received from A to both C and D. B will ignore (drop, without forwarding) any source-A packets it receives from any other nodes (for example from routers C or D).

Let us now consider node C, which will receive a source-A packet directly from A as well as from B. Since B is not on C's own shortest path back to A, C will ignore any Soure-A packets it receives from B. On the other hand, when C receives a source A packet directly from A, it will forward the packet to nodes B, E, F.



key: pkt will be forwarded. →

: pkt will not forwarded beyond receiving router.

Reverse path forwarding. (figure 4.44)

Spanning Tree Broadcast.

While Sequence-number flooding and RPF avoid broadcast storms, they do not Completely avoid the transmission of redundant broadcast packets. For Ex. in figure 4.4, nodes B, C, D, E and F receive either one or two

redundant packets. Ideally every node should receive only one copy of the broadcast packet. Examining the tree consisting of the nodes connected by thick lines in figure 4.45(a) you can see that if broadcast packets were forwarded only along links within this tree, each and every network node would receive exactly one copy of the broadcast packet — exactly the solution we were looking for. This tree is an example of a spanning tree — a tree that contains each and every node in a graph. More formally, a spanning tree — a tree that contains each and every node in a graph. Spanning tree of a graph $G=(N,E)$ is a graph $G'=(N,E')$ such that E' is a subset of E, G' is connected, G' contains no cycles, and G' contains all the original nodes in G. If each link has an associated cost and the cost of a tree is the sum of the link costs, then a spanning tree whose cost is the minimum of all the graph's spanning tree is called a minimum spanning tree.

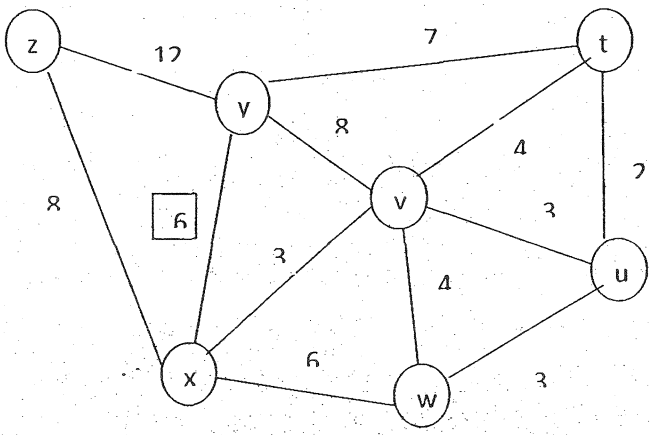
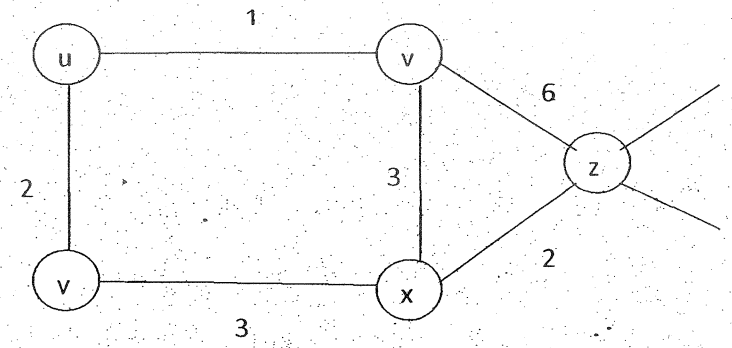
When a source node wants to send a broadcast packet, it sends the packet out on all of the incident links that belong to the spanning tree. A node receiving a broadcast packet then forwards the packet to all its neighbors in the spanning tree.

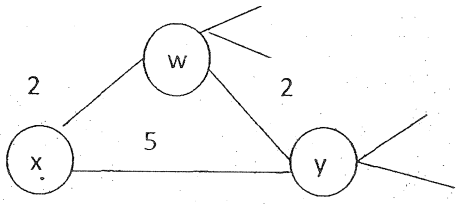
R.N.S. INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

ASSIGNMENT-2

Semester: **V**
Sub Name: **Computer Networks** Sub Code: **15CS52** Faculty: **Mrs. Sudha V**

Q. No	Question	Level	CO Mapping
1.	Discuss why each input port in a high speed router stores a shadow copy of the forwarding table.	L4	CO3
2.	Describe how packet loss can occur at output ports. Can this loss be prevented by increasing the switch fabric speed?	L4	CO3
3.	List and briefly describe three types of switching fabrics. Which, if any, can send multiple packets across the fabric in parallel?	L2,L4	CO3
4.	What is HOL blocking? Does it occur in input ports or output ports?	L2	CO3
5	Describe how packet loss can occur at input ports and describe how packet loss at input ports can be eliminated (without using infinite buffers)	L4	CO3
6	Do routers have IP addresses? if so, how many?	L5	CO3
7	What is the 32 bit binary equivalent of the IP address 223.1	L5	CO3
8	Draw FSM of Congestion Control	L1	CO2
9.	Justify why multicasting is not possible with TCP and explain about TCP connection management with neat diagram.	L5	CO2
11.	Sketch TCP segment structure and explain.	L2	CO2
12.	Write short notes on following a. TCP SYN segment b. SYNACK segment c. FIN_WAIT_1 d. FIN_WAIT_2 e. RST segment	L2	CO2
13.	Explain the costs of congestion of various scenarios.	L2	CO2
14.	Explain various versions of TCP. (TCP Tahoe, TCP Reno, TCP Vegas)	L2	CO2
15.	How does flooding lead to a broadcast storm?	L2	CO2
16.	Explain how routing tables are updated in RIP protocol with an example.	L2	CO3

17.	Consider the following network. With the indicated link costs, use Dijkstra's shortest path algorithm to compute the shortest path from x to all network nodes. Show how the algorithm works by computing a table 1.  	L5	CO3
18	Consider the network shown in question no.17 using Dijkstra's algorithm and showing your work using a table 1, do the following: a. Computer the shortest path from t to all network nodes. b. Compute the shortest path from u to all network nodes. c. Compute the shortest path from w to all network nodes. d. Compute the shortest path from y to all network nodes. e. Compute the shortest path from z to all network nodes.	L5	CO3
19	Consider the network shown below, and assume that each node initially knows the costs to each of its neighbors. Consider the distance vector algorithm and show the distance table entries at node z.  	L5	CO3
20	What are most important BGP attributes? Explain.	L2	CO3

21	Consider the network fragment shown below. X has only two attached neighbors, w and y. w has a minimum-cost path to destination u (not shown) of 5 and y has minimum-cost path to u of 6. The complete paths from w and y to u(and between w and y) are not shown. All link costs in the network have strictly positive integer values.  		
22.	Explain OSPF protocol.	L2	CO3
23.	Describe how BGP works in inter AS networks	L3	CO3
24.	Explain different attributes of BGP protocol	L2	CO3
25.	Write Link-State Routing algorithm.	L2	CO3
26.	Write Distance-Vector Routing algorithm.	L2	CO3
27.	Differentiate IPV4 and IPv6 Header formats.	L4	CO3
28.	Differentiate Link-State Routing and Distance Vector Routing.	L4	CO3
29.	Explain RIP Protocol. List out its drawbacks.	L4	CO3
30.	Explain Interior Gateway Protocols and Exterior Gateway Protocols.	L2	CO3

WIRELESS AND MOBILE NETWORKS

BTS
BSC
BSS

MODULE - 4

Cellular Internet Access

In cellular Network architecture includes terminology of the Global System for Mobile Communications (GSM) standards.

Cellular technology, they often classify the technology as belonging to one of several "generations".

* First Generation (1G) systems were analog FDMA systems designed exclusively for voice only communication.

* The Original 2G systems were also designed for voice, but later extended (2.5G) to support data (i.e., Internet) as well as Voice Service.

* The 3G systems that currently being deployed also support voice and data, but with an ever increasing emphasis on data capabilities and higher speed radio access links.

Cellular Network Architecture, 2G: Voice Connections to the Telephone Network

The term cellular refers to the fact that the region covered by a cellular network

is partitioned into a number of geographic coverage areas, known as cells. Each cell contains a base transceiver station (BTS) that transmits signals to and receives signals from the mobile stations in its cell. The coverage area of a cell depends on many factors, including the transmitting power of the BTS, the transmitting power of the user devices, obstructing buildings in the cell, and the height of base station antennas.

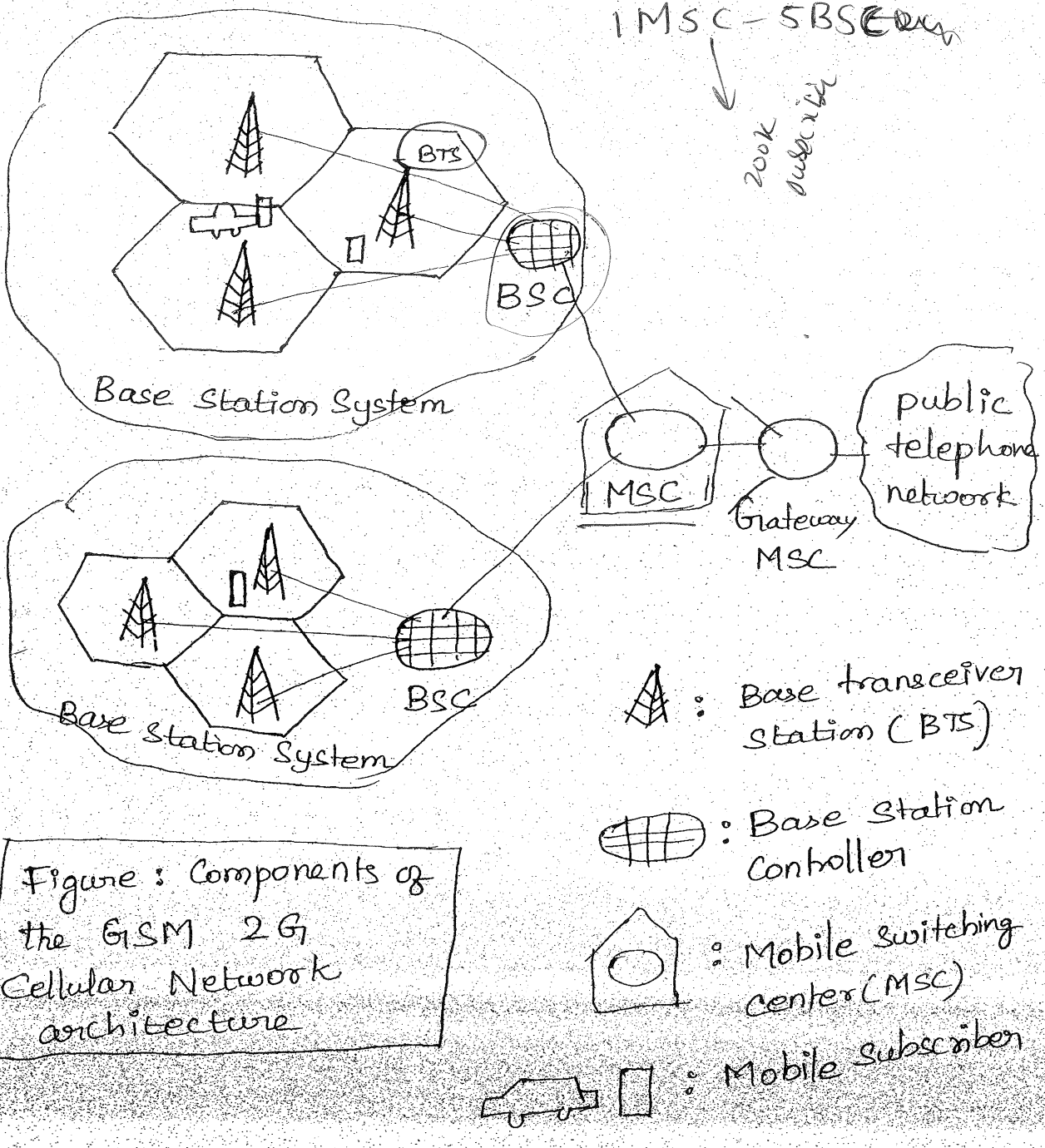


Figure: Components of the GSM 2G Cellular Network architecture

* The GSM standard for 2G cellular Systems uses combined FDM/TDM (radio) for the air interface.

* In combined FDM/TDM systems, the channel is partitioned into a number of frequency sub-bands; within each sub-band, time is partitioned into frames and slots. Thus, for a combined FDM/TDM system, if the channel is partitioned into F sub-bands and time is partitioned into T slots, then the channel will be able to support F-T simultaneous calls.

* A GSM network's base station controller (BSC) will typically service several tens of base transceiver stations (BTS)

* The role of the BSC is to allocate BTS radio channels to mobile subscribers, perform paging (finding the cell in which a mobile user is resident) and perform handoff of mobile users. The base station controller and its controlled base transceiver stations collectively constitute a GSM base station system (BSS).

* Mobile switching center (MSC) plays the central role in user authorization and accounting (e.g. determining whether a mobile device is allowed to connect to the cellular network), call establishment and tear-down and handoff.

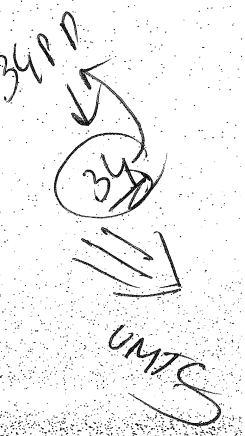
* A single MSC will typically contain upto five BSC's, resulting in approximately 200k subscribers per MSC.

* A cellular provider's network will have a number of MSCs with special MSCs known as gateway MSCs connecting the provider's cellular network to the larger public telephone network.

3G Cellular Data Networks: Extending the Internet to Cellular Subscribers

Users likes to read email, access the Web, get location-dependent services (e.g. maps and restaurant recommendations) and perhaps even watch streaming video while travelling. To do this, Smartphone will need to run a TCP/IP protocol stack and connect into the Internet via the cellular data Network.

3G technology focus on the UMTS (Universal mobile Telecommunication Service) 3G standards developed by the 3rd Generation Partnership project (3GPP).

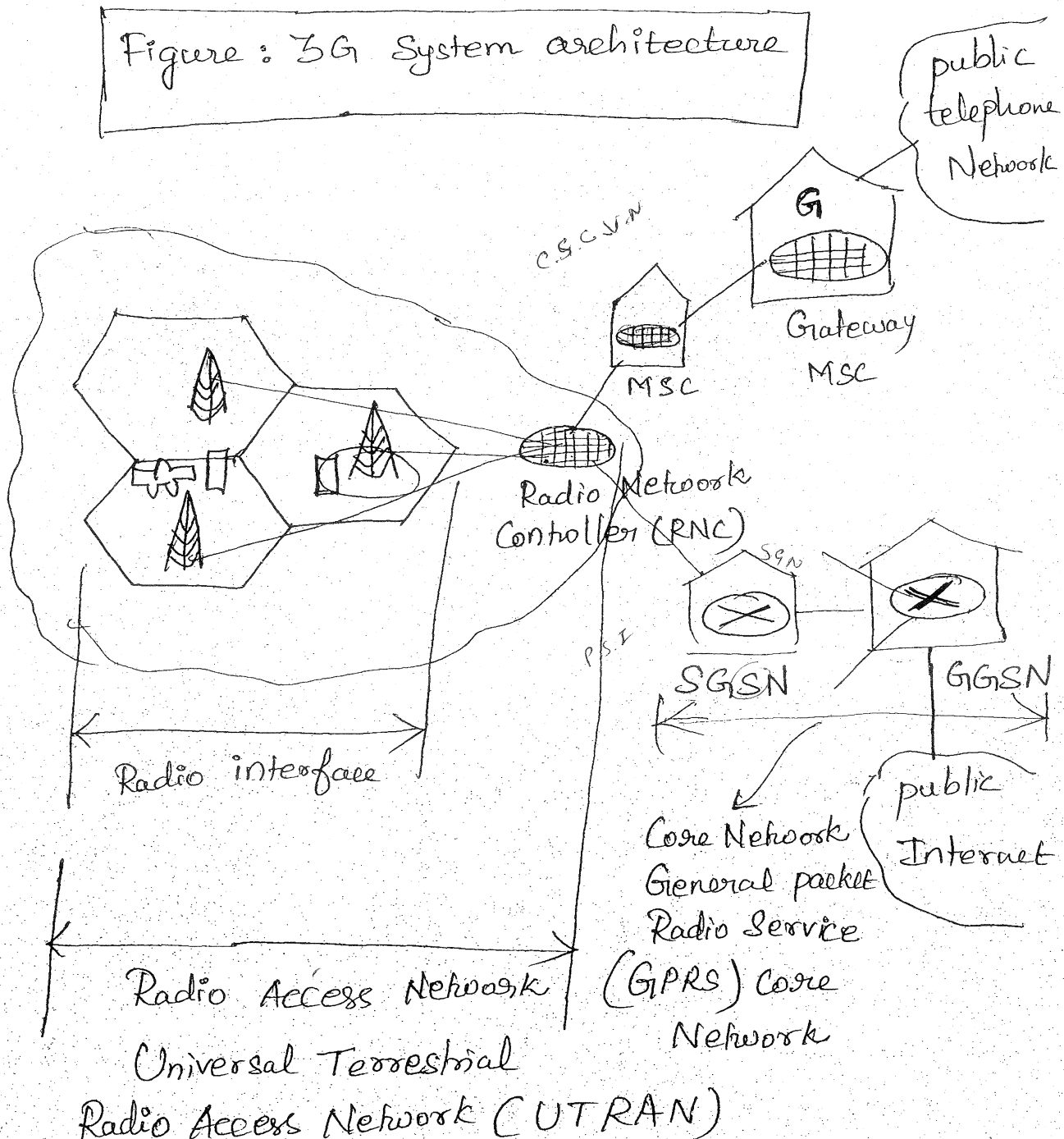


3G Core Network

The 3G core cellular data Network connects radio access networks to the public Internet. The approach taken by the designers of 3G data services is clear: leave the

existing core GSM cellular voice network untouched, adding additional cellular data functionality in parallel to the existing cellular voice Network.

Figure : 3G System architecture



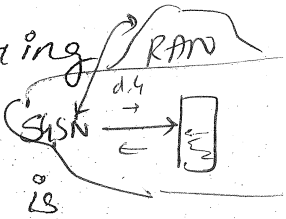
Key

: Serving GPRS Support Node (SGSN)

G : Gateway GPRS Support Node

There are two types of nodes in the 3G core Network. Serving GPRS Support Nodes (SGSNs) and Gateway GPRS Support Nodes (GGSNs).

* An SGSN is responsible for delivering datagrams to/from the mobile nodes in the radio access network to which the SGSN is attached.



* The SGSN interacts with the cellular voice Network's MSC for that area, providing User authorization and handoff, maintaining location (cell) information about active mobile nodes & performing datagram forwarding between mobile Nodes in the radio access network and a GGSN.

* The GGSN is thus acts as a gateway, Connecting multiple SGSN's into larger Internet. It is a last piece of 3G infrastructure that a datagram Originating at a mobile node encounters before entering the larger Internet.

* The Radio Network Controller typically controls several cell base transceiver Stations Similar to the base Stations that we encountered in 2G Systems.

3G Radio Access Network : The Wireless Edge

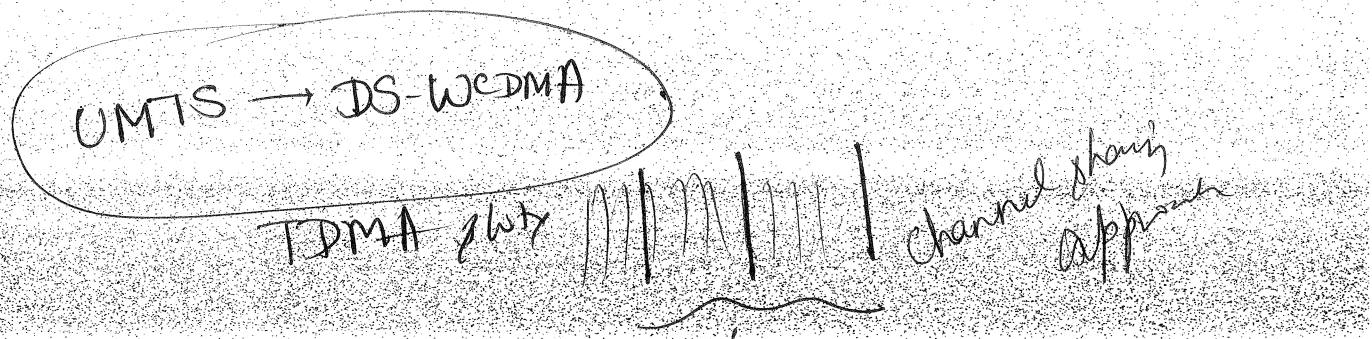
The RNC connects to both the Circuit Switched cellular voice network via an MSC, and to the packet Switched Internet via an SGSN. Thus, while 3G cellular voice and cellular data services use different Core Networks, they share a common first/last-hop radio access network.

* UMTS Uses a CDMA technique known as a Direct Sequence wideband CDMA. (DS-WCDMA) within TDMA slots.

* TDMA slots, in turn, are available on multiple frequencies.

* Thus 3G uses all three dedicated Channel-Sharing approaches.

* The data service associated with the WCDMA specification is known as HSP (High Speed Packet Access) and Promises downlink data rates of upto 14 Mbps.



Comparison between 1G, 2G, 3G, 4G

Name	1G	2G	3G	4G
Introduced In the year	1980's	1993	2001	2009
Location of first Commercialization	USA	Finland	Japan	South korea
Technology	AMPS (Advanced Mobile phone system) NMT, TACS	Is-95, GSM	IMT-2000, WCDMA	LTE; WiMAX
Multiple Address/ Access system	FDMA	TDMA, CDMA	CDMA	OFDMA, CDMA
Switching Type	Circuit Switching	Circuit Switching for voice and packet Switching for Data	Packet Switching except for Air Interface	Packet Switching

	1G	2G	3G	4G
(Bandwidth)	2 kbps	14-64 kbps	2 Mbps	200 Mbps
Major Services	Voice	Voice Internet (Text only)	Voice Internet (Text, Images)	Voice over IP Rich Internet
Core network	Circuit Based	Circuit Based	circuit and packet Based	Fully IP Based
Advantages	Spotty Coverage	Improved voice quality	Introducting of packet data Higher data rates (up to 144 kbps)	High Data rate capability for Multi media applicabons

Limitations of 2G:

1) Strong digital signals are required to make the mobile phones work.

2) Digital signals would be weak if there is no proper network coverage in the specified area.

3) Difficult to handle complex data such as video etc.

Limitations of 3G:

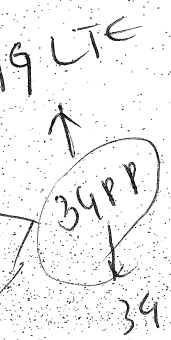
- It requires higher bandwidth
- Cost for the 3G mobile phone is high
- Size of the phone is large
- Difficult to build the infrastruce for 3G
- The amount is high for 3G Licenses Services

On 4G: LTE

The 4G Long Term Evolution (LTE) standard put forward by the 3GPP has two important innovations over 3G systems:

→ Evolved Packet Core (EPC)

→ LTE Radio Access Network



Evolved Packet core (EPC):

* The EPC is a simplified all IP core network that unifies the separate circuit switched cellular voice network and the packet switched cellular data network shown in Figure 6.19.

* It is an "all-IP" network in that both voice and data will be carried in IP datagrams.

* IP's "best effort" service model is not inherently well-suited to the stringent performance requirements of voice-over IP (VoIP) traffic unless network resources are carefully managed to avoid congestion.

* Thus, a key task of the EPC is to manage network resources to provide this high quality of service.

* The EPC also makes a clear separation between the network control and user data planes, with many of the mobility support features being implemented in the control plane.

* The EPC allows multiple types of radio access networks including legacy 2G and 3G radio access networks, to attach to the core network.

LTE Radio Access Network:

LTE uses a combination of frequency division multiplexing and time division multiplexing on the downstream channel, known as orthogonal frequency division multiplexing (OFDM)

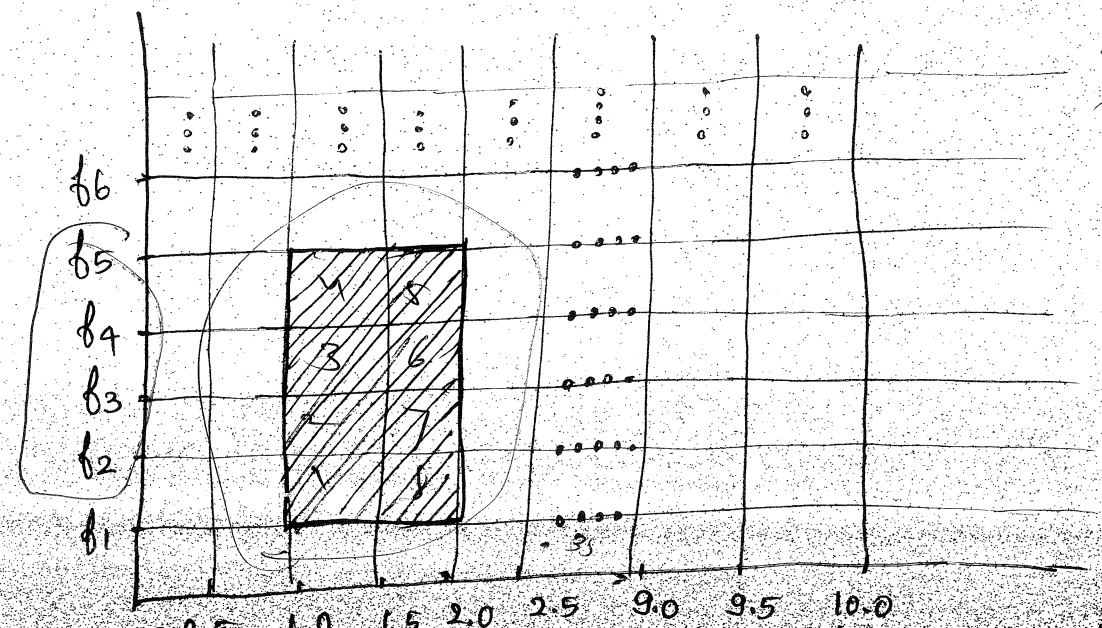
In LTE, each active mobile node is allocated one or more 0.5 ms time slots in one or more of the channel frequencies.

Figure 6.20 shows an allocation of eight time slots over four frequencies. By being allocated increasingly more time slots, a mobile node is able to achieve increasingly higher transmission rates. Slot (re) allocation among mobile nodes can be performed as often as once every milliseconds.

Different modulation schemes can also be used to change the transmission rate. Slot (re) allocation among mobile nodes can be performed as often as once every millisecond.

Another innovation in the LTE radio network is the use of sophisticated multiple-input, multiple output (MIMO) antennas.

The Maximum data rate for an LTE user is 100 Mbps in the downstream direction and 50 Mbps in the upstream direction, when using 20 MHz worth of wireless spectrum.



Twenty 0.5 ms slots organized into 10ms frames at each ... slot allocation is shown shaded.

Mobility Management: Principles

A Mobile node is one that changes its point of attachment into the network over time. Because the term mobility has taken on many meanings in both the Computer and telephony worlds, it will serve us well first to consider several dimensions of mobility in some detail.

From the Network layer's standpoint, how mobile is a User? A physically mobile User will present a very different set of challenges to the network layer, depending on how he or she moves between points of attachment to the network. At one end of the spectrum in figure 6.21, a User may carry a laptop with a wireless network interface Card around in a building.

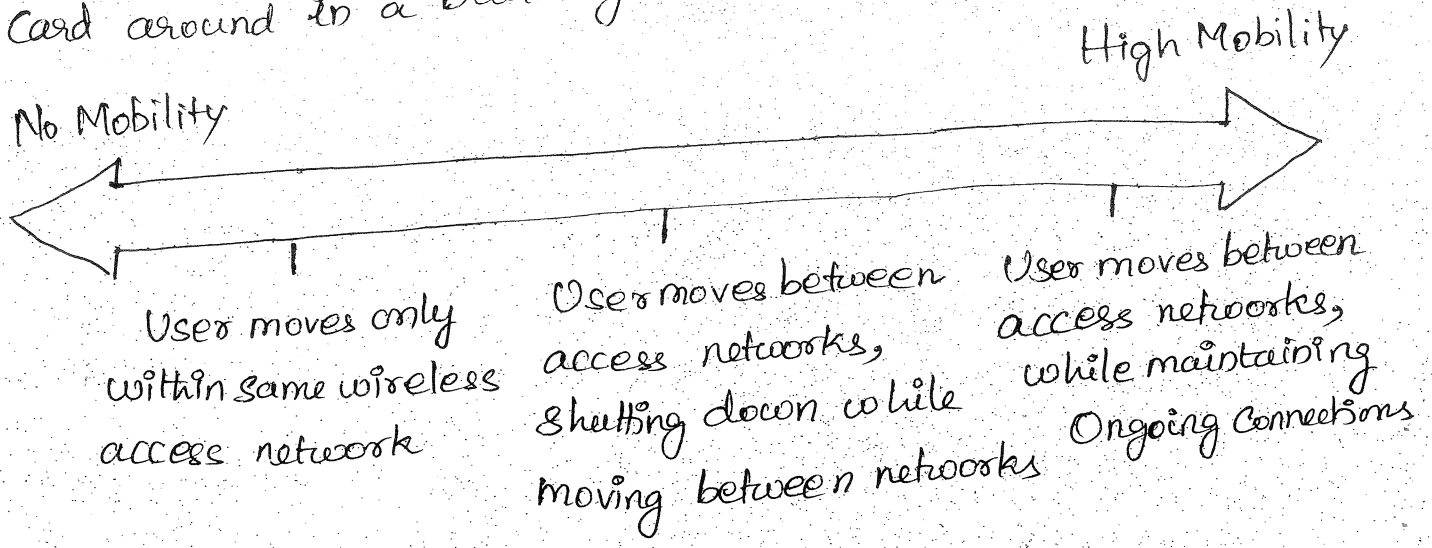


Figure 6.21. Various degrees of mobility, from the network layer's point of view

How important is it for the mobile node's address to always remain the same?

With mobile telephony, your phone number - essentially the network layer address of your phone - remains the same as you travel from one provider's mobile phone network to another. Must a laptop similarly maintain the same IP address while moving between IP networks?

The answer to this question will depend strongly on the applications being run. For the BMW driver who wants to maintain an uninterrupted TCP connection to a remote application while zipping along the autobahn, it would be convenient to maintain the same IP address. Internet application needs to know the IP address and port number of the remote entity with which it is communicating. If a mobile entity is able to maintain its IP address as it moves, mobility becomes invisible from the application standpoint. There is great value to this transparency - an application need not be concerned with a potentially changing IP address, and the same application code serves mobile IP provides this transparency, allowing a mobile node to maintain its permanent IP address while moving among networks.

On the other hand, a less glamorous mobile user might simply want to turn off an office laptop, bring that laptop home, power up, and work from home. If the laptop functions primarily as a client in client-server applications (e.g. send/read e-mail, browse the Web, Telnet to a remote host) from home, the particular IP address used by the laptop is not that important. In particular one could get by fine with an address that is temporarily allocated to the laptop by the ISP serving the home. DHCP provides this functionality.

- What supporting wired infrastructure is available? In all of our scenarios above, we've implicitly assumed that there is a fixed infrastructure to which the mobile user can connect — for ex, the home ISP's network, the wireless access network in the office or the wireless access networks lining the autobahn. What if no such infrastructure exists? If two users are within communication proximity of each other, can they establish a network connection in the absence of any other network layer infrastructure? Adhoc networking provides precisely these capabilities. This rapidly developing area is at the cutting edge of mobile networking research and is beyond the scope of this book.

In order to illustrate the issues involved in allowing a mobile user to maintain ongoing connections while moving between networks, let's consider a human analogy. A twenty-something adult moving out of the family home becomes mobile, living in a series of dormitories and/or apartments and often changing addresses. If an old friend wants to get in touch, how can that friend find that address of her mobile friend? One common way is to contact the family. Since mobile adult will often register his or her current address with the family (if for no other reason than so that the parents can send money to help pay the rent!) The family home, with its permanent address, becomes that one place that others can go as a first step in communicating with the mobile adult. Later communication from the friend may be either indirect (for example, with mail being sent first to the parent's home and then forwarded to the mobile adult) or direct (for example with the friend using the address obtained from the parents to send mail directly to her mobile friend)

In a network setting, the permanent home of a mobile node (such as a laptop or Smart phone) is known as home network, and the entity within the home network that performs the mobility management functions discussed below on behalf of the mobile node is known as the home agent.

The network in which the mobile node is currently residing is known as the foreign (or visited) network, and the entity within the foreign network that helps the mobile node with the mobility management functions discussed below is known as a foreign agent. For mobile professionals, their home network might likely be their company network, while the visited network might be the network of a colleague they are visiting. A correspondent is the entity wishing to communicate with the mobile node. Figure 6.22 illustrates these concepts, as well as addressing concepts considered below. Agents are shown as being collocated with routers (e.g. as processes running on routers) but alternatively they could be executing on other hosts or servers in the network.

Addressing

When a mobile node is resident in a foreign network, all traffic addressed to the node's permanent address now needs to be routed to the foreign network. How can this be done? One option is for the foreign network to advertise to all other networks that the mobile node is resident in its network. This could be via the usual exchange of intradomain and interdomain routing information and would require few changes to the existing routing infrastructure.

The foreign network could simply advertise to its neighbors that it has a highly specific route to the mobile node's permanent address. These neighbors would then propogate this routing information throughout the network as part of the normal procedure of updating routing information and forwarding tables. When the mobile node leaves one foreign network and joins another, and new foreign network would advertise a new, highly specific route to the mobile node, and the old foreign network would withdraw its routing information regarding the mobile node.

This solves two problems at once, and it does so without making significant changes to the network-layer infrastructure. Other networks know the location of the mobile node, and it is easy to route datagrams to the mobile node, since the forwarding tables will direct datagrams to the foreign network. A significant drawback, however, is that of scalability. If mobility management were to be the responsibility of network routers, the routers would have to maintain forwarding table entries for potentially millions of mobile nodes, and update these entries as nodes move. Some additional drawbacks are explored in the problems at the end of this chapter.

An alternative approach is to push mobility functionality from the network core to the network edge — a recurring theme in our study of Internet architecture.

(+)
A Natural way to do this is via the mobile node's home network.

Let's now consider the foreign agent in more detail. The conceptually simplest approach, shown in figure 6.22, is to locate foreign agents at the edge routers in the foreign network. One role of the foreign agent is to create so called care-of-address (COA) for the mobile node, with the network portion of the COA matching that of the foreign network. There are thus two addresses associated with a mobile node, its permanent address and its COA. Sometimes known as foreign address.

In the Example in figure 6.22, the permanent address of the mobile is 128.119.40.186 when visiting network 79.129.13/24 the mobile node has a COA of 79.129.13.2. A second role of the foreign agent is to inform the home agent that the mobile node is resident in its (the foreign agent's) network and has given COA. COA will be used to route "reroute" datagrams to the mobile node via its foreign agent.

Although we have separately the functionality of the mobile node and the foreign agent, it is worth noting that the mobile node can assume the responsibilities of the foreign agent. For Exp the mobile node could obtain a COA in the foreign network and itself inform the home agent of its COA.

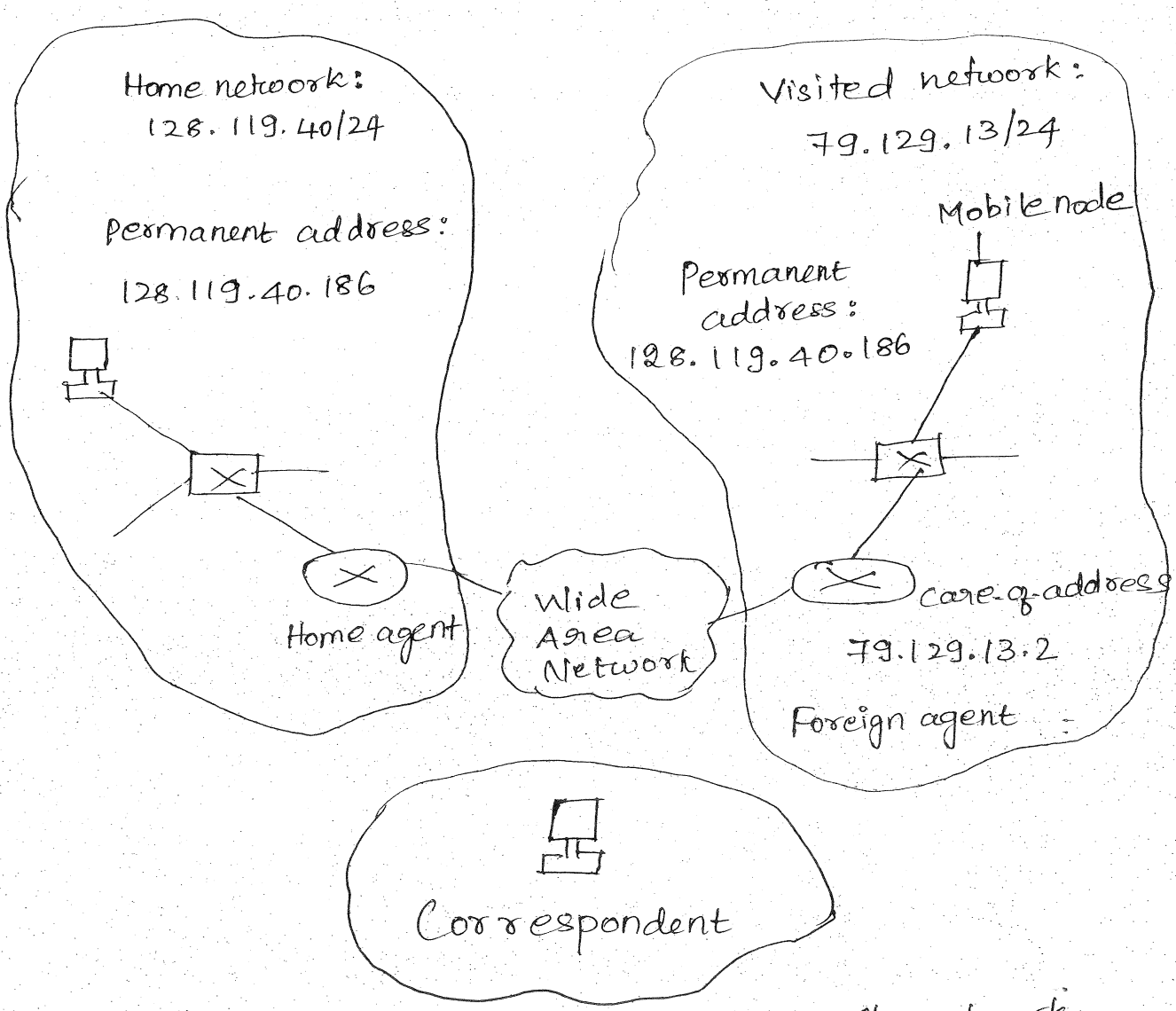


Figure 6.22. Initial element of a mobile network architecture

Routing to a Mobile Node

How should datagrams be addressed and forwarded to the mobile node? Since only the home agent knows the location of the mobile node, it will no longer suffice to simply address a datagram to the mobile node's permanent address and send it into the network layer infrastructure. Something more must be done. Two approaches can be identified which we will refer to as indirect

and direct routing.

Indirect Routing to a Mobile Node

The correspondent simply addresses the datagram to the mobile node's permanent address and sends the datagram into the network, blissfully unaware of whether the mobile node is resident in its home network or is visiting a foreign network; mobility is thus completely transparent to the correspondent. Such datagrams are first routed, as usual to the mobile node's home network.

Let's now turn our attention to the home agent. In addition to being responsible for interacting with a foreign agent to track the mobile node's COA, the home agent has another very important function. Its second job is to be on the lookout for arriving datagrams addressed to nodes whose home network is that of the home agent but that are currently resident in a foreign network. The home agent intercepts these datagrams and then forwards them to a mobile node in a two-step process. The datagram is first forwarded to the foreign agent, using the mobile node's COA (step 2 in figure 6.23) and then forwarded from the foreign agent to the mobile node (step 3 in figure 6.23).

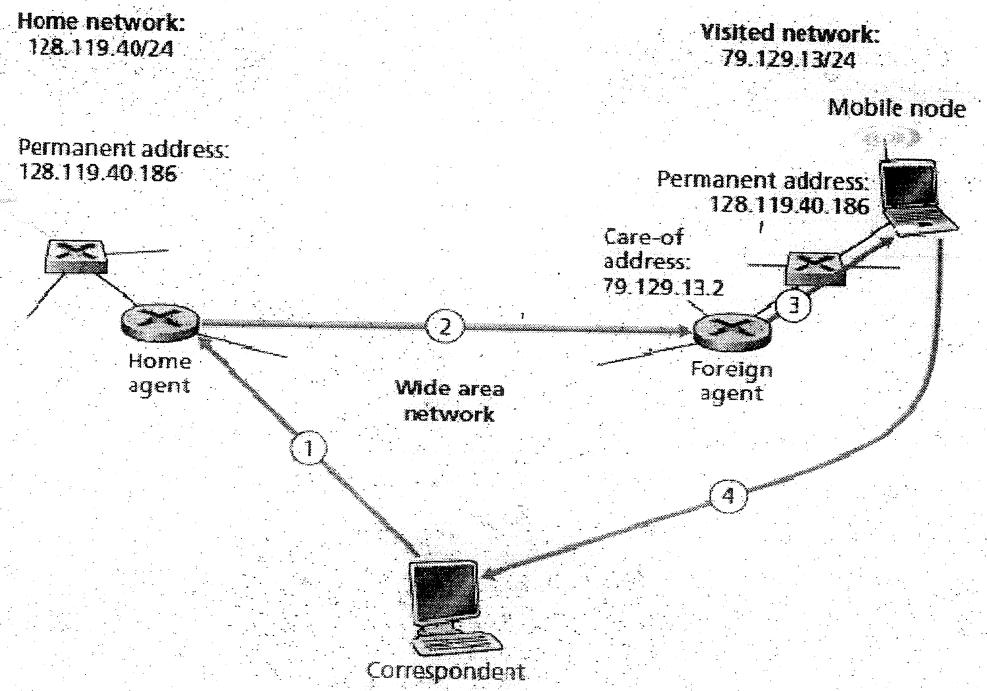


Figure 6.23 • Indirect routing to a mobile node

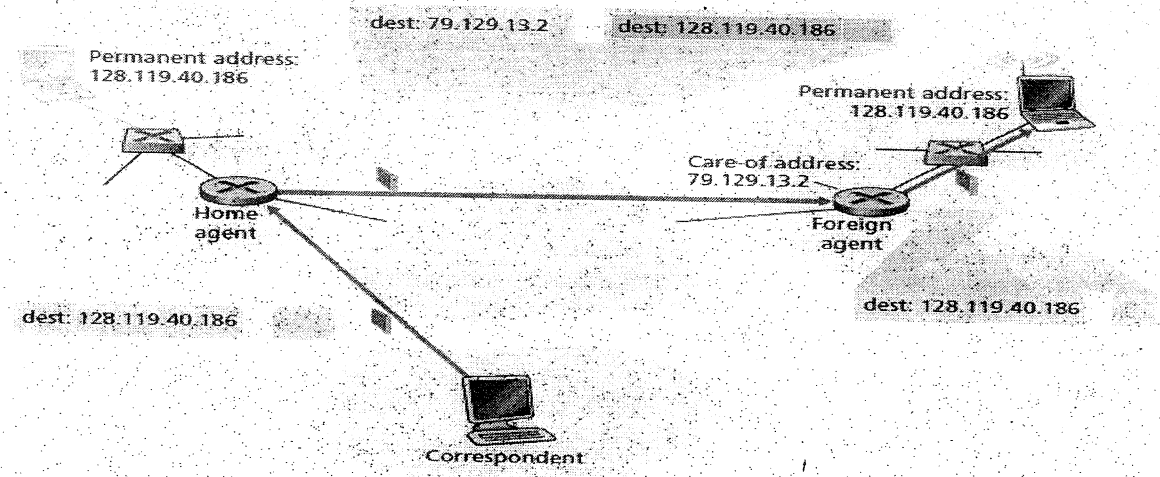


Figure 6.24 • Encapsulation and decapsulation

It is instructive to consider this rerouting in more detail. The home agent will need to address the datagram using the mobile node's CoA, so that the network layer will route the datagram to the foreign network. On the other hand, it is desirable to leave the correspondent's datagram intact, since the application receiving the datagram should be unaware that the datagram was forwarded via the home agent. Both goals can be satisfied by having the home agent encapsulate the correspondent Original complete datagram within a new (larger) datagram. This larger datagram is addressed and delivered to the mobile node's CoA. The foreign agent, who "owns" the CoA, will receive and decapsulate the datagram - that is, remove the correspondent's original datagram from within the larger encapsulating datagram and forward the original datagram to the mobile node. Figure 6.24 shows a correspondent original datagram being sent to home network, an encapsulated datagram being sent to the foreign agent, and the original datagram being delivered to the mobile node. The sharp reader will note that the encapsulating / decapsulation described here is identical to the notion of tunneling, in the context of the IP multicast and IPV6.

Let's next consider a how a mobile node sends datagrams to correspondent. This is quite simple, as the mobile node can address its datagram directly to the correspondent. Since the mobile node knows the correspondent's address, there is no need to route the datagram back through the home agent.

A Mobile-node to foreign agent Protocol. The Mobile node will register with the foreign agent when attaching to the foreign network. Similarly, a mobile node will deregister with the foreign agent when it leaves the foreign network.

A foreign agent to home agent registration protocol. the foreign agent will register the mobile node's COA with the home agent. A foreign agent need not explicitly deregister a COA when a mobile node leaves its network, because the subsequence registration of a new COA, when the mobile node moves to a new network, will take care of this.

A home-agent datagram encapsulation protocol. Encapsulation and forwarding of the correspondent's Original datagram within a datagram addressed to COA.

Direct Routing to a Mobile Node

The indirect routing approach illustrated in figure 6.23 suffers from an inefficiency known as the triangle routing problem – datagrams addressed to the mobile node must be routed first to the home agent and then to the foreign network, even when a much more efficient route exists between the correspondent and the mobile node. In the worst case, imagine a mobile user who is visiting the foreign network of a colleague. The two are sitting side by side and exchanging data over the network. Data grams from the correspondent are routed to the mobile user's home agent and then back again to the foreign network.

Direct routing overcomes the inefficiency of triangle routing, but does so at the cost of additional complexity. In the direct routing approach, a correspondent agent in the correspondent's network first learns the COA of the mobile node. This can be done by having the correspondent agent query the home agent, assuming that the mobile node has an up-to-date value for its COA registered with its home agent. It is also possible for the correspondent itself to perform the function of the correspondent agent, just as a mobile node could perform the functions of the foreign agent.

The correspondent agent then tunnels datagrams directly to the mobile node's COA, in a manner analogous to the tunneling performed by the home agent, steps 3 and 4 in figure 6.25.

While direct routing overcomes the triangle routing problem, it introduces two important additional challenges:

- A mobile user location protocol is needed for the correspondent agent to query the home agent to obtain the mobile node's COA.
- When the mobile node moves from one foreign network to another, how will data now be forwarded to the new foreign network? In the case of indirect routing, this problem was easily solved by updating the COA maintained by the home agent. However, with direct routing, the home agent is queried for the COA by the correspondent agent only once, at the beginning of the session. Thus, updating the COA at the home agent, while necessary, will not be enough to solve the problem of routing data to the mobile node's new foreign network.

One solution would be to create a new protocol to notify the correspondent of the changing COA. An alternate solution, and one that we'll see adopted in practice in GSM networks, works as follows,

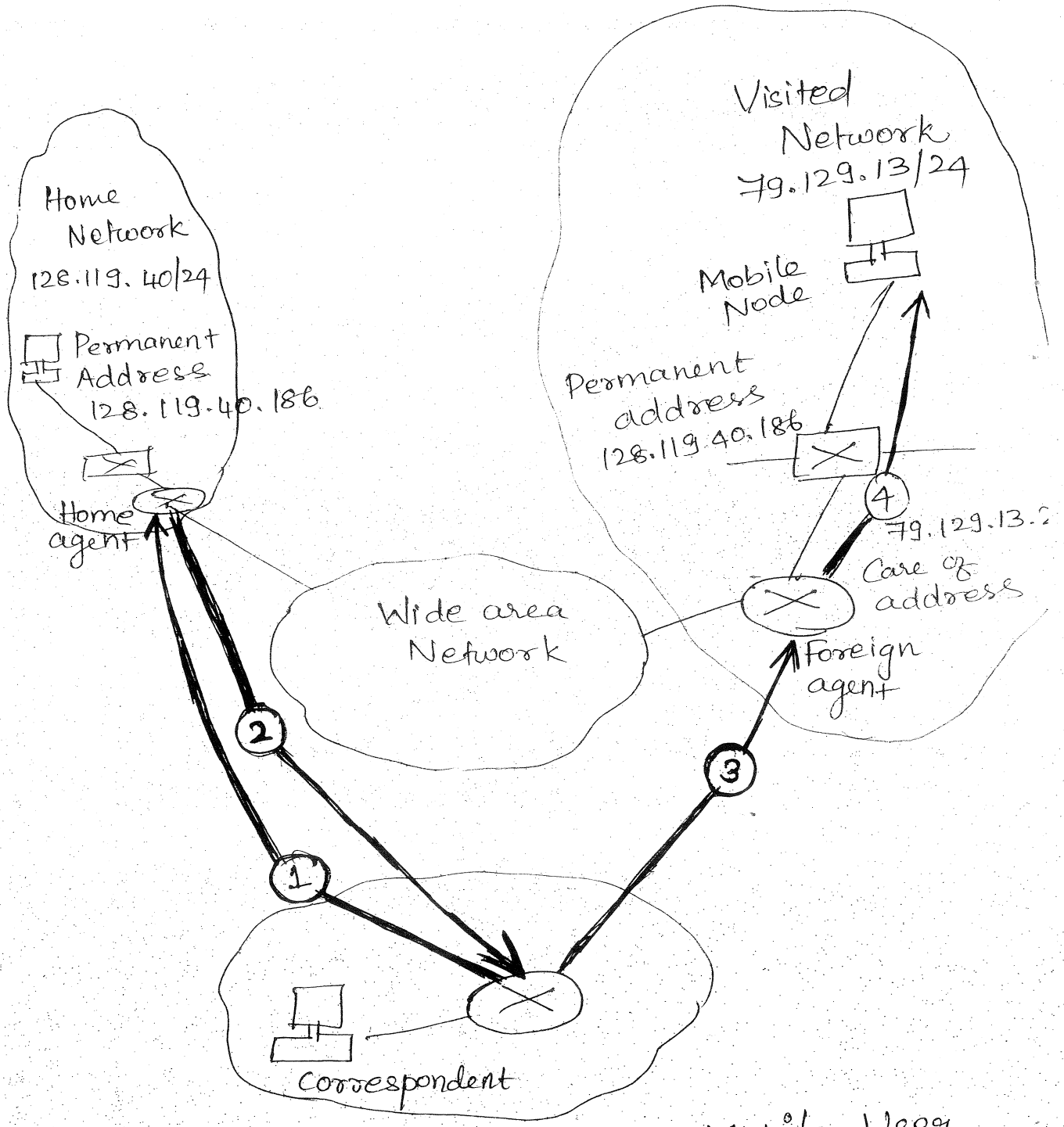


Figure 6.25. Direct Routing to a Mobile User.

Suppose data is currently being forwarded to the mobile node in the foreign network where the mobile node was located when the session first started. We'll identify the foreign agent in that foreign network where the mobile node was first found as the anchor foreign agent. When the mobile node moves to a new foreign network the mobile node registers with the new foreign agent and the new foreign agent provides the anchor foreign agent with the mobile node's new COA. When the anchor foreign agent receives an encapsulated datagram for a departed mobile node, it can then decapsulated the datagram and forward it to the mobile node (step 5) using the new COA. If the mobile node later moves yet again to a new foreign network, the foreign agent in that new visited network would then contact the anchor foreign agent in order to set up forwarding to this new foreign network.

Mobile IP

The Internet architecture and protocols for supporting mobility, collectively known as mobile IP, are defined primarily in RFC 5944 for IPV4. Mobile IP, is a flexible standard, supporting many different modes of operation (for example operation with or without foreign agent), multiple ways for agents and mobile ~~agent~~ nodes to discover each other, use of single or

multiple COAs, and multiple forms of encapsulation. As such, mobile IP is a complex standard. One modest goal here is to provide an Overview of the most important aspects of mobile IP and to illustrate its use in a few common case scenarios.

The Mobile IP architecture contains many of the elements including

- home agents
- foreign agents
- care-of address
- encapsulation/decapsulation.

The current standard (RFC 5944) specifies the use of indirect routing to the mobile node.

The Mobile IP standard consists of three main pieces:

1) Agent discovery: Mobile IP defines the protocols used by a home or foreign agent to advertise its services to mobile nodes, and protocols for mobile nodes to solicit the services of a foreign or home agent.

2) Registration with the home agent. Mobile IP defines the protocols used by the mobile node and/or foreign agent to register and deregister COAs with a mobile node's home agent.

Indirect routing of datagrams. The standard also defines the manner in which datagrams are forwarded to mobile nodes by a home agent, including rules for forwarding datagrams, rules for handling error conditions and several forms of encapsulation.

Agent Discovery

A Mobile IP node arriving to a new network, whether attaching to a foreign network or returning to its home network, must learn the identity of the corresponding foreign or home agent. Indeed, it is the discovery of a new foreign agent, with a new network address, that allows the network layer in a mobile node to learn that it has moved into a new foreign network. This process is known as agent discovery. Agent discovery can be accomplished in one of two ways: Via agent advertisement or via agent solicitation.

With agent advertisement, a foreign or home agent advertises its services using an extension to the existing router discovery protocol [RFC 1256]. The agent periodically broadcasts an ICMP message with a type field of 9 (router discovery) on all links to which it is connected. The router discovery message contains the IP address of the router thus allowing a mobile node to learn the agent's IP address. The router discovery message also contains a mobility agent advertisement extension

that Contains additional information needed by the mobile node. Among the more important fields in the extension are the following;

- Home agent bit (H): Indicates that the agent is a home agent for the network in which it resides.
- Foreign agent bit (F): Indicates that the agent is a foreign agent for the network in which it resides.
- Registration required bit (R): Indicates that a mobile user in this network must register with a foreign agent. In particular, a mobile user cannot obtain a care-of-address in the foreign network and assume the functionality of the foreign agent for itself, without registering with the foreign agent.
- M, G encapsulation bits. Indicate whether a form of encapsulation other IP-in-IP encapsulation will be used.
- care-of-address (COA) fields. A list of one or more care-of-addresses provided by the foreign agent. In our example below, the COA will be associated with the foreign agent, who will receive datagrams sent to the COA and then forward them to the appropriate mobile node. The mobile user will select one of these addresses as its COA when registering with its home agent.

With agent Solicitation, a mobile node wanting to learn about agents without waiting to receive an agent advertisement can broadcast an agent solicitation message, which is simply an ICMP message with type value 10. An agent receiving the solicitation will unicast an agent advertisement directly to the mobile node, which can then proceed as if it had received an unsolicited advertisement.

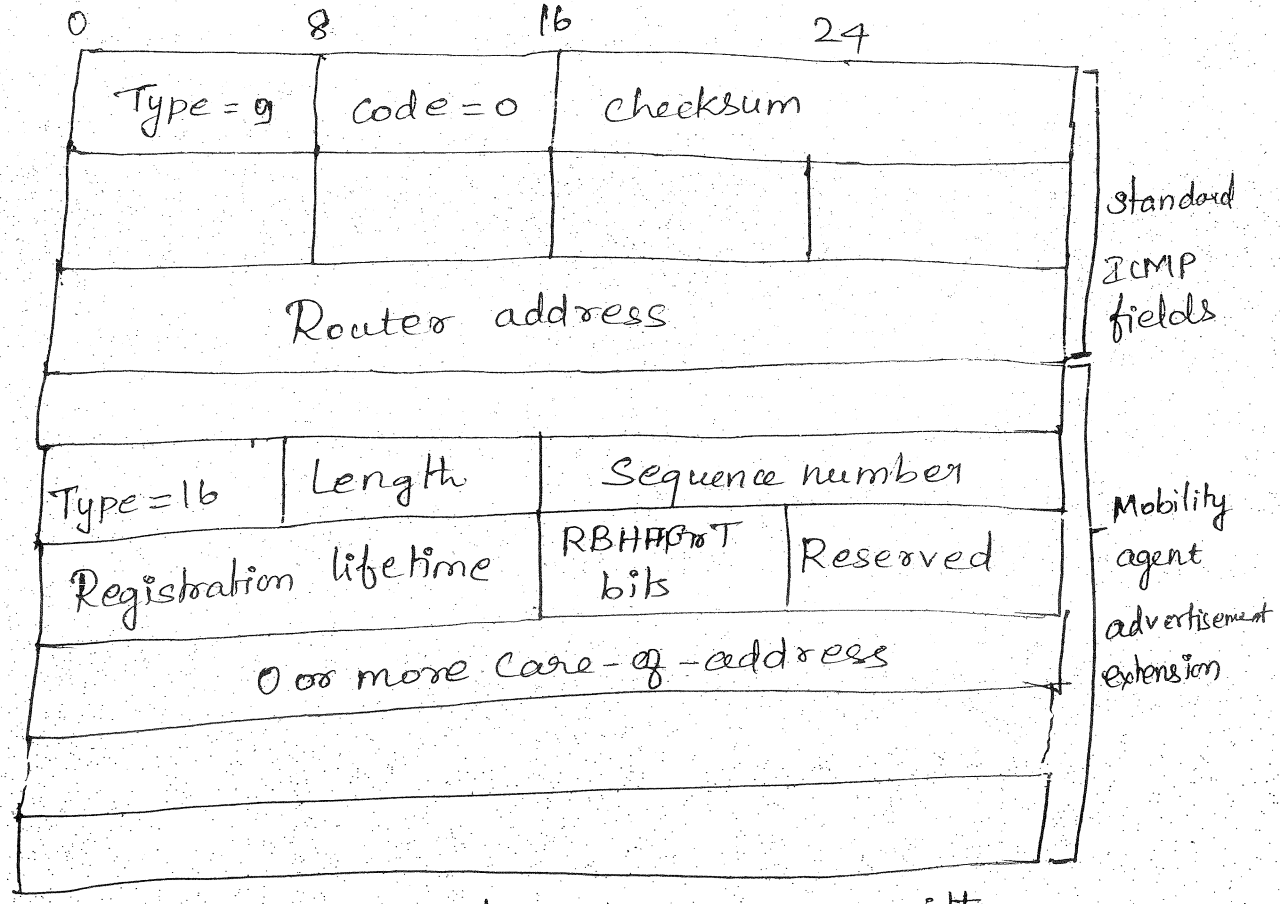


Figure 6.27. ICMP router discovery message with mobility agent advertisement extension.

Registration with the Home agent

Once a mobile IP node has received a CoA, that address must be registered with the home agent. This can be done either via the foreign agent or directly by the mobile IP node itself.

Four steps are involved.

1] Following the receipt of a foreign agent advertisement a mobile node sends a mobile IP registration message to the foreign agent. The registration message is carried within a UDP datagram and sent to port 434. The registration message carries a COA advertised by the foreign agent, the address of the home agent (HA) the permanent address of the mobile node (MA), the requested lifetime of the registration, and a 64 bit registration identification. The requested registration lifetime is the number of seconds that the registration is to be valid. If the registration will become invalid. The registration identifier acts like a sequence number and serves to match a received registration reply with a registration request, as discussed below.

2] The foreign agent receives the registration message and records the mobile node's permanent IP address. The foreign agent now knows that it should be looking for datagrams containing an encapsulated datagram whose destination address matches the permanent address of the mobile node. The foreign agent then sends a mobile IP registration message (again, within a UDP datagram) to port 434 of the home agent. The message contains the COA, HA, MA, encapsulation format requested registration lifetime, and

registration identification.

3] The home agent receives the registration request and checks for authenticity and correctness. The home agent binds the mobile node's permanent IP address with the COA; in the future, datagrams arriving at the home agent and addressed to the mobile node will now be encapsulated and tunneled to the COA. The home agent sends a mobile IP registration reply containing the HA, MA actual registration lifetime, and the registration identification of the request that is being satisfied with this reply.

4] The foreign agent receives the registration reply and then forwards it to the mobile node.

At this point, registration is complete, and the mobile node can receive datagrams sent to its permanent address. Figure 6.28 illustrates these steps. Note that the home agent specifies a lifetime that is smaller than a lifetime requested by the mobile node.

A foreign agent need not explicitly deregister a COA when a mobile node leaves its network. This will occur automatically, when the mobile node moves to a new network and registers new COA.

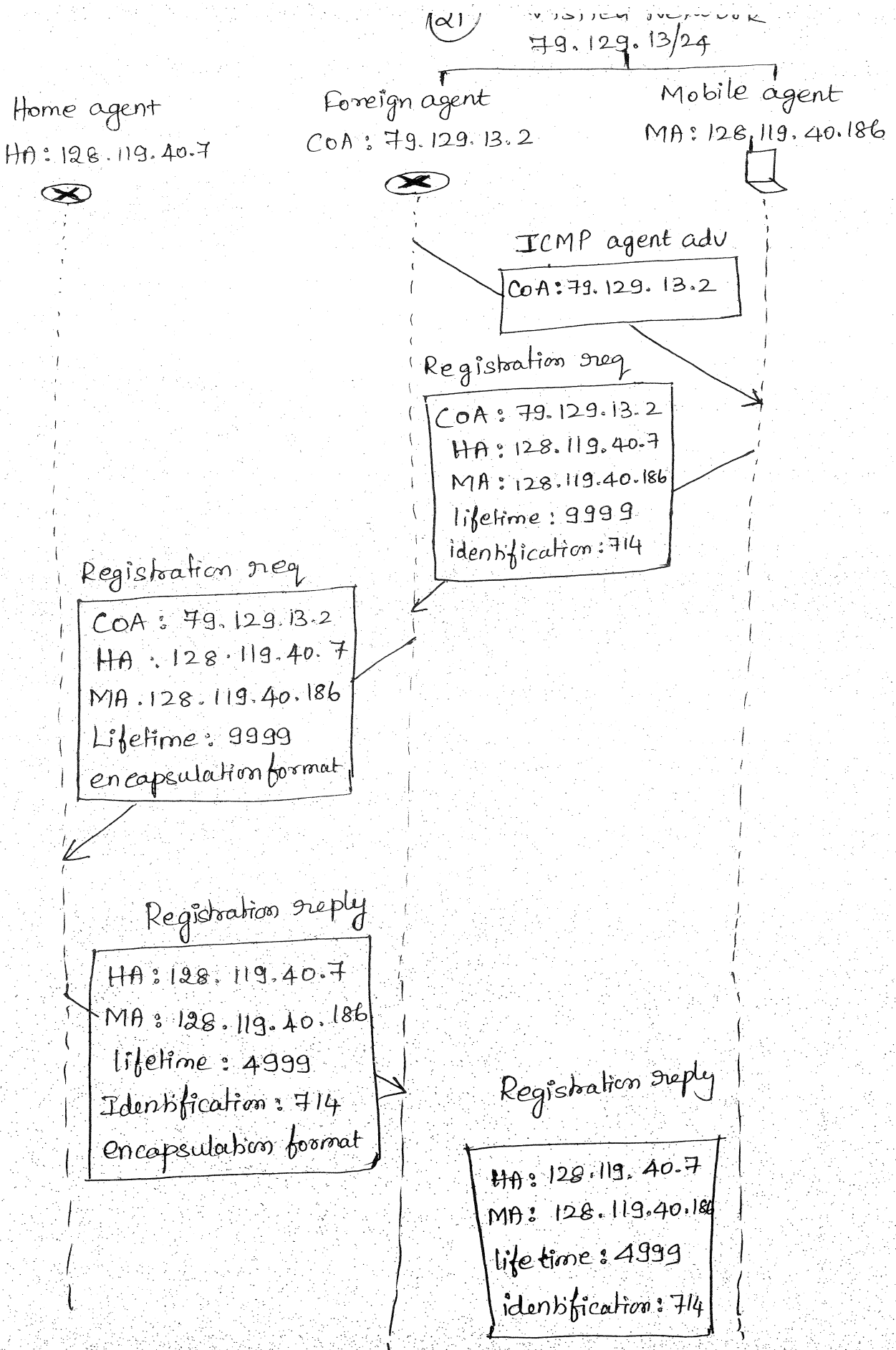


Figure 6.28. Agent advertisement and mobile IP registration.

6.7. Managing Mobility in Cellular Networks

Like mobile IP, GSM adopts an indirect routing approach first routing the correspondent's call to the mobile user's home network and from there to the visited network. In GSM terminology, the mobile user's home network is referred to as the mobile user's home public Land mobile network (home PLMN). Since PLMN acronym is a bit of a mouthful, and mindful of our quest to avoid an alphabet soup of acronyms, GSM home PLMN simply as the home Network. The home Network is the cellular provider with which the mobile user has a subscription. (i.e., the provider that bills the user for monthly cellular service). The Visited PLMN, refer to simply as the visited network, is the network in which the mobile user is currently residing.

Responsibilities of the home and visited networks are

1) The home network maintains a database known as the home location register (HLR), which contains the permanent cell phone number and subscriber profile for each of its subscriber. Importantly the HLR also contains information about the current locations of these subscribers. That is, if a mobile user is currently roaming in another provider's cellular network, the HLR contains enough information to obtain (via a process) an address in the visited

to which a call to the mobile user should be routed. Special switch in the home network, known as the Gateway Mobile Services Switching Center (GMSC) is contacted by a correspondent when a call is placed to a mobile user. Again, in our quest to avoid an alphabet soup of acronyms, home MSC.

The Visited Network maintains a database known as the visitor location register (VLR). The VLR contains an entry for each mobile user that is currently in the portion of the network served by the VLR. VLR entries thus come and go as mobile users enter and leave the network. A VLR is usually co-located with the mobile switching center (MSC) that co-ordinates the setup of a call to and from the visited network.

6.7.1. Routing Calls to a Mobile User

How a call is placed to a mobile GSM user in a Visited Network. The steps, as illustrated in figure.

1. The correspondent dials the mobile user's phone number. This number itself does not refer to a particular telephone line or location. The leading digits in the number are sufficient to globally identify the mobile's home network. The call is routed from the Correspondent through the PSTN to the home MSC in the mobile's home network. This is the first leg of the call.

2] The home MSC receives the call and interrogates the HLR to determine the location of the mobile user. In the simplest case, the HLR returns the mobile station roaming number (MSRN), which we will refer to as the roaming number. Note that this number is different from the mobile's permanent phone number which is associated with the mobile's home network. The roaming number is ephemeral. It is temporarily assigned to a mobile when it enters a visited Network. The roaming Number serves a role similar to that of the care-of-address in mobile IP and like the CoA, is invisible to the correspondent and the mobile. (cont

If HLR does not have the roaming number, it returns the address of the VLR in the visited network. In this case, the home MSC will need to query the VLR to obtain the roaming number of the mobile node.

3] Given the roaming number, the Home MSC setup the second leg of the call ~~through the network to the~~ MSC sets up the second leg of the call to the MSC in the Visited Network. The call is completed, being routed from the correspondent to the home MSC, and from there to the visited MSC, and from there to the base station serving the mobile user.

Correspondent → Home MSC → Visited MSC → BS → Mobile

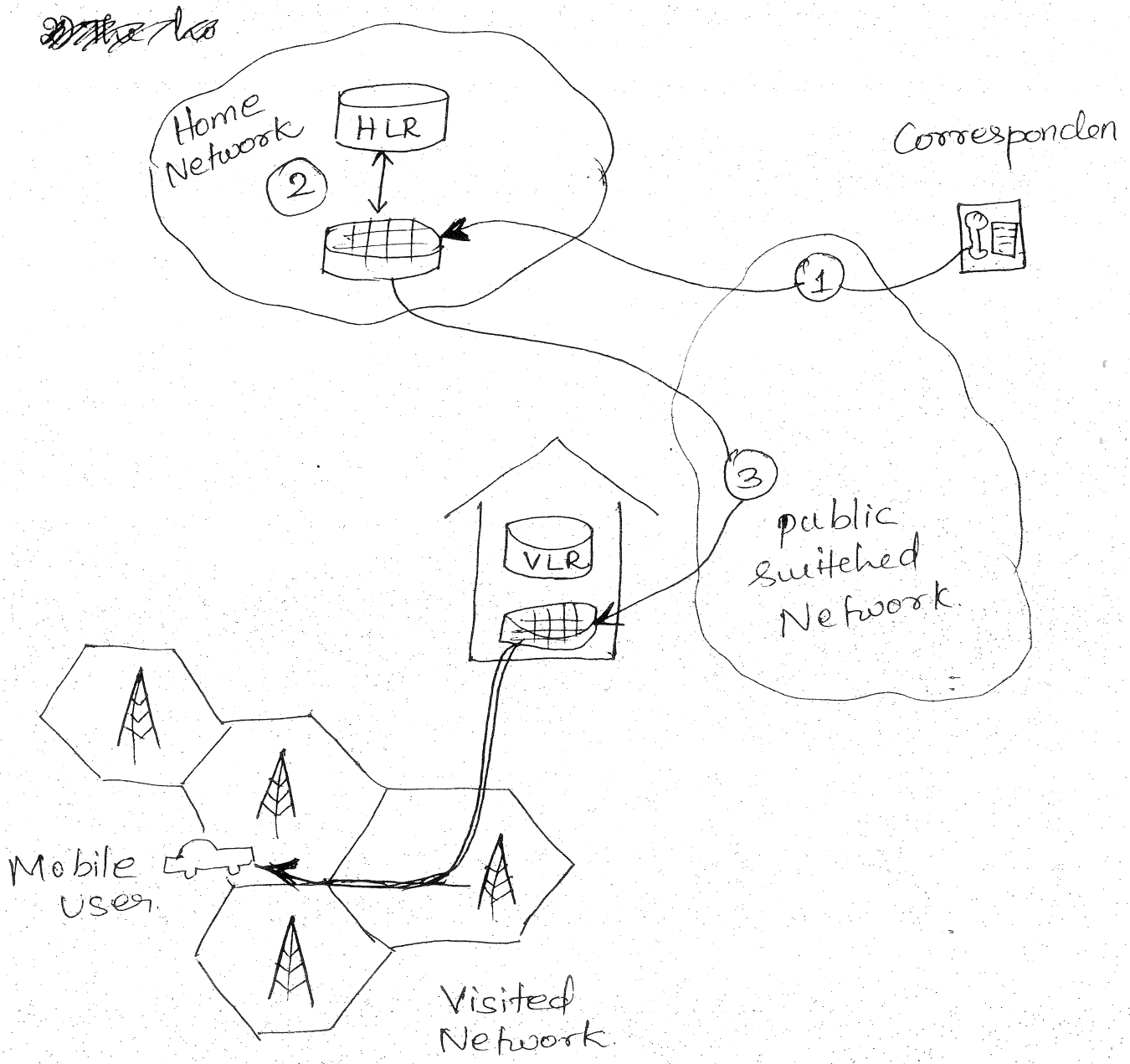


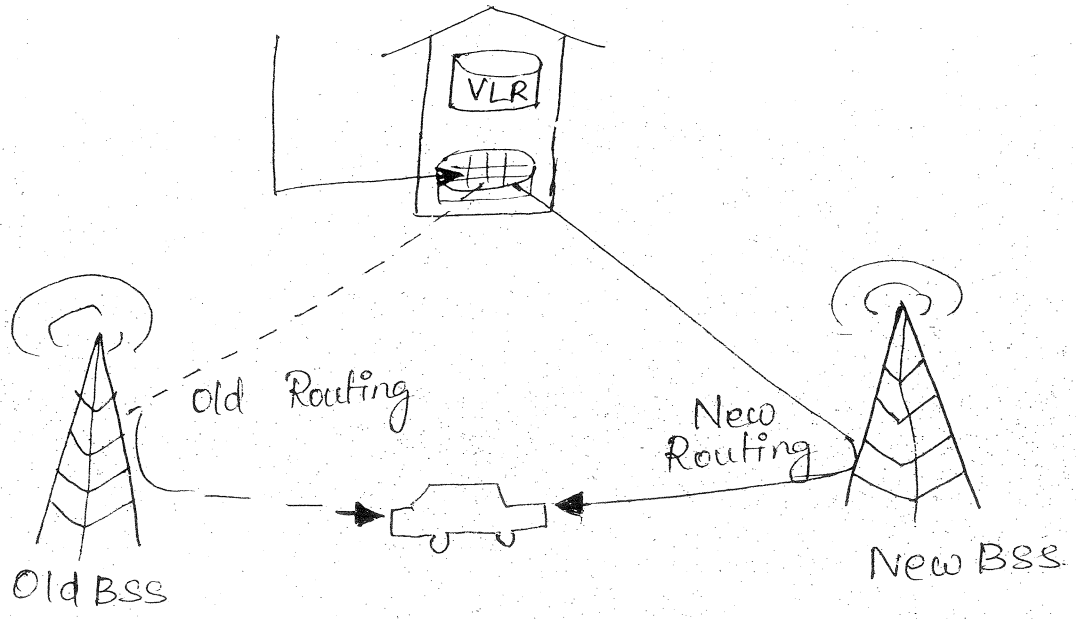
Figure 6.29. placing a call to a Mobile User

Indirect routing

Corr – HLR – Public switched n/w – VLR – Mobile User

6.4.2. Handoffs in GSM

A handoff occurs when a mobile station changes its association from one base station to another during a call.



Handoff scenario between base stations with a common MSC.

As shown in figure, a mobile's call is initially (before handoff) routed to the mobile through one base station (which we'll refer to as the old base station), and after handoff is routed to the mobile through another base station (which we'll refer to as the new base station). Note that a handoff between base stations results not only in the mobile transmitting/receiving to/from a new base station, but also in the rerouting of the ongoing call from a switching point within the networks to the new base station. Let's initially

assume that the old and new base stations share the same MSC, and that the rerouting occurs at this MSC.

There may be several reasons for handoff to occur, including

1) The signal between the current base station and the mobile may have deteriorated to such an extent that the call is in danger of being dropped,

2) A cell may have become overloaded, handling a large number of calls. This congestion may be alleviated by handing off mobiles to less congested nearby cells.

Figure 6.31 illustrates the steps involved when a base station does decide to handoff a mobile user:

1) The old base station (BS) informs the visited MSC that a handoff is to be performed and the BS (or possible set of BSs) to which the mobile is to be handed off.

2) The visited MSC initiates path setup to the new BS, allocating the resources needed to carry the rerouted call, and signaling the new BS that a handoff is about to occur.

3) The new BS allocates and activates a radio channel for use by the mobile.

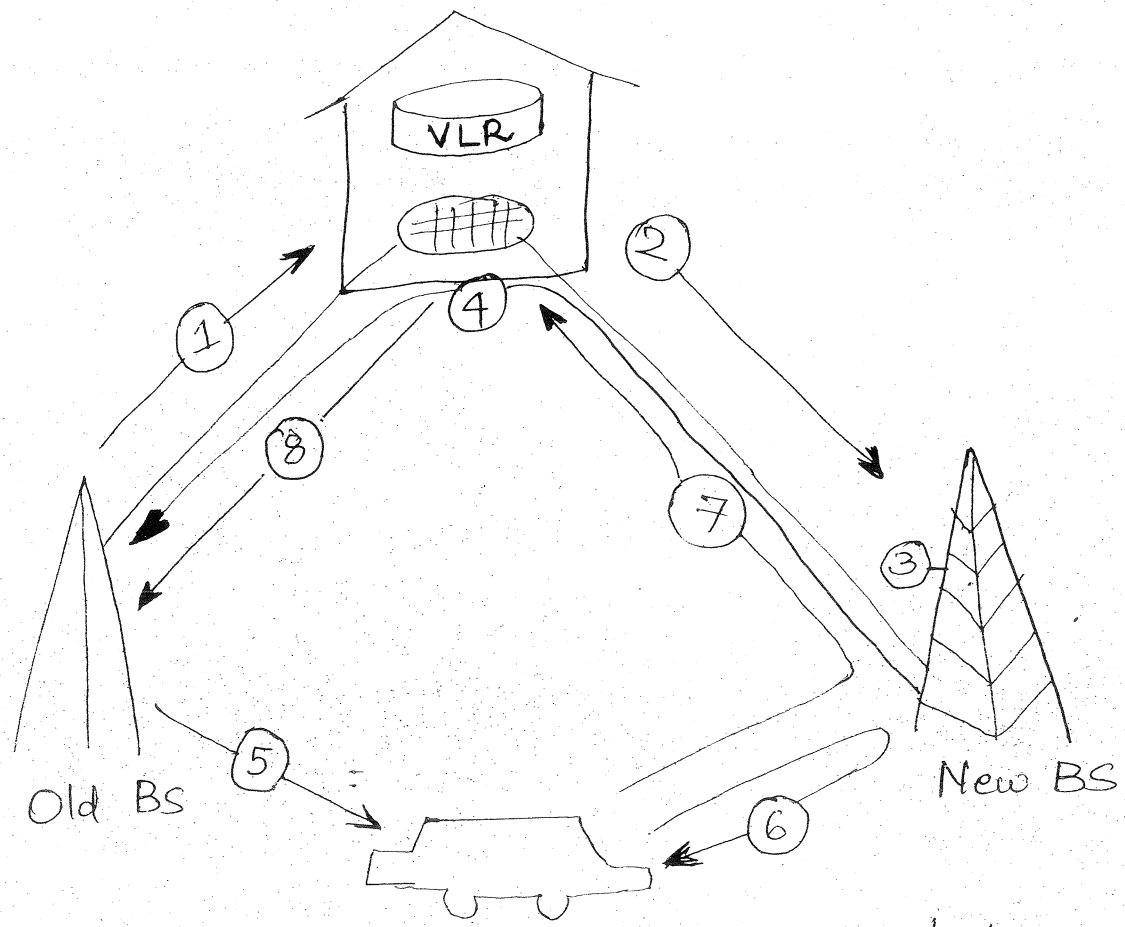
4) The new BS signals back to the visited MSC and the old BS that the visited MSC-to-new BS path has been established and that the mobile should be informed of the impending handoff. The new BS provides all of the information that the mobile will need to associate with the new BS.

5) The mobile is informed that it should perform a handoff. Note that up until this point, the mobile has been blissfully unaware that the network has been laying the groundwork. (e.g. allocating a channel in the new BS and allocating a path from the visited MSC to the new BS) for a hand off.

6) The mobile and the new BS exchange one or more messages to fully activate the new channel in the new BS.

7) The mobile sends a handoff complete message to the new BS, which is forwarded up to the visited MSC. The visited MSC then reroutes the ongoing call to the mobile via the new BS.

8) The resources allocated along the path to the old BS are then released.



Steps in accomplishing a handoff between base stations with a common MSC.

Let's conclude our discussion of handoff by considering what happens when the mobile moves to a BS that is associated with a different MSC than the Old BS, and what happens when this inter-MSC handoff occurs more than once.

As shown in figure 6.32 GSM defines the notion of an anchor MSC. The anchor MSC is the MSC visited by the mobile when a call first begins; the anchor MSC thus remains unchanged during the call. Throughout the call's duration and regardless of the number of inter-MSC transfers performed by the mobile, the call is

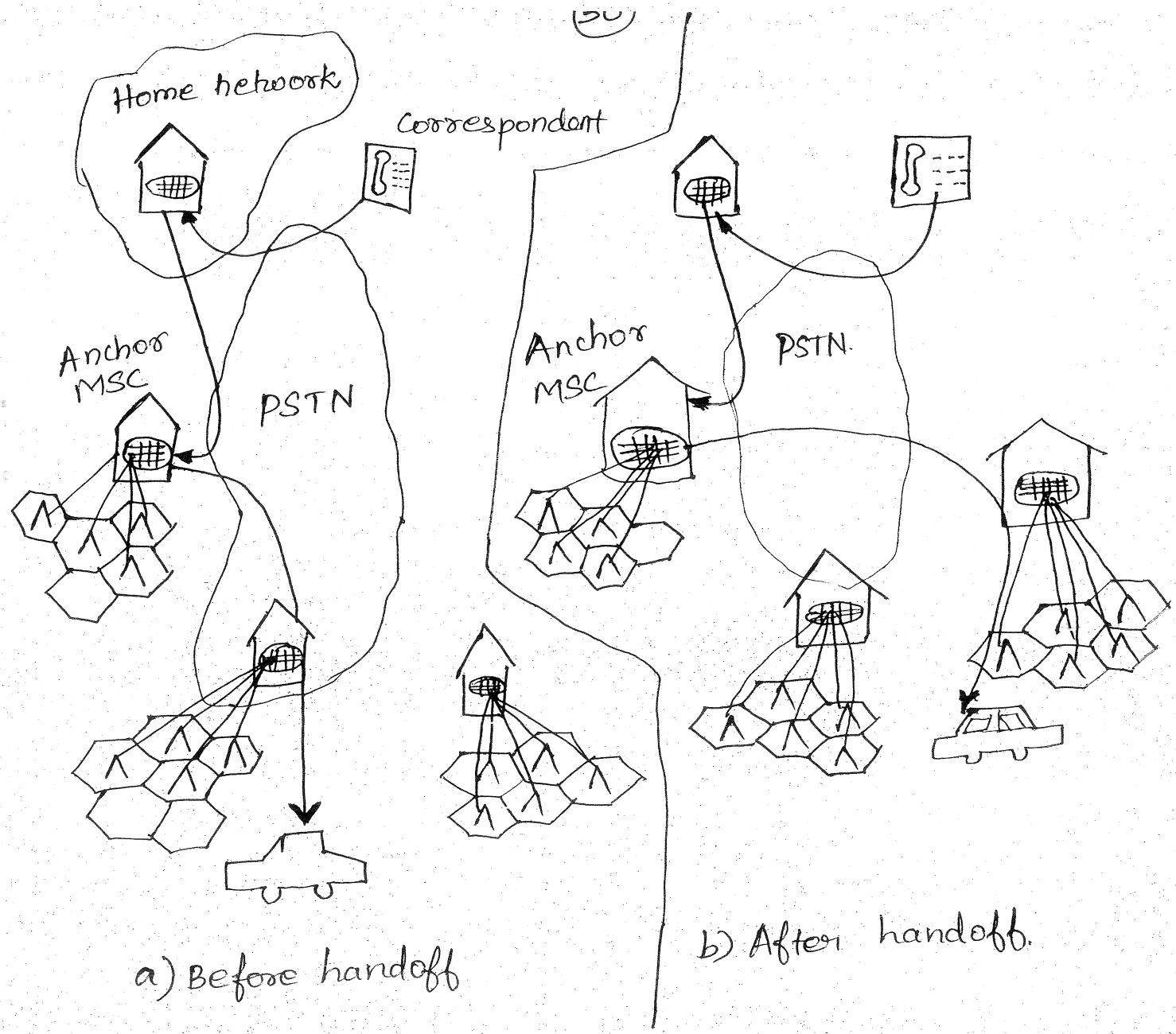


Figure 6.32 Rerouting via the anchor MSC.

routed from the home MSC to the anchor MSC, and then from the anchor MSC to the visited MSC where the mobile is currently located. when a Mobile moves from the coverage area of one MSC to another, the Ongoing call is routed from the anchor MSC to the new visited MSC containing the new base station. Thus, at all times there are at most three MSC's (the home MSC, the anchor MSC, and the visited MSC) between the correspondent and the Mobile. Figure 6.32 illustrates the routing of call among the MSC's visited by a mobile user

GSM element	Comment on GSM element	Mobile IP element
Home System	Network to which the Mobile user's permanent phone number belongs.	Home network.
Gateway Mobile Switching center or simply home MSC, Home location Register (HLR)	Home MSC: point of contact to obtain routable address of mobile user HLR: database in home system containing permanent phone number, profile information, current location of mobile user, subscription information	Home agent
Visited System	Network other than home System where mobile user is currently residing.	Visited Network.
Visited mobile Services switching center, visitor Location Register (VLR)	Visited MSC: responsible for setting up calls to/ from mobile nodes in cells associated with MSC. VLR: temporary database entry in visited system, Containing subscription information for each visiting mobile user.	Foreign agent
Mobile station roaming number (MSRN) or simply roaming number	Routable address for tele phone call segment between home MSC and visited MSC, visible to neither the mobile nor the Component	care-of-address

Commonalities between mobile IP and GSM Mobility.

6.8. Wireless and Mobility: Impact on Higher Layer Protocols

Wireless networks differ significantly from their wired counterparts at both the link layer and at the network layer. But are there important differences at the transport and application layers?

* It's tempting to think that these differences will be minor, since the network layer provides the same best-effort delivery service model to upper layers in both wired and wireless networks.

* Similarly, if protocols such as TCP or UDP are used to provide transport layer should remain unchanged as well.

* Transport protocols in general, and TCP in particular, can sometimes have very different performance in wired and wireless networks, and it is here, in terms of performance that differences are manifested.

* TCP retransmits a segment that is either lost or corrupted on the path between sender and receiver. In case of mobile users, loss can result from either network congestion (router buffer overflow) or from handoff. (e.g. from delays in rerouting segments to a mobile's new point of attachment to the network)

* In Wired, If segment lost due to congestion, or due to detected bit errors, the sender's response is the same — to retransmit the segment. TCP congestion control response is also same in all cases — TCP decreases its congestion window.

In Wireless, When such bit errors occur or when handoff loss occurs, there's really no reason for the TCP sender to decrease its Congestion Window.

TCP's Congestion-Control response could be problematic in a wireless setting. Three broad classes of approaches are possible for dealing with this problem:

* Local recovery: Local recovery protocols recover from bit errors when and where (e.g. at the wireless link) they occur. e.g., the 802.11 ARQ protocol or more sophisticated approaches that use both ARQ and FEC.

* TCP sender awareness of wireless links. In the local recovery approaches, the TCP sender is unaware that its segments are traversing a wireless link. An alternative approach is for the TCP sender and Receiver to be aware of the existence of a wireless link, to distinguish between Congestive losses occuring in the wired network and Corruption/loss occuring at the Wireless link, and to invoke Congestion Control only in response to Congestive Wired-network losses.

* Split-connection approaches. In a split connection approach the end-to-end connection between the mobile user and the other end point is broken into two transport layer connections; one from the mobile host to the Wireless access point, and one from the Wireless access point to the other Communication end point (which we'll assume here is a Wired host). The end-to-end connection is thus formed by the Concatenation

of a wireless part and a Wired part. The transport layer over the Wireless segment can be a Standard TCP connection or a Specially tailored error recovery protocol on top of UDP. Selective repeat protocol over the wireless connection.

Next consider the effect of Wireless and mobility on application-layer protocols. Important Considerations is that Wireless links often have relatively low bandwidths, applications that operate over Wireless links, particularly over cellular wireless links, must treat bandwidth as a Scarce commodity.

For Ex. a Web server serving content to a Web browser executing on a 3G phone will likely not be also to provide the same rich content that it gives to a browser operating over a Wired connection.

Although wireless links do provide challenges at the application layer, the mobility they enable also makes possible a rich set of location-aware and context-aware applications.