Marudhar Kesari Jain College For Women, Vaniyambadi.
Class : II M.COM CA
Subject name : MOBILE COMPUTING
Subject code : DECP 44A

MOBILE COMPUTING:

UNIT-2

TOPICS:

=> **Overview of Mobile IP**
=> **Features of Mobile IP**
=> **Key Mechanism in Mobile IP**
=> **Route Optimization**
=> **Overview of TCP / IP**
=> **Architecture of TCP/IP**
=> **Adaptation of TCP / IP Window**
=> **Improvement in TCP Performance**

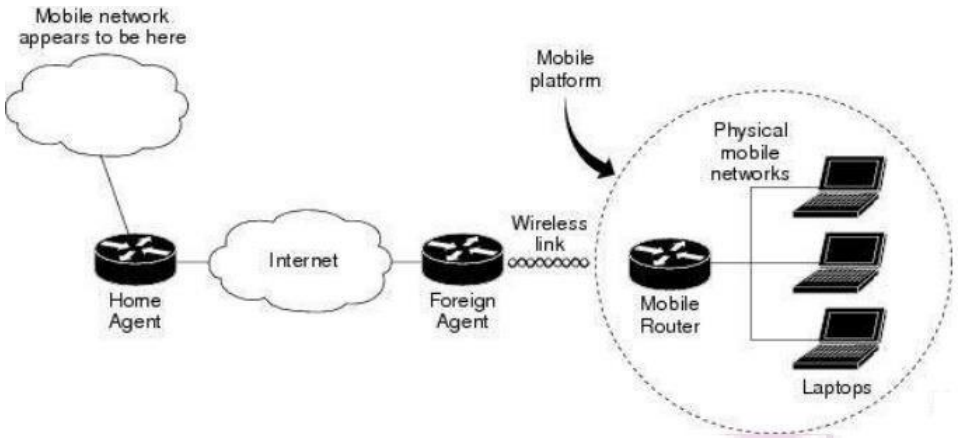
OVERVIEW OF MOBILE IP

- ✓ Mobile IP is an **open standard**, defined by the Internet Engineering Task Force (IETF) RFC 3220. By using Mobile IP, you can **keep the same IP address**, stay connected, and maintain ongoing applications while roaming between IP networks.
- ✓ Mobile IP is scalable for the **Internet because it is based on IP**—any media that can support IP can support Mobile IP.
- ✓ **The Cisco Mobile Networks feature** enables a mobile access router and its subnets to be mobile and maintain all IP connectivity, transparent to the IP hosts connecting through this mobile access router.
- ✓ Currently, this feature is a **static network implementation** that supports stub routers only. In IP networks, routing is based on stationary IP addresses. A device on a network is reachable through normal IP routing by the IP address it is assigned on the network.

- ✓ When a device roams away from its home network, it is no longer reachable by using normal IP routing. This results in the active sessions of the device being terminated.

Mobile IP enables users to keep the same IP address while travelling to a different network, ensuring that a roaming individual can continue communication without sessions or connections being dropped. **Because the mobility functions of Mobile IP are performed at the network layer rather than the physical layer, the mobile device can span different types of wireless and wire line networks while maintaining connections**.

- ✓ Remote login, remote printing, and file transfers are examples of applications where it is desirable not to interrupt communications while an individual roams across network boundaries.
- ✓ Also, certain network services, such as software licenses and access privileges, are based on IP addresses. Changing these IP addresses could compromise the network services.
- ✓ A device that can roam while appearing to a user to be at its home network is called a mobile node.
- ✓ Examples of mobile nodes include: a personal digital assistant, a laptop computer, or a data-ready cellular phone—that can change its point of attachment from one network or subnet to another.
    - This mobile node can travel from link to link and maintain communications using the same IP address.
    - There is no need for any changes to applications, because the solution is at the network layer, which provides the transparent network mobility. The Cisco Mobile Networks feature comprises three components
    - The mobile access router (MR),
    - Home agent (HA), and
    - Foreign agent (FA). Figure shows the three components (mobile access router, home agent, and foreign agent) and their relationships within the mobile network.



- The mobile access router functions similarly to the mobile node with one key difference—the mobile access router allows entire networks to roam.

- For example, an airplane with a mobile access router can fly around the world while passengers stay connected to the Internet.

- This communication is accomplished by Mobile IP aware routers tunnelling packets, which are destined to hosts on the mobile networks, to the location where the mobile access router is visiting.

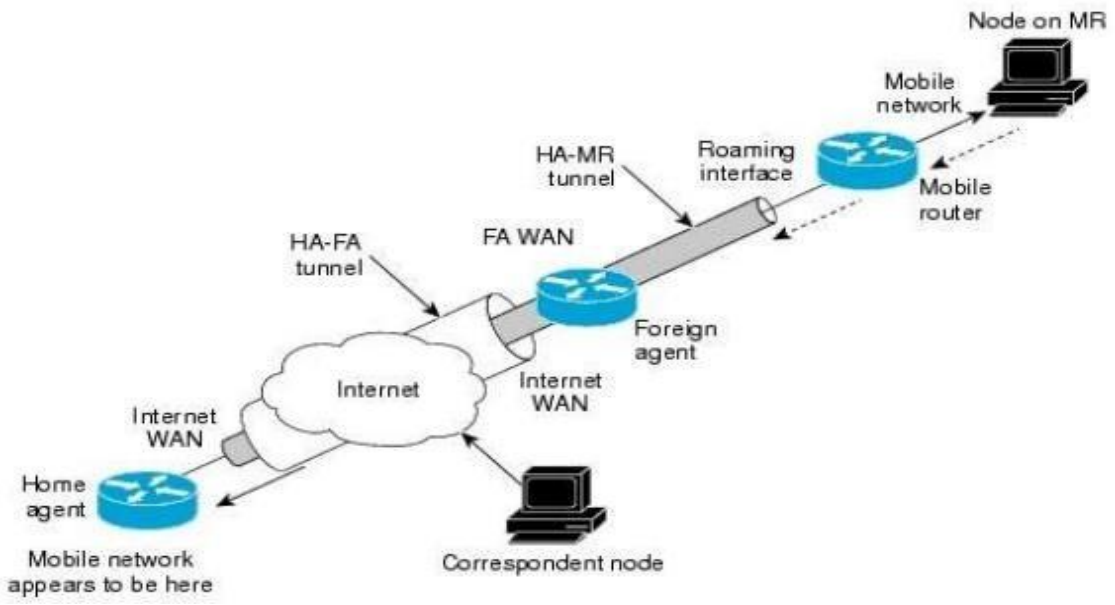
- The mobile access router then forwards the packets to the destination device. These devices can be mobile nodes without Mobile IP client software.

- The mobile access router eliminates the need for a Mobile IP client. The mobile access router ―hides‖ the IP roaming from the local IP nodes so that the local nodes appear to be directly attached to the home network.

- A home agent is a router on the home network of the mobile access router. It provides the point for the mobile networks.

- The home agent maintains an association between the home IP address of the mobile access router and its care-of address, which is the current location of the mobile access router on a foreign or visited network.
- The home agent is responsible for keeping track of where the mobile access router roams and tunnelling packets to the current location of the mobile network. The home agent also inserts the mobile networks into its routing table.

✓ A foreign agent is a router on a foreign network that assists the mobile access router in informing its home agent of its current care-of address.
✓ It functions as the point of attachment to the mobile access router, delivering packets from the home agent to the mobile access router.
✓ The foreign agent is a fixed router with a direct logical connection to the mobile access router.
✓ The mobile access router and foreign agent need not be connected directly by a wireless link.
✓ For example, if the mobile access router is roaming, the connection between the foreign agent and mobile access router occurs on interfaces that are not on the same subnet.
✓ This feature does not add any new functionality to the foreign agent component.



FEATURES OF MOBILE IP

- ✓ Mobile Internet Protocol (Mobile IP) was created in order to provide better mobile connectivity without interrupting computers that are already connected to a network.
- ✓ When mobile devices were introduced, there was no network technology in place for these devices to connect to the Internet.
- ✓ Mobile IP created a new subset of IP connectivity that worked within the already established system, keeping network engineers from having to scrap and reinvent the way Internet connection works.

Roaming Connectivity

- Mobile IP allows mobile devices to connect to the Internet when they are not at their home network. This lets laptops connect to hotspots and it lets phones connect through 3G and other Internet network sources.
- An IP address lets a network know where to send and receive information from on a network. Mobile IP uses an address that references its home network while finding a location on the new network.
- This keeps Mobile IP from knocking other computers off of a network, because each computer comes from a unique network and has a unique number.

Compatibility

- Mobile IP is compatible with most networks that offer the Internet. This include the 3G network used for mobile televisions;
- Internet hotspots found in cafes, airports and book stores; and all home network devices.
- Early attempts at Mobile IP would only work with certain routers or certain types of networks. Mobile IP today has no special requirements because the system is universal and fits within the original IP infrastructure.

Tunnelling and Reverse Tunnelling

- The method by which mobile IP receives **information from a network** is called tunnelling.
- A network cannot directly send information to a mobile IP device. In order to get this information the mobile device must create an IP address within its new IP address.
- This allows the network to send information to the IP address through the ―tunnel‖ of the two new IPs. Firewalls and routers can sometimes block tunnelling by enabling what is called ingress filtering.
- Mobile IP also can use the process of reverse tunnelling, which is a similar process that reverses the flow of information to achieve the same result as tunnelling.

Cordless

- The greatest feature of Mobile IP is that there are no cords needed to complete the network connection.
- The standard IP required that networks be connected by a phone line or Ethernet cord. With Mobile IP, the device finds the network automatically and attempts to establish a connection.
- Some mobile capable devices like laptop computers have the ability to connect using the Mobile IP or using the standard IP with an Ethernet or phone cord.

KEY MECHANISM IN MOBILE IP

The Mobile IP process has three main phases, which are discussed in the following sections.

i. **Agent Discovery** - A Mobile Node discovers its Foreign and Home Agents during agent discovery.

ii. **Registration** - The Mobile Node registers its current location with the Foreign Agent and Home Agent during registration.

iii. **Tunnelling** - A reciprocal tunnel is set up by the Home Agent to the care-of address (current location of the Mobile Node on the foreign network) to route packets to the Mobile Node as it roams.

i. Agent Discovery

During the agent discovery phase, the Home Agent and Foreign Agent advertise their services on the network by using the ICMP Router Discovery Protocol (IRDP).

The Mobile Node listens to these advertisements to determine if it is connected to its home network or foreign network.

The IRDP advertisements carry Mobile IP extensions that specify whether an agent is a Home Agent, Foreign Agent, or both; its care-of address; the types of services it will provide such as reverse tunnelling and generic routing encapsulation (GRE); and the allowed registration lifetime or roaming period for visiting Mobile Nodes.

Rather than waiting for agent advertisements, a Mobile Node can send out an agent solicitation. This solicitation forces any agents on the link to immediately send an agent advertisement.

If a Mobile Node determines that it is connected to a foreign network, it acquires a care-of address. Two types of care-of addresses exist:

- Care-of address acquired from a Foreign Agent
- Co-located care-of address

A Foreign Agent care-of address is an IP address of a Foreign Agent that has an interface on the foreign network being visited by a Mobile Node.

A Mobile Node that acquires this type of care-of address can share the address with other Mobile Nodes. A co-located care-of address is an IP address temporarily assigned to the interface of the Mobile Node itself.

A co-located care-of address represents the current position of the Mobile Node on the foreign network and can be used by only one Mobile Node at a time. When the Mobile Node hears a Foreign Agent advertisement and detects that it has moved outside of its home network, it begins registration.

ii. Registration

The Mobile Node is configured with the IP address and mobility security association (which includes the shared key) of its Home Agent. In addition, the Mobile Node is configured with either its home IP address, or another user identifier, such as a Network Access Identifier.

The Mobile Node uses this information along with the information that it learns from the Foreign Agent advertisements to form a Mobile IP registration request. It adds the registration request to its pending list and sends the registration request to its Home Agent either through the Foreign Agent or directly if it is using a co-located care-of address and is not required to register through the Foreign Agent.

If the registration request is sent through the Foreign Agent, the Foreign Agent checks the validity of the registration request, which includes checking that the requested lifetime does not exceed its limitations, the requested tunnel encapsulation is available, and that reverse tunnel is supported.

If the registration request is valid, the Foreign Agent adds the visiting Mobile Node to its pending list before relaying the request to the Home Agent. If the registration request is not valid, the Foreign Agent sends a registration reply with appropriate error code to the Mobile Node.

The Home Agent checks the validity of the registration request, which includes authentication of the Mobile Node. If the registration request is valid, the Home Agent creates a mobility binding (an association of the Mobile Node with its care-of address), a tunnel to the care-of address, and a routing entry for forwarding packets to the home address through the tunnel.

The Home Agent then sends a registration reply to the Mobile Node through the Foreign Agent (if the registration request was received via the Foreign Agent) or directly to the Mobile Node. If the registration request is not valid, the Home Agent rejects the request by sending a registration reply with an appropriate error code.

The Foreign Agent checks the validity of the registration reply, including ensuring that an associated registration request exists in its pending list. If the registration reply is valid, the Foreign Agent adds the Mobile Node to its visitor list, establishes a tunnel to the Home Agent, and creates a routing entry for forwarding packets to the home address. It then relays the registration reply to the Mobile Node.

Finally, the Mobile Node checks the validity of the registration reply, which includes ensuring an associated request is in its pending list as well as proper authentication of the Home Agent. If the registration reply is not valid, the Mobile Node discards the reply. If a valid registration reply specifies that the registration is accepted, the Mobile Node is confirmed that the mobility agents are aware of its roaming. In the co-located care-of address case, it adds a tunnel to the Home Agent. Subsequently, it sends all packets to the Foreign Agent.

The Mobile Node reregisters before its registration lifetime expires. The Home Agent and Foreign Agent update their mobility binding and visitor entry, respectively, during re-registration. In the case where the registration is denied, the

Mobile Node makes the necessary adjustments and attempts to register again. For example, if the registration is denied because of time mismatch and the Home Agent sends back its time stamp for synchronization, the Mobile Node adjusts the time stamp in future registration requests.

Thus, a successful Mobile IP registration sets up the routing mechanism for transporting packets to and from the Mobile Node as it roams.

iii. Tunnelling

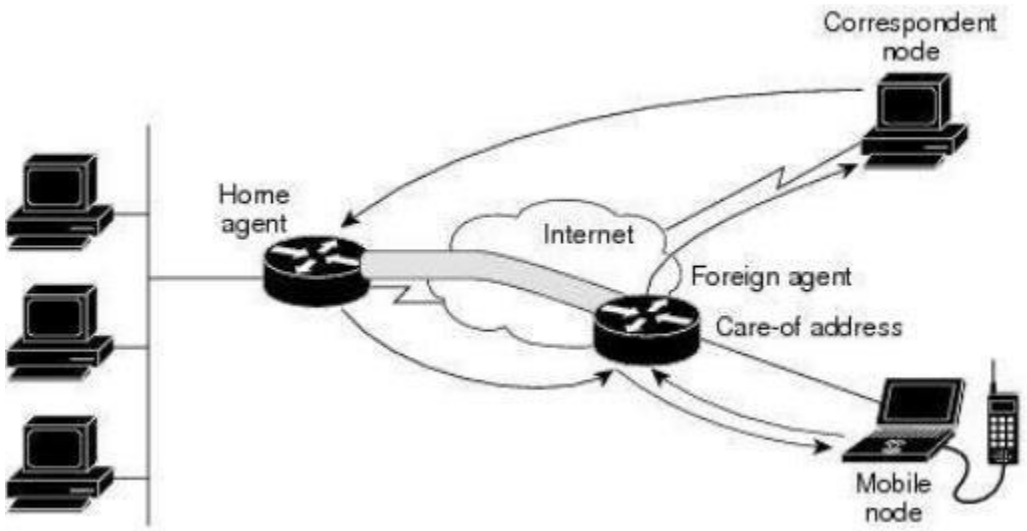
The Mobile Node sends packets using its home IP address, effectively maintaining the appearance that it is always on its home network. Even while the Mobile Node is roaming on foreign networks, its movements are transparent to correspondent nodes.

Data packets addressed to the Mobile Node are routed to its home network, where the Home Agent now intercepts and tunnels them to the care-of address toward the Mobile Node.

Tunnelling has two primary functions: encapsulation of the data packet to reach the tunnel endpoint, and decapsulation when the packet is delivered at that endpoint.

The default tunnel mode is IP Encapsulation within IP Encapsulation. Optionally, GRE and minimal encapsulation within IP may be used.Typically, the Mobile Node sends packets to the Foreign Agent, which routes them to their final destination, the Correspondent Node, as shown in Figure 2.

Packet Forwarding

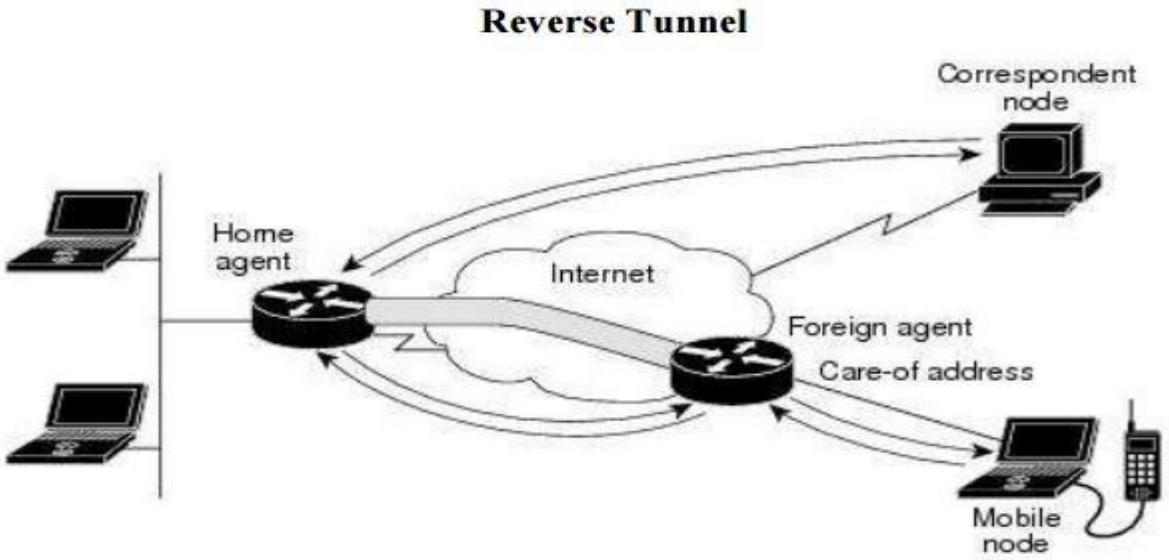


However, this data path is topologically incorrect because it does not reflect the true IP network source for the data — rather, it reflects the home network of the Mobile Node.

Because the packets show the home network as their source inside a foreign network, an access control list on routers in the network called ingress filtering drops the packets instead of forwarding them.

A feature called reverse tunnelling solves this problem by having the Foreign Agent tunnel packets back to the Home Agent when it receives them from the Mobile Node.

Reverse Tunnel



Tunnel MTU discovery is a mechanism for a tunnel encapsulator such as the Home Agent to participate in path MTU discovery to avoid any packet fragmentation in the routing path between a Correspondent Node and Mobile Node.

For packets destined to the Mobile Node, the Home Agent maintains the MTU of the tunnel to the care-of address and informs the Correspondent Node of the reduced packet size.

This improves routing efficiency by avoiding fragmentation and reassembly at the tunnel endpoints to ensure that packets reach the Mobile Node.

Security

Mobile IP uses a strong authentication scheme for security purposes. All registration messages between a Mobile Node and Home Agent are required to contain the Mobile-Home Authentication Extension (MHAE).

The integrity of the registration messages is protected by a preshared 128-bit key between a Mobile Node and Home Agent. The keyed message digest algorithm 5 (MD5) in "prefix+suffix" mode is used to compute the authenticator value in the appended MHAE, which is mandatory.

Mobile IP also supports the hash-based message authentication code (HMAC-MD5). The receiver compares the authenticator value it computes over the message with the value in the extension to verify the authenticity.

Optionally, the Mobile-Foreign Authentication Extension and Foreign-Home Authentication Extension are appended to protect message exchanges between a Mobile Node and Foreign Agent and between a Foreign Agent and Home Agent, respectively.

Replay protection uses the identification field in the registration messages as a timestamp and sequence number. The Home Agent returns its time stamp to synchronize the Mobile Node for registration.

Cisco IOS software allows the mobility keys to be stored on an authentication, authorization, and accounting (AAA) server that can be accessed using TACACS+ or RADIUS protocols. Mobile IP in Cisco IOS software also contains registration filters, enabling companies to restrict who is allowed to register.

ROUTE OPTIMIZATION

Mobile IPv4 route optimization

Mobile IPv4 route optimization is a proposed extension to the Mobile IPv4 protocol. It provides enhancements to the routing of data grams between the mobile node and to the correspondent node.

The enhancements provide means for a correspondent node to tunnel data grams directly to the mobile node or to its foreign agent care-of address.

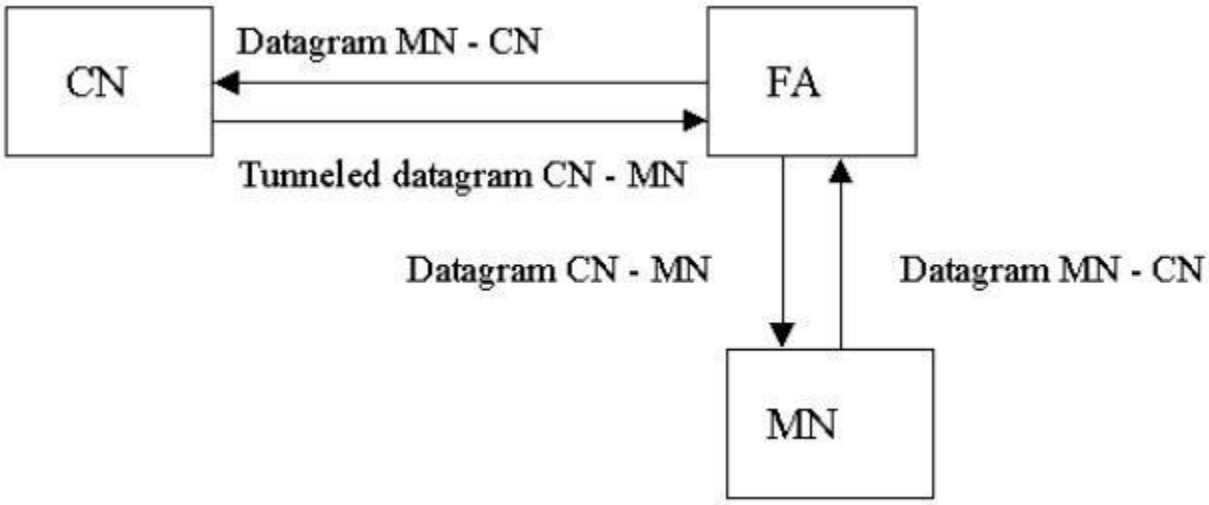
Route optimization messages and data structures

The route optimization extension adds a conceptual data structure, the binding cache, to the correspondent node and to the foreign agent.

The binding cache contains bindings for mobile nodes' home addresses and their current care-of addresses. With the binding the correspondent node can tunnel data grams directly to the mobile node's care-of address.

Every time the home agent receives a datagram that is destined to a mobile node currently away from home, it sends a binding update to the correspondent node to update the information in the correspondent node's binding cache. After this the correspondent node can directly tunnel packets to the mobile node. Thus direct bi-directional communication is achieved with route optimization.

Direct routing with route optimization and foreign agent care-of address.



Route optimization adds four new UDP-messages to the Mobile IPv4 protocol:

Binding update informs the correspondent node or foreign agent of the mobile node's new location. It is sent by the home agent or in the case of previous foreign agent notification, by the new foreign agent, as shown in Figure 4. The binding update contains the care-of address and the home address of the mobile node and also the lifetime of the binding.

It also must contain a mobile IP authentication extension. An identification number may also be present to provide a way of matching updates with acknowledgements and to protect against replay attacks.

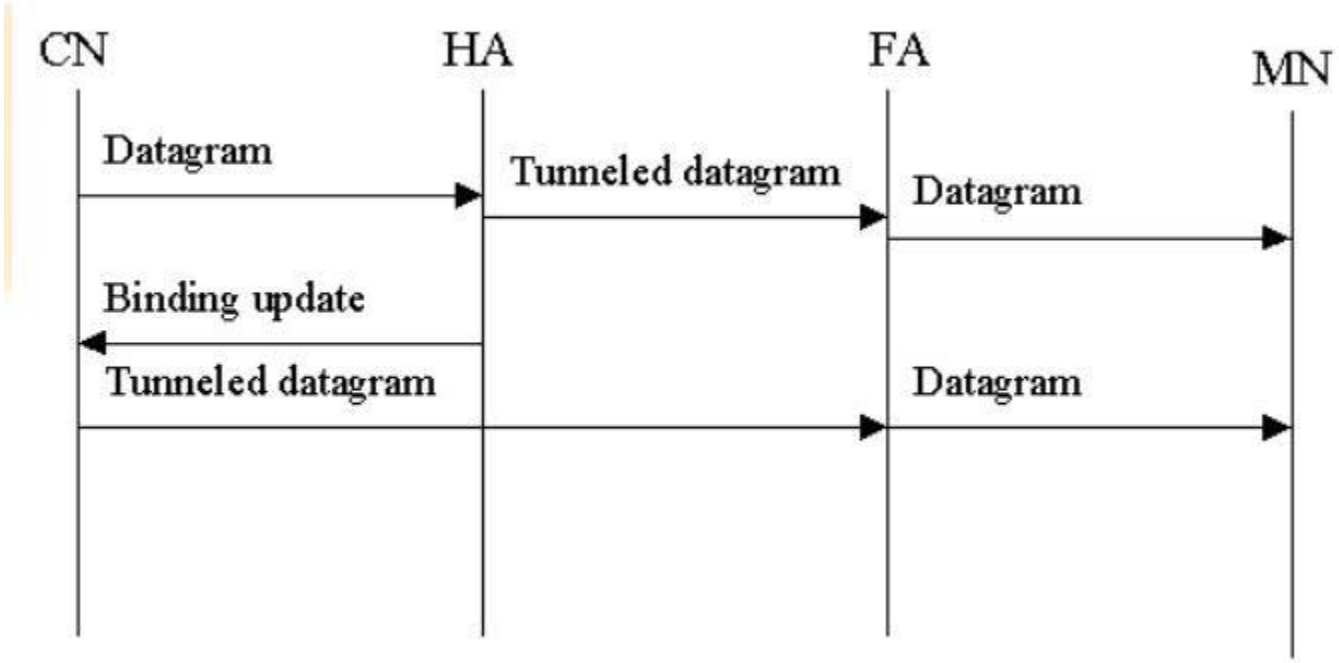
Binding acknowledgement is sent by the correspondent node or the foreign agent in response to the binding update. It contains the mobile node's home address and a status code. It also contains an identification number, if there was one in the corresponding binding update.

Binding request is sent by the correspondent node to the home agent to request a binding update. It contains the home address of the queried mobile node and possibly an identification number.

Binding warning is sent by the previous foreign agent in response to receiving a tunnelled datagram for a mobile node for which it has a binding and for which it is not acting as the current foreign agent.

The binding warning is sent to the home agent. It contains the home address of the mobile node and the address of the correspondent node that does not have up to date information of the mobile node's current care-of address. With this information the home agent can send a binding update to the correspondent node.

Binding update to correspondent node



The effect on static routes

As the correspondent node learns the care-of address of the mobile node from the binding update, it can tunnel data grams directly to the mobile node's care-of address .

Thus only the first data grams are routed via the home agent. This reduces the network load and also reduces the delays caused by routing. Thus the optimization is valuable to mobile nodes that visit networks located far from their home agent.

However, the overhead caused by tunnelling is not decreased. The correspondent node's use of minimal encapsulation is a partial remedy, if both the encapsulator and the decapsulator support it. Ingress filtering may also prevent the mobile node from sending data grams directly to the correspondent node.

The use of direct reverse tunnelling from the care-of address to the correspondent node's address is a possible solution to ingress filtering. However, it is not possible with foreign agent care-of addresses, since the current reverse tunnelling standard requires the foreign agent to tunnel all packets to the home agent of the mobile node.

Smooth handoffs with route optimization

In the static case the protocol is fairly simple, but handoffs somewhat complicate the situation. When the correspondent node has an out of date entry for the mobile node's care-of address it tries to send the tunnelled datagram to the mobile node's previous location and the datagram is lost.

To solve this problem the protocol includes the previous foreign agent notification mechanism, which adds a binding cache to the foreign agent.

When a mobile node moves to a new sub network it sends a registration request to the new foreign agent.

The registration request may contain a previous foreign agent notification extension. Upon receiving such a request the foreign agent builds a binding update and sends it to the previous foreign agent.

previous foreign agent can then, after authenticating the update, create a binding for the mobile node. With this binding it can re-tunnel data grams to the mobile node's new care-of address.

The re-tunnelling requires foreign agent care-of addresses in order for the agents to act as tunnel endpoints.

The previous foreign agent notification mechanism provides temporary localization of the handoffs. It does not reduce the signalling load between the home agent and the mobile node, but reduces the number of data grams lost due to correspondent nodes with out-of date bindings.

Security considerations

Since the correspondent nodes and foreign agents have binding caches, which change the routing of data grams destined to mobile nodes, the binding updates must be authenticated.

The authentication is performed in a similar manner as in base Mobile IPv4. All binding updates contain a route optimization or smooth handoff authentication extension. This extension contains a hash, which is calculated from the datagram and the shared secret.

The correspondent node and the mobile node's home agent need a security association. This association is used for the authentication of the binding updates.

Since the mobile node sends a binding update directly to its previous foreign agent, they also need a security association. If the security associations are not preconfigured they can be established via a key management protocol such as ISAKMP or SKIP.

General deployment requirements

In order to make use of the binding updates the correspondent nodes must be able to process and authenticate them and be able to encapsulate data grams.

To establish this, the network stacks of the operating systems require changes. Since correspondent nodes need to establish a security association with the home agent and foreign agents need to establish one with the mobile node, a widely deployed key management system is obviously needed. Otherwise only nodes with statically configured security associations can benefit from the binding updates.

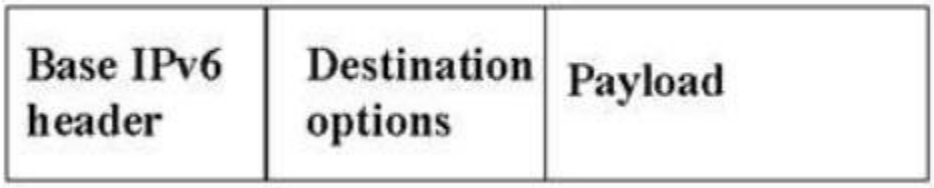
Mobile IPv6 and route optimization

Main characteristics of Mobile IPv6

Whereas Mobile IP was added on top of the IPv4 protocol, in IPv6 mobility support is built into the IP-layer. In mobile IPv6 route optimization is an essential part of the protocol. Mobile nodes have a binding update list, which contains the bindings other nodes have for it.

Correspondent nodes and home agents have a binding cache, which contains the home and care-of addresses of mobile nodes they have been recently communicating with. All signalling is performed via destination options that are appended to the base IPv6 header. Thus all signalling traffic can be piggybacked on data grams with a data payload, as in Figure 5.

Destination option.



The destination options are:

- Binding update option, which is sent by the mobile node to its home agent and correspondent nodes to inform them of a change of location.
- Binding acknowledgement option, which is sent in response to the binding update.
- Binding request option, with which a node can request a new binding update from the mobile node, when the binding is about to expire.

• Home address option, which the mobile node appends to all data grams it sends while away from its home network.

The home address option is used to avoid the negative effects of ingress filtering by using the topologically correct care-of address as the source address and including the home address in the option.

The receiving node will then copy the home address to the source address before passing the packet to any transport level protocol.

All care-of addresses in Mobile IPv6 are co-located; thus foreign agents are not a part of the protocol.

Since all nodes are only required to understand the home address option, triangle routing will occur also with mobile IPv6. However, if the correspondent node implements the draft fully, only the first data grams it sends will be routed via the home agent.

The mobile node always sends a binding update to the original sender of a tunnelled datagram. With this binding the correspondent node can send data grams directly to the mobile node using a routing header.

A datagram with a routing header contains the care-of address as the destination address and the home address in the routing extension header as the final destination.

Thus the datagram will be normally routed to the care-of address. When the mobile node receives a datagram with a routing header it swaps the final destination with the destination address field. The home address option and the routing header make the mobility transparent with direct routing.

The Effect on Routing

By using direct routes in both directions the consumption of network resources is minimized. The 40-byte IPv6 headers consume extra bandwidth when compared to 20 byte IPv4 headers.

However the use of routing header and home address option removes the need for constant tunnelling, thus decreasing the bandwidth consumption. Although they both add overhead to packets they still are considerably smaller than IPv6 headers, which would be used in tunnelling.

The destination options used for signalling can be piggybacked [4] which decreases the signalling overhead considerably, since the options are relatively small when compared to UDP packets.

The effect on handoffs

The IPv6 mobility support provides the previous router notification mechanism, with which the amount of lost of packets in handoffs can be reduced. In IPv6 the mobile node sends a binding update directly to the previous router, which

consumes more bandwidth but is faster than the mechanism used with Mobile IPv4 route optimization.

Problems solved

Mobile IPv6 provides improvements on routing and signalling efficiency. As the signalling can be mostly piggybacked on data packets there will be considerably less signalling overhead between the mobile node and the correspondent nodes than in mobile IPv4 route optimization between the home agent and the correspondent nodes.

The minimum requirements for the correspondent node provide at least triangle routing even in the worst case, since care-of address can be used as the source address.

Hosts that are likely to communicate with mobile nodes will probably implement the binding cache and communicate directly with the mobile node. In both cases the routing saves network capacity and decreases delays, when compared to reverse bi-directional tunnelling between the mobile node and correspondent node.

The key management problem is not solved Mobile IPv6 does not solve the key management problem, but the integration of IPSec into IPv6 is likely to result in support for key management protocols in most operating systems implementing IPv6.

OVERVIEW OF TCP / IP

TCP/IP (**Transmission Control Protocol/Internet Protocol)** is the basic communication language or protocol of the Internet. It can also be used as a communications protocol in a private network (either an intranet or an extranet). When you are set up with direct access to the Internet, your computer is provided with a copy of the TCP/IP program just as every other computer that you may send messages to or get information from also has a copy of TCP/IP.

TCP/IP is a two-layer program.

The higher layer, Transmission Control Protocol, manages the assembling of a message or file into smaller packets that are transmitted over the Internet and received by a TCP layer that reassembles the packets into the original message. The lower layer, Internet Protocol, handles the address part of each packet so that it gets to the right destination.

Each gateway computer on the network checks this address to see where to forward the message. Even though some packets from the same message are routed differently than others, they'll be reassembled at the destination.

TCP/IP uses the client/server model of communication in which a computer user (a client) requests and is provided a service (such as sending a Web page) by another computer (a server) in the network.

TCP/IP communication is primarily point-to-point, meaning each communication is from one point (or host computer) in the network to another point or host computer.

TCP/IP and the higher-level applications that use it are collectively said to be "stateless" because each client request is considered a new request unrelated to any previous one (unlike ordinary phone conversations that require a dedicated connection for the call duration).

Many Internet users are familiar with the even higher layer application protocols that use TCP/IP to get to the Internet. These include the World Wide Web's Hypertext Transfer Protocol (HTTP), the File Transfer Protocol (FTP), Telnet (Telnet) which lets you logon to remote computers, and the Simple Mail Transfer Protocol (SMTP). These and other protocols are often packaged together with TCP/IP as a "suite."

Personal computer users with an analog phone modem connection to the Internet usually get to the Internet through the Serial Line Internet Protocol (SLIP) or the Point-to- Point Protocol (PPP). These protocols encapsulate the IP packets so that they can be sent over the dial-up phone connection to an access provider's modem.

Protocols related to TCP/IP include the User Datagram Protocol (UDP), which is used instead of TCP for special purposes. Other protocols are used by network host computers for exchanging router information.

These include the Internet Control Message Protocol (ICMP), the Interior Gateway Protocol (IGP), the Exterior Gateway Protocol (EGP), and the Border Gateway Protocol (BGP).

TCP/IP Protocols for the Web

Web browsers and servers use TCP/IP protocols to connect to the Internet. Common TCP/IP protocols are:

i.HTTP - Hyper Text Transfer Protocol

HTTP takes care of the communication between a web server and a web browser. HTTP is used for sending requests from a web client (a browser) to a web server, returning web content (web pages) from the server back to the client.

ii. HTTPS - Secure HTTP

HTTPS takes care of secure communication between a web server and a web browser. HTTPS typically handles credit card transactions and other sensitive data.

iii. FTP - File Transfer Protocol

FTP takes care of transmission of files between computers.

TCP/IP Protocols for Email

E-mail programs use TCP/IP for sending and receiving e-mails. The TCP/IP protocols for email are:

i. SMTP - Simple Mail Transfer Protocol

SMTP takes care of sending emails. Often emails are sent to an email server (SMTP server), then to other servers, and finally to its destination. SMTP can only transmit pure text. It cannot transmit binary data like pictures, sounds or movies.

ii. MIME - Multi-purpose Internet Mail Extensions

The MIME protocol lets SMTP transmit multimedia files including voice, audio, and binary data across TCP/IP networks. The MIME protocol converts binary data to pure text, before it is sent.

iii. POP - Post Office Protocol

The POP protocol is used by email programs to retrieve emails from an email server. If your email program uses POP, all your emails are downloaded to

your email program (also called email client), each time it connects to your email server.

iv. IMAP - Internet Message Access Protocol

The IMAP protocol works much like the POP protocol. The main difference is that the IMAP protocol will not automatically download all your emails each time your email program connects to your email server.

The IMAP protocol allows you to look through your email messages at the email server before you download them. With IMAP you can choose to download your messages or just delete them. This way IMAP is perfect if you need to connect to your email server from different locations, but only want to download your messages when you are back in your office.

Other TCP/IP Protocols

ARP - Address Resolution Protocol

ARP is used by IP to find the hardware address of a computer network card based on the IP address.

BOOTP - Boot Protocol

BOOTP is used for booting (starting) computers from the network.

DHCP - Dynamic Host Configuration Protocol

DHCP is used for allocation of dynamic IP addresses to computers in a network.

ICMP - Internet Control Message Protocol

ICMP takes care of error-handling in the network.

LDAP - Lightweight Directory Access Protocol

LDAP is used for collecting information about users and e-mail addresses from the internet.

NTP - Network Time Protocol

NTP is used to synchronize the time (the clock) between computers.

PPTP - Point to Point Tunnelling Protocol

PPTP is used for setting up a connection (tunnel) between private networks.

RARP - Reverse Address Resolution Protocol

RARP is used by IP to find the IP address based on the hardware address of a computer network card.

SNMP - Simple Network Management Protocol

SNMP is used for administration of computer networks.

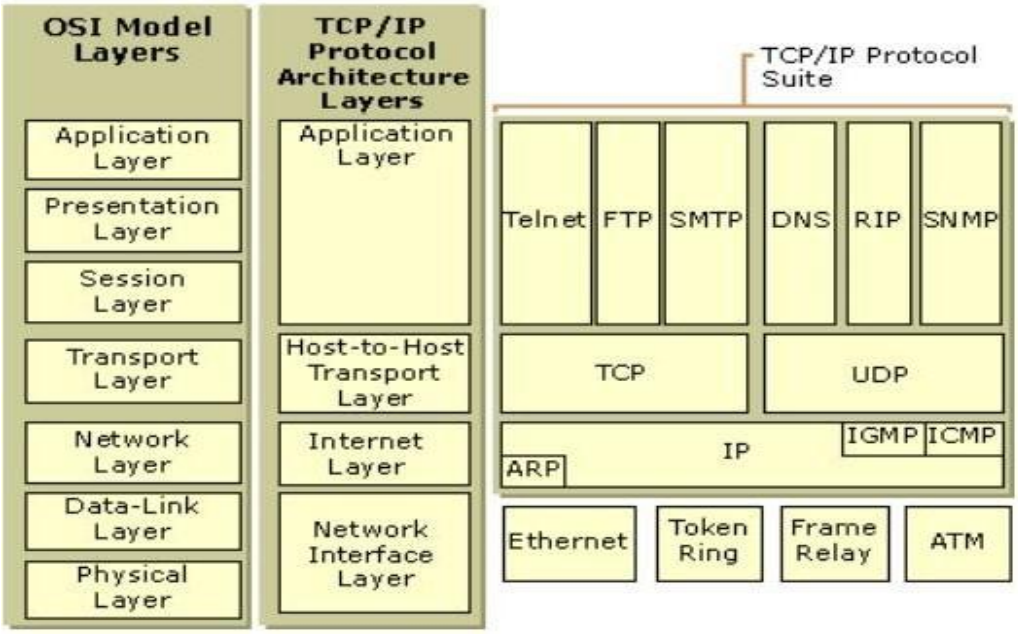
SSL - Secure Sockets Layer

The SSL protocol is used to encrypt data for secure data transmission.

TLS - Transport Layer Security

The TLS protocol is a newer and more secure version of SSL.

ARCHITECTURE OF TCP / IP



When communication among computers from different vendors is desired, the software development effort can be a nightmare. Different vendors use different data formats and data exchange protocols. Even within one

vendor's product line, different model computers may communicate in unique ways.

As the use of computer communications and computer networking proliferates, a one-at-a-time, special-purpose approach to communications software development is too costly to be acceptable. The only alternative is for computer vendors to adopt and implement a common set of conventions. For this to happen, standards are needed. Such standards would have two benefits:

i. Vendors feel encouraged to implement the standards because of an expectation that, because of wide usage of the standards, their products would be less marketable without them.

ii. Customers are in a position to require that the standards be implemented by any vendor wishing to propose equipment to them.

However, no single standard will suffice. Any distributed application, such as electronic mail or client/server interaction, requires a complex set of communications functions for proper operation. Many of these functions, such as reliability mechanisms, are common across many or even all applications.

Thus, the communications task is best viewed as consisting of a modular architecture, in which the various elements of the architecture perform the various required functions. Hence, before standards can be developed, there should be a structure, or protocol architecture, that defines the communications tasks.

Two protocol architectures have served as the basis for the development of interoperable communications standards: the *TCP/IP* protocol suite and the Open Systems Interconnection (*OSI*) reference model. TCP/IP is the most widely used interoperable architecture, and has won the "protocol wars." Although some useful standards have been developed in the context of OSI, TCP/IP is now the universal interoperable protocol architecture. No product should be considered as part of a business information system that does not support TCP/IP.

TCP/IP is a result of protocol research and development conducted on the experimental packet-switched network, ARPANET, funded by the Defence Advanced Research Projects Agency (DARPA), and is generally referred to as the *TCP/IP* protocol suite.

This protocol suite consists of a large collection of protocols that have been issued as Internet standards by the Internet Activities Board (IAB).

TCP/IP Layers

There is no official TCP/IP protocol model, as there is in the case of OSI. However, based on the protocol standards that have been developed, we can organize the communication task for TCP/IP into five relatively independent layers:

- Application layer
- Host-to-host, or transport layer
- Internet layer
- Network access layer
- Physical layer

The physical layer covers the physical interface between a data transmission device (such as a workstation or computer) and a transmission medium or network. This layer is concerned with specifying the characteristics of the transmission medium, the nature of the signals, the data rate, and related matters.

The network access layer is concerned with the exchange of data between an end system and the network to which it's attached. The sending computer must provide the network with the address of the destination computer, so that the network can route the data to the appropriate destination. The sending computer may need to invoke certain services, such as priority, that might be provided by the network.

The specific software used at this layer depends on the type of network to be used; different standards have been developed for circuit-switching, packet-switching (for example, frame relay), local area networks (such as Ethernet), and others. Thus, it makes sense to separate those functions having to do with network access into a separate layer.

By doing this, the remainder of the communications software, above the network access layer, need not be concerned about the specifics of the network to be used. The same higher-layer software should function properly regardless of the particular network to which the computer is attached.

The network access layer is concerned with access to and routing data across a network for two end systems attached to the same network. In those cases where two devices are attached to different networks, procedures are needed to allow data to traverse multiple interconnected networks.

This is the function of the Internet layer. The Internet protocol (IP) is used at this layer to provide the routing function across multiple networks. This protocol is implement not only in the end systems but also in routers. A router is a processor that connects two networks; its primary function is to relay data from one network to the other on its route from the source to the destination end system.

Regardless of the nature of the applications that are exchanging data, there is usually a requirement that data be exchanged reliably. That is, we want to be assured that all of the data arrives at the destination application, in the order in which it was sent.

The mechanisms for providing reliability are essentially independent of the nature of the applications. Thus, it makes sense to collect those mechanisms in a common layer shared by all applications; this is referred to as the host-to-host or transport layer. The transmission control protocol (TCP) is the most commonly used protocol to provide this functionality.

Finally, the application layer contains the logic needed to support the various user applications. For each type of application, such as file transfer, a separate module is needed that's peculiar to that application.

The Application Layer

The application layer defines how certain services operate and how they can be used. Examples are the FTP service for transferring files, HTTP for serving Web pages and SMTP for e-mail.

These services are defined in a rather abstract manner. Two parties, called the client and the server, set up a connection over which they exchange messages in accordance with a specific protocol.

The client starts the protocol by requesting the service. Often the next step is for the server to authenticate the client, for example by asking for a password or by executing a public-key based protocol.

Taking e-mail as an example, the protocol in question is called the Simple Mail Transfer Protocol (SMTP). The client and the server set up an SMTP connection over which they exchange identifying information. The client then tells who the message is from and who the intended recipient is.

The server then indicates whether it accepts or refuses the message (for example if it's spam or the intended recipient is unknown). If the message is accepted, the client sends the actual content of the message and the server stores it in the right mailbox.

The Transport Layer

On the Internet, the transport layer is realized by two protocols. The first is the Transmission Control Protocol (TCP) and the second is the User Datagram Protocol (UDP). Both break up a message that an application wants to send into packets and attempt to deliver those packets to the intended recipient. At the recipient's side, both take the payload from the received packets and pass those to the application layer.

The main difference between TCP and UDP is that TCP is reliable and UDP is not. TCP will collect incoming packets, put them in the right order and thereby reassemble the original message.

If necessary, TCP requests retransmission of lost or damaged packets. UDP merely takes each incoming packet and delivers the payload (the original message) to the application layer. Any errors or out-of-order data should be taken care of by the application.

UDP is much faster than TCP, and so is mainly used for applications like audio and video streaming, where the occasional error is less important than getting all the data there at the right time.

More generally, UDP is designed for applications that do not require the packets to be in any specific order. Because of this, UDP is sometimes called a "connection-less" protocol.

Taking the example of e-mail again, the e-mail client and server communicate over a reliable TCP connection. The server listens on a certain port (port 25) until a connection request arrives from the client. The server acknowledges the request, and a TCP connection is established. Using this connection the client and server can exchange data.

The content of this data is not really relevant at this level: that's the responsibility of the application layer. The e-mail message and all the other information exchanged at that SMTP application layer are merely payload, data that needs to be transported. Hence the name transport layer.

The Network Layer

The network layer is responsible for transmitting and routing data packets over the network. The Internet uses the Internet Protocol or IP as its network layer. Each node on the network has an address, which of course is called the IP address. Data is sent as IP packets.

A transport layer connection is made up up of a large number of IP packets exchanged by the client and server.

The Internet Protocol (IP) is very simple: a packet has a source, a destination and a payload, and it's passed from one node in the network to another until it gets to the destination.

The IP does not notice that a packet gets lost. It just never gets to the destination. If a particular node cannot pass the packet to the next node along the normal route, it will do its best to find an alternative path. That's why IP is sometimes called a "best-effort" protocol.

When the client sends its TCP connection request, the network layer puts the request in a number of packets and transmits each of them to the server. Each packet can take a different route, and some of the packets may get lost along the way.

If they all make it, the transport layer at the server is able to reconstruct the request, and it will prepare a response confirming that a TCP connection has been set up. This response is sent back again in a number of IP packets that will hopefully make it to the client.

The Link Layer

The Internet Protocol basically assumes all computers are part of one very large "web" of nodes that can all pass packets to other nodes. There's always a route from one node to another, even if sometimes a very large number of intermediate nodes get involved. The link layer is what makes this assumption true.

The link layer provides a network connection between hosts on a particular local network, as well as interconnection between such local networks. The e-mail client runs on a personal computer in someone's home network, which is set up using the Ethernet protocol.

The link layer now is that Ethernet network. The IP packets that this computer transmits, are added as payload to Ethernet packets (called "frames") that are transmitted over the local network to the ADSL modem that connects the local network to the provider.

A different kind of link layer protocol is used to transmit the payload taken from the Ethernet frames from the ADSL modem to the provider. At the provider this payload is again passed forward using yet another link level protocol. The "web of nodes" that the Internet Protocol relies on thus actually is made up of a large number of local networks, each with their own link layer protocol, that each forward the IP packet by putting it into their own kind of message that is then sent over the local network.

The Physical Layer

The lowest layer is the physical layer, which defines how the cables, network cards, wireless transmitters and other hardware connect computers to networks and networks to the rest of the Internet.

Examples of physical layer networks are Ethernet, WiFi, Token Ring and Fiber Data Distributed Interface (FDDI). Note that many of these technologies also have their own link layer protocol. Often link and physical layer are closely related.

The physical layer provides the means to transfer the actual bits from one computer to another. In an Ethernet network (a link layer protocol), a computer is connected by plugging a network cable into its Ethernet card, and then plugging the other end of that cable into a router or switch.

The physical layer specifies how bits of data are sent over that cable: how do the electrical currents or the pulses the card sends get turned back into the data for the higher level layers. For wireless networks, this works exactly the same, except of course there is no cable.

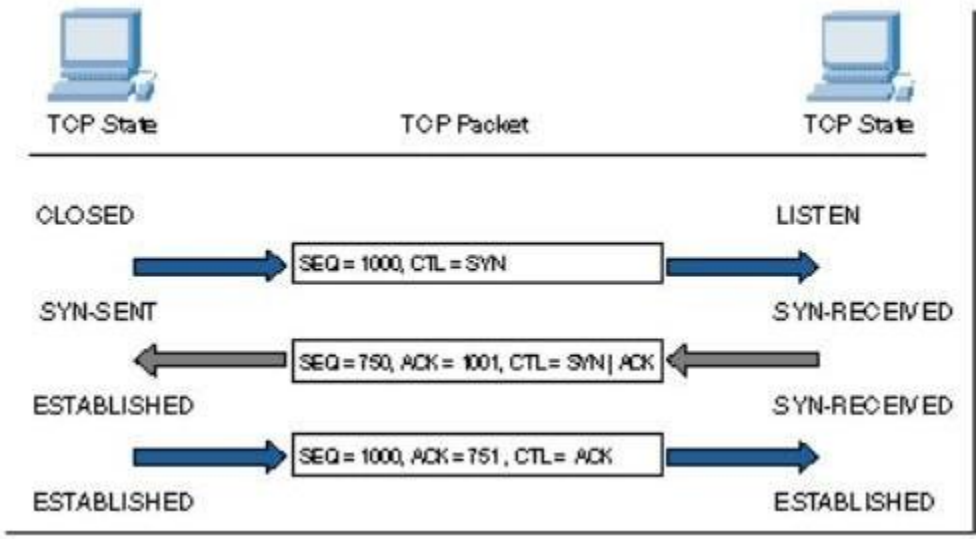
ADAPTATION OF TCP WINDOW

The first phase of a TCP session is establishment of the connection. This requires a three-way handshake, ensuring that both sides of the connection have an unambiguous understanding of the sequence number space of the remote side for this session. The operation of the connection is as follows:

- The local system sends the remote end an initial sequence number to the remote port, using a SYN packet.

- The remote system responds with an ACK of the initial sequence number and the initial sequence number of the remote end in a response SYN packet.

- The local end responds with an ACK of this remote sequence number.
- The performance implication of this protocol exchange is that it takes one and a half round-trip times (RTTs) for the two systems to synchronize state before any data can be sent.



After the connection has been established, the TCP protocol manages the reliable exchange of data between the two systems. The algorithms that determine the various retransmission timers have been redefined numerous times.

TCP is a sliding-window protocol, and the general principle of flow control is based on the management of the advertised window size and the management of retransmission timeouts, attempting to optimize protocol performance within the observed delay and loss parameters of the connection.

Tuning a TCP protocol stack for optimal performance over a very low-delay, high-bandwidth LAN requires different settings to obtain optimal performance over a dialup Internet connection, which in turn is different for the requirements of a high-speed wide-area network.

Although TCP attempts to discover the delay bandwidth product of the connection, and attempts to automatically optimize its flow rates within the estimated parameters of the network path, some estimates will not be

accurate, and the corresponding efforts by TCP to optimize behavior may not be completely successful.

Another critical aspect is that TCP is an adaptive flow-control protocol. TCP uses a basic flow-control algorithm of increasing the data-flow rate until the network signals that some form of saturation level has been reached (normally indicated by data loss). When the sender receives an indication of data loss, the TCP flow rate is reduced; when reliable transmission is reestablished, the flow rate slowly increases again.

For example, a single TCP flow across an otherwise idle network attempts to fill the network path with data, optimizing the flow rate within the available network capacity. If a second TCP flow opens up across the same path, the two flow-control algorithms will interact so that both flows will stabilize to use approximately half of the available capacity per flow.

The objective of the TCP algorithms is to adapt so that the network is fully used whenever one or more data flows are present. In design, tension always exists between the efficiency of network use and the enforcement of predictable session performance. With TCP, you give up predictable throughput but gain a highly utilized, efficient network.

IMPROVEMENT IN TCP PERFORMANCE

The protocols to improve the performance of TCP are:

Link-layer protocols

There have been several proposals for reliable link-layer protocols. The two main classes of techniques employed by these protocols are: error correction (using techniques such as forward error correction (FEC)), and retransmission of lost packets in response to automatic repeat request (ARQ) messages.

The link-layer protocols for the digital cellular systems in the U.S. — both CDMA and TDMA — primarily use ARQ techniques. While the TDMA protocol guarantees reliable, in-order delivery of link-layer frames, the CDMA protocol only makes a limited attempt and leaves it to the (reliable) transport layer to recover from errors in the worst case.

The AIRMAIL protocol employs a combination of FEC and ARQ techniques for loss recovery. The main advantage of employing a link-layer protocol for loss recovery is that it fits naturally into the layered structure of network protocols.

The link-layer protocol operates independently of higher-layer protocols (which makes it applicable to a wide range of scenarios), and consequently, does not maintain any per-connection state. The main concern about link-layer protocols is the possibility of adverse effect on certain transport-layer protocols such as TCP.

Indirect-TCP (I-TCP) protocol

This was one of the early protocols to use the split-connection approach. It involves splitting each TCP connection between a sender and receiver into two separate connections at the base station —

one TCP connection between the sender and the base station, and the other between the base station and the receiver. In our classification of protocols, ITCP is a split-connection solution that uses regular TCP for its connection over wireless

link. I-TCP, like other split-connection proposals, attempts to separate loss recovery over the wireless link from that across the wireline network, thereby shielding the original TCP sender from the wireless link.

However, as experiments indicate, the choice of TCP over the wireless link results in several performance problems. Since TCP is not well-tuned for the lossy link, the TCP sender of the wireless connection often times out, causing the original sender to stall.

In addition, every packet incurs the overhead of going through TCP protocol processing twice at the base station (as compared to zero times for a non-split-connection approach), although extra copies are avoided by an efficient kernel implementation.

Another disadvantage of this approach is that the end-to end semantics of TCP acknowledgments is violated, since acknowledgments to packets can now reach the source even before the packets actually reach the mobile host. Also, since this protocol maintains a significant amount of state at the base station per TCP connection, handoff procedures tend to be complicated and slow.

The Snoop Protocol

The snoop protocol introduces a module, called the snoop agent, at the base station. The agent monitors every packet that passes through the TCP connection in both directions and maintains a cache of TCP segments sent across the link that have not yet been acknowledged by the receiver.

A packet loss is detected by the arrival of a small number of duplicate acknowledgments from the receiver or by a local timeout.

The snoop agent retransmits the lost packet if it has it cached and suppresses the duplicate acknowledgments. In classification of protocols, the snoop protocol is a link-layer protocol that takes advantage of the knowledge of the higher-layer transport protocol (TCP).

The main advantage of this approach is that it suppresses duplicate acknowledgments for TCP segments lost and retransmitted locally, thereby avoiding unnecessary fast retransmissions and congestion control invocations by the sender.

The per-connection state maintained by the snoop agent at the base station is soft, and is not essential for correctness. Like other link-layer solutions, the snoop

approach could also suffer from not being able to completely shield the sender from wireless losses.

Selective Acknowledgments

Since standard TCP uses a cumulative acknowledgment scheme, it often does not provide the sender with sufficient information to recover quickly from multiple packet losses within a single transmission window.

Several studies have shown that TCP enhanced with selective acknowledgments performs better than standard TCP in such situations. SACKs were added as an option to TCP by RFC 1072. However, disagreements over the use of SACKs prevented the specification from being adopted, and the SACK option was removed from later TCP RFCs. Recently, there has been renewed interest in adding SACKs to TCP.

Two of the more interesting proposals are the TCP SACKs Internet Draft and the SMART scheme. The Internet Draft proposes that each acknowledgment contain information about up to three non-contiguous blocks of data that have been received successfully.

Each block of data is described by its starting and ending sequence number. Due to the limited number of blocks it is best to inform the sender about the most recent blocks received.

An alternate proposal, SMART, uses acknowledgments that contain the cumulative acknowledgment and the sequence number of the packet that caused the receiver to generate the acknowledgment (this information is a subset of the three-blocks scheme proposed in the Internet Draft).

The sender uses these SACKs to create a bitmask of packets that have been successfully received.

This scheme trades off some resilience to reordering and lost acknowledgments in exchange for a reduction in overhead to generate and transmit acknowledgments.