                            A TCP/IP Tutorial

Status of this Memo

   This RFC is a tutorial on the TCP/IP protocol suite, focusing
   particularly on the steps in forwarding an IP datagram from source
   host to destination host through a router. It does not specify an
   Internet standard. Distribution of this memo is unlimited.

Table of Contents

1. Introduction

   This tutorial contains only one view of the salient points of TCP/IP,
   and therefore it is the "bare bones" of TCP/IP technology. It omits
   the history of development and funding, the business case for its
   use, and its future as compared to ISO OSI. Indeed, a great deal of
   technical information is also omitted. What remains is a minimum of
   information that must be understood by the professional working in a
   TCP/IP environment. These professionals include the systems
   administrator, the systems programmer, and the network manager.

   This tutorial uses examples from the UNIX TCP/IP environment, however
   the main points apply across all implementations of TCP/IP.

   Note that the purpose of this memo is explanation, not definition.
   If any question arises about the correct specification of a protocol,
   please refer to the actual standards defining RFC.

The next section is an overview of TCP/IP, followed by detailed
descriptions of individual components.

2. TCP/IP Overview

The generic term "TCP/IP" usually means anything and everything
related to the specific protocols of TCP and IP. It can include
other protocols, applications, and even the network medium. A sample
of these protocols are: UDP, ARP, and ICMP. A sample of these
applications are: TELNET, FTP, and rcp. A more accurate term is
"internet technology". A network that uses internet technology is
called an "internet".

2.1 Basic Structure

To understand this technology you must first understand the following
logical structure:

```
                  ---------------------------
                 |    network applications   |
                 |                           |
                 |...  \ | /  ..  \ | /  ...|
                 |     -----       -----     |
                 |     |TCP|       |UDP|      |
                 |     -----       -----     |
                 |        \        /         |
                 |       --------           |
                 |      |  IP   |           |
                 |  -----   -*------         |
                 | |ARP|     |              |
                 |  -----    |              |
                 |     \     |              |
                 |    ------              |
                 |    |ENET|              |
                 |    ---@--              |
                  ----------|----------------
                           |
        -----------------------o---------
            Ethernet Cable
```
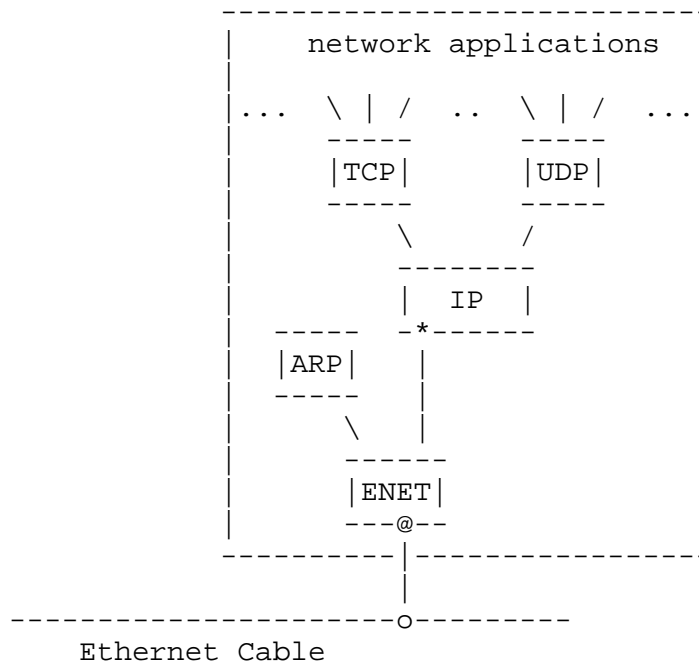
Figure 1. Basic TCP/IP Network Node

This is the logical structure of the layered protocols inside a
computer on an internet. Each computer that can communicate using
internet technology has such a logical structure. It is this logical
structure that determines the behavior of the computer on the
internet. The boxes represent processing of the data as it passes
through the computer, and the lines connecting boxes show the path of

data. The horizontal line at the bottom represents the Ethernet
cable; the "o" is the transceiver. The "*" is the IP address and the
"@" is the Ethernet address. Understanding this logical structure is
essential to understanding internet technology; it is referred to
throughout this tutorial.

2.2 Terminology

The name of a unit of data that flows through an internet is
dependent upon where it exists in the protocol stack. In summary: if
it is on an Ethernet it is called an Ethernet frame; if it is between
the Ethernet driver and the IP module it is called a IP packet; if it
is between the IP module and the UDP module it is called a UDP
datagram; if it is between the IP module and the TCP module it is
called a TCP segment (more generally, a transport message); and if it
is in a network application it is called a application message.

These definitions are imperfect. Actual definitions vary from one
publication to the next. More specific definitions can be found in
RFC 1122, section 1.3.3.

A driver is software that communicates directly with the network
interface hardware. A module is software that communicates with a
driver, with network applications, or with another module.

The terms driver, module, Ethernet frame, IP packet, UDP datagram,
TCP message, and application message are used where appropriate
throughout this tutorial.

2.3 Flow of Data

Let's follow the data as it flows down through the protocol stack
shown in Figure 1. For an application that uses TCP (Transmission
Control Protocol), data passes between the application and the TCP
module. For applications that use UDP (User Datagram Protocol), data
passes between the application and the UDP module. FTP (File
Transfer Protocol) is a typical application that uses TCP. Its
protocol stack in this example is FTP/TCP/IP/ENET. SNMP (Simple
Network Management Protocol) is an application that uses UDP. Its
protocol stack in this example is SNMP/UDP/IP/ENET.

The TCP module, UDP module, and the Ethernet driver are n-to-1
multiplexers. As multiplexers they switch many inputs to one output.
They are also 1-to-n de-multiplexers. As de-multiplexers they switch
one input to many outputs according to the type field in the protocol
header.

```
         1   2 3 ...   n            |            1   2 3 ...   n        ^
          \  |  |   /      |                    \  |  |    /        |
           \ | |  /        |                     \ | |   /          |
          -------------   flow              ----------------      flow
          |multiplexer|    of               |de-multiplexer|       of
          -------------   data              ----------------      data
             |          |                        |              |
             |          |                        |              |
             |          v                        |              |
             1                                    1
```
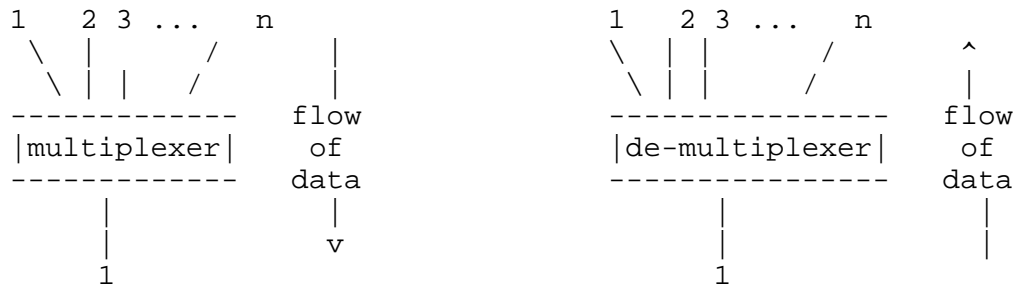
             Figure 2. n-to-1 multiplexer and 1-to-n de-multiplexer

If an Ethernet frame comes up into the Ethernet driver off the
network, the packet can be passed upwards to either the ARP (Address
Resolution Protocol) module or to the IP (Internet Protocol) module.
The value of the type field in the Ethernet frame determines whether
the Ethernet frame is passed to the ARP or the IP module.

If an IP packet comes up into IP, the unit of data is passed upwards
to either TCP or UDP, as determined by the value of the protocol
field in the IP header.

If the UDP datagram comes up into UDP, the application message is
passed upwards to the network application based on the value of the
port field in the UDP header. If the TCP message comes up into TCP,
the application message is passed upwards to the network application
based on the value of the port field in the TCP header.

The downwards multiplexing is simple to perform because from each
starting point there is only the one downward path; each protocol
module adds its header information so the packet can be de-
multiplexed at the destination computer.

Data passing out from the applications through either TCP or UDP
converges on the IP module and is sent downwards through the lower
network interface driver.

Although internet technology supports many different network media,
Ethernet is used for all examples in this tutorial because it is the
most common physical network used under IP. The computer in Figure 1
has a single Ethernet connection. The 6-byte Ethernet address is
unique for each interface on an Ethernet and is located at the lower
interface of the Ethernet driver.

The computer also has a 4-byte IP address. This address is located
at the lower interface to the IP module. The IP address must be
unique for an internet.

A running computer always knows its own IP address and Ethernet
address.

2.4 Two Network Interfaces

If a computer is connected to 2 separate Ethernets it is as in Figure
3.

```
              ---------------------------
              |    network applications  |
              |                          |
              |...  \ | /  ..  \ | /  ...|
              |     -----      -----     |
              |     |TCP|      |UDP|     |
              |     -----      -----     |
              |       \          /       |
              |       --------           |
              |       |  IP  |           |
              |  -----  -*----*-  -----  |
              |  |ARP|   |    |   |ARP|  |
              |  -----   |    |   -----  |
              |    \     |    |    /     |
              |    ------    ------      |
              |    |ENET|    |ENET|      |
              |    ---@--    ---@--      |
              ----------|-------|---------
                        |       |
                        |       ---O---------------------------
                        |            Ethernet Cable 2
                ---------------O----------
                  Ethernet Cable 1
```
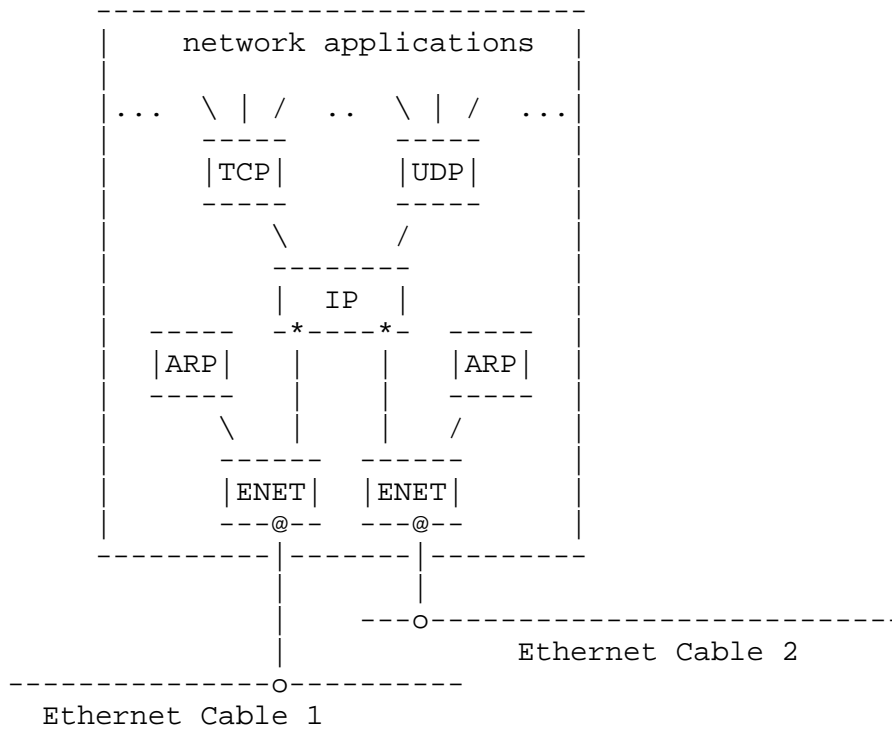
Figure 3. TCP/IP Network Node on 2 Ethernets

Please note that this computer has 2 Ethernet addresses and 2 IP
addresses.

It is seen from this structure that for computers with more than one
physical network interface, the IP module is both a n-to-m
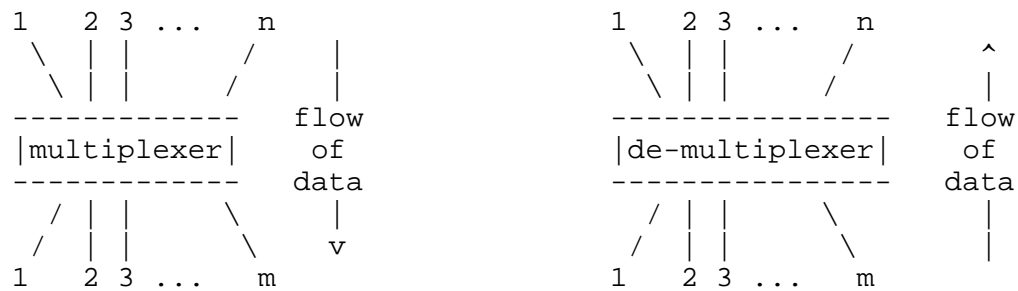multiplexer and an m-to-n de-multiplexer.

```
   1   2 3 ...   n     |            1   2 3 ...   n       ^
    \  | |     /       |             \  | |     /         |
     \ | |    /        |              \ | |    /          |
   ------------- flow        ----------------    flow
   |multiplexer|   of        |de-multiplexer|    of
   ------------- data        ----------------   data
    / | |    \         |             / | |    \          |
   /  | |     \   v           /  | |     \         |
   1   2 3 ...  m              1   2 3 ...   m
```

        Figure 4.   n-to-m multiplexer and m-to-n de-multiplexer

   It performs this multiplexing in either direction to accommodate
   incoming and outgoing data. An IP module with more than 1 network
   interface is more complex than our original example in that it can
   forward data onto the next network. Data can arrive on any network
   interface and be sent out on any other.

```
                      TCP        UDP
                        \        /
                         \      /
                     --------------
                     |     IP     |
                     |            |
                     |    ---     |
                     |   /   \    |
                     |  /     v   |
                     --------------
                      /          \
                     /            \
                  data           data
                comes in        goes out
                  here           here
```
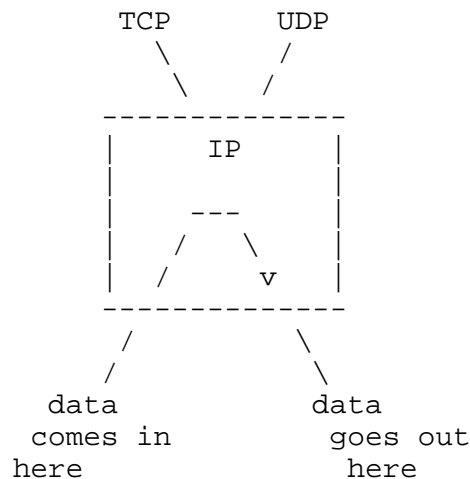
            Figure 5.   Example of IP Forwarding a IP Packet

   The process of sending an IP packet out onto another network is
   called "forwarding" an IP packet. A computer that has been dedicated
   to the task of forwarding IP packets is called an "IP-router".

   As you can see from the figure, the forwarded IP packet never touches
   the TCP and UDP modules on the IP-router. Some IP-router
   implementations do not have a TCP or UDP module.

2.5 IP Creates a Single Logical Network

   The IP module is central to the success of internet technology. Each
   module or driver adds its header to the message as the message passes

down through the protocol stack. Each module or driver strips the
corresponding header from the message as the message climbs the
protocol stack up towards the application. The IP header contains
the IP address, which builds a single logical network from multiple
physical networks. This interconnection of physical networks is the
source of the name: internet. A set of interconnected physical
networks that limit the range of an IP packet is called an
"internet".

2.6 Physical Network Independence

IP hides the underlying network hardware from the network
applications. If you invent a new physical network, you can put it
into service by implementing a new driver that connects to the
internet underneath IP. Thus, the network applications remain intact
and are not vulnerable to changes in hardware technology.

2.7 Interoperability

If two computers on an internet can communicate, they are said to
"interoperate"; if an implementation of internet technology is good,
it is said to have "interoperability". Users of general-purpose
computers benefit from the installation of an internet because of the
interoperability in computers on the market. Generally, when you buy
a computer, it will interoperate. If the computer does not have
interoperability, and interoperability can not be added, it occupies
a rare and special niche in the market.

2.8 After the Overview

With the background set, we will answer the following questions:

When sending out an IP packet, how is the destination Ethernet
address determined?

How does IP know which of multiple lower network interfaces to use
when sending out an IP packet?

How does a client on one computer reach the server on another?

Why do both TCP and UDP exist, instead of just one or the other?

What network applications are available?

These will be explained, in turn, after an Ethernet refresher.

3. Ethernet

    This section is a short review of Ethernet technology.

    An Ethernet frame contains the destination address, source address,
    type field, and data.

    An Ethernet address is 6 bytes. Every device has its own Ethernet
    address and listens for Ethernet frames with that destination
    address. All devices also listen for Ethernet frames with a wild-
    card destination address of "FF-FF-FF-FF-FF-FF" (in hexadecimal),
    called a "broadcast" address.

    Ethernet uses CSMA/CD (Carrier Sense and Multiple Access with
    Collision Detection). CSMA/CD means that all devices communicate on
    a single medium, that only one can transmit at a time, and that they
    can all receive simultaneously. If 2 devices try to transmit at the
    same instant, the transmit collision is detected, and both devices
    wait a random (but short) period before trying to transmit again.

3.1 A Human Analogy

    A good analogy of Ethernet technology is a group of people talking in
    a small, completely dark room. In this analogy, the physical network
    medium is sound waves on air in the room instead of electrical
    signals on a coaxial cable.

    Each person can hear the words when another is talking (Carrier
    Sense). Everyone in the room has equal capability to talk (Multiple
    Access), but none of them give lengthy speeches because they are
    polite. If a person is impolite, he is asked to leave the room
    (i.e., thrown off the net).

    No one talks while another is speaking. But if two people start
    speaking at the same instant, each of them know this because each
    hears something they haven't said (Collision Detection). When these
    two people notice this condition, they wait for a moment, then one
    begins talking. The other hears the talking and waits for the first
    to finish before beginning his own speech.

    Each person has an unique name (unique Ethernet address) to avoid
    confusion. Every time one of them talks, he prefaces the message
    with the name of the person he is talking to and with his own name
    (Ethernet destination and source address, respectively), i.e., "Hello
    Jane, this is Jack, ..blah blah blah...". If the sender wants to
    talk to everyone he might say "everyone" (broadcast address), i.e.,
    "Hello Everyone, this is Jack, ..blah blah blah...".

4. ARP

   When sending out an IP packet, how is the destination Ethernet
   address determined?

   ARP (Address Resolution Protocol) is used to translate IP addresses
   to Ethernet addresses. The translation is done only for outgoing IP
   packets, because this is when the IP header and the Ethernet header
   are created.

4.1 ARP Table for Address Translation

   The translation is performed with a table look-up. The table, called
   the ARP table, is stored in memory and contains a row for each
   computer. There is a column for IP address and a column for Ethernet
   address. When translating an IP address to an Ethernet address, the
   table is searched for a matching IP address. The following is a
   simplified ARP table:

```
               ------------------------------------
               |IP address          Ethernet address |
               ------------------------------------
               |223.1.2.1           08-00-39-00-2F-C3|
               |223.1.2.3           08-00-5A-21-A7-22|
               |223.1.2.4           08-00-10-99-AC-54|
               ------------------------------------
                  TABLE 1.  Example ARP Table
```

   The human convention when writing out the 4-byte IP address is each
   byte in decimal and separating bytes with a period. When writing out
   the 6-byte Ethernet address, the conventions are each byte in
   hexadecimal and separating bytes with either a minus sign or a colon.

   The ARP table is necessary because the IP address and Ethernet
   address are selected independently; you can not use an algorithm to
   translate IP address to Ethernet address. The IP address is selected
   by the network manager based on the location of the computer on the
   internet. When the computer is moved to a different part of an
   internet, its IP address must be changed. The Ethernet address is
   selected by the manufacturer based on the Ethernet address space
   licensed by the manufacturer. When the Ethernet hardware interface
   board changes, the Ethernet address changes.

4.2 Typical Translation Scenario

   During normal operation a network application, such as TELNET, sends
   an application message to TCP, then TCP sends the corresponding TCP
   message to the IP module. The destination IP address is known by the

application, the TCP module, and the IP module. At this point the IP
packet has been constructed and is ready to be given to the Ethernet
driver, but first the destination Ethernet address must be
determined.

The ARP table is used to look-up the destination Ethernet address.

4.3 ARP Request/Response Pair

But how does the ARP table get filled in the first place? The answer
is that it is filled automatically by ARP on an "as-needed" basis.

Two things happen when the ARP table can not be used to translate an
address:

   1. An ARP request packet with a broadcast Ethernet address is sent
      out on the network to every computer.

   2. The outgoing IP packet is queued.

Every computer's Ethernet interface receives the broadcast Ethernet
frame. Each Ethernet driver examines the Type field in the Ethernet
frame and passes the ARP packet to the ARP module. The ARP request
packet says "If your IP address matches this target IP address, then
please tell me your Ethernet address". An ARP request packet looks
something like this:

```
            ------------------------------------------
            |Sender IP Address   223.1.2.1            |
            |Sender Enet Address 08-00-39-00-2F-C3|
            ------------------------------------------
            |Target IP Address   223.1.2.2            |
            |Target Enet Address <blank>             |
            ------------------------------------------
               TABLE 2.   Example ARP Request
```

Each ARP module examines the IP address and if the Target IP address
matches its own IP address, it sends a response directly to the
source Ethernet address. The ARP response packet says "Yes, that
target IP address is mine, let me give you my Ethernet address". An
ARP response packet has the sender/target field contents swapped as
compared to the request. It looks something like this:

```
       -------------------------------------
      |Sender IP Address    223.1.2.2       |
      |Sender Enet Address 08-00-28-00-38-A9|
       -------------------------------------
      |Target IP Address    223.1.2.1       |
      |Target Enet Address 08-00-39-00-2F-C3|
       -------------------------------------
           TABLE 3.   Example ARP Response
```

   The response is received by the original sender computer. The
   Ethernet driver looks at the Type field in the Ethernet frame then
   passes the ARP packet to the ARP module. The ARP module examines the
   ARP packet and adds the sender's IP and Ethernet addresses to its ARP
   table.

   The updated table now looks like this:

```
         ----------------------------------
        |IP address      Ethernet address |
         ----------------------------------
        |223.1.2.1       08-00-39-00-2F-C3|
        |223.1.2.2       08-00-28-00-38-A9|
        |223.1.2.3       08-00-5A-21-A7-22|
        |223.1.2.4       08-00-10-99-AC-54|
         ----------------------------------
          TABLE 4.   ARP Table after Response
```

4.4 Scenario Continued

   The new translation has now been installed automatically in the
   table, just milli-seconds after it was needed. As you remember from
   step 2 above, the outgoing IP packet was queued. Next, the IP
   address to Ethernet address translation is performed by look-up in
   the ARP table then the Ethernet frame is transmitted on the Ethernet.
   Therefore, with the new steps 3, 4, and 5, the scenario for the
   sender computer is:

   1. An ARP request packet with a broadcast Ethernet address is sent
      out on the network to every computer.

   2. The outgoing IP packet is queued.

   3. The ARP response arrives with the IP-to-Ethernet address
      translation for the ARP table.

4. For the queued IP packet, the ARP table is used to translate the
   IP address to the Ethernet address.

5. The Ethernet frame is transmitted on the Ethernet.

In summary, when the translation is missing from the ARP table, one
IP packet is queued. The translation data is quickly filled in with
ARP request/response and the queued IP packet is transmitted.

Each computer has a separate ARP table for each of its Ethernet
interfaces. If the target computer does not exist, there will be no
ARP response and no entry in the ARP table. IP will discard outgoing
IP packets sent to that address. The upper layer protocols can't
tell the difference between a broken Ethernet and the absence of a
computer with the target IP address.

Some implementations of IP and ARP don't queue the IP packet while
waiting for the ARP response. Instead the IP packet is discarded and
the recovery from the IP packet loss is left to the TCP module or the
UDP network application. This recovery is performed by time-out and
retransmission. The retransmitted message is successfully sent out
onto the network because the first copy of the message has already
caused the ARP table to be filled.

5. Internet Protocol

The IP module is central to internet technology and the essence of IP
is its route table. IP uses this in-memory table to make all
decisions about routing an IP packet. The content of the route table
is defined by the network administrator. Mistakes block
communication.

To understand how a route table is used is to understand
internetworking. This understanding is necessary for the successful
administration and maintenance of an IP network.

The route table is best understood by first having an overview of
routing, then learning about IP network addresses, and then looking
at the details.

5.1 Direct Routing

The figure below is of a tiny internet with 3 computers: A, B, and C.
Each computer has the same TCP/IP protocol stack as in Figure 1.
Each computer's Ethernet interface has its own Ethernet address.
Each computer has an IP address assigned to the IP interface by the
network manager, who also has assigned an IP network number to the
Ethernet.

```
                   A         B         C
                   |         |         |
                 --o------o------o--
                   Ethernet 1
                   IP network "development"
```
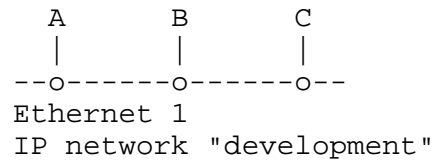
Figure 6. One IP Network

When A sends an IP packet to B, the IP header contains A's IP address
as the source IP address, and the Ethernet header contains A's
Ethernet address as the source Ethernet address. Also, the IP header
contains B's IP address as the destination IP address and the
Ethernet header contains B's Ethernet address as the destination
Ethernet address.

```
        -----------------------------------------
        |address               source  destination|
        -----------------------------------------
        |IP header             A       B          |
        |Ethernet header       A       B          |
        -----------------------------------------
```
TABLE 5. Addresses in an Ethernet frame for an IP packet
                      from A to B

For this simple case, IP is overhead because the IP adds little to
the service offered by Ethernet. However, IP does add cost: the
extra CPU processing and network bandwidth to generate, transmit, and
parse the IP header.

When B's IP module receives the IP packet from A, it checks the
destination IP address against its own, looking for a match, then it
passes the datagram to the upper-level protocol.

This communication between A and B uses direct routing.

5.2 Indirect Routing

The figure below is a more realistic view of an internet. It is
composed of 3 Ethernets and 3 IP networks connected by an IP-router
called computer D. Each IP network has 4 computers; each computer
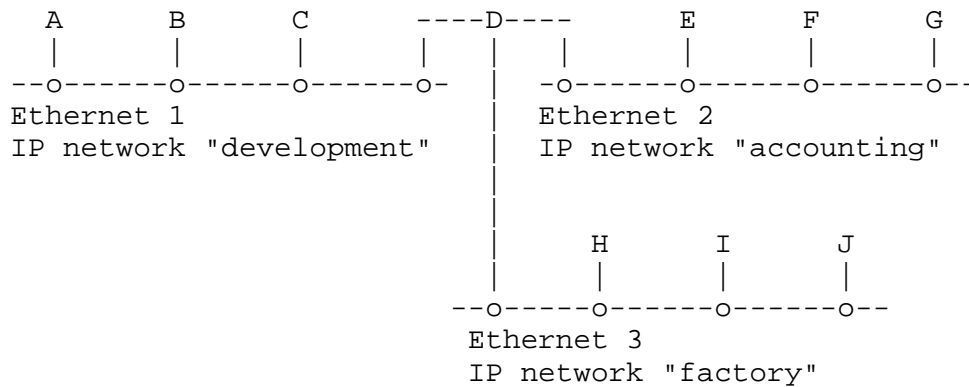has its own IP address and Ethernet address.

```
          A       B       C     ----D----     E       F       G
          |       |       |       | |         |       |       |
       --o------o------o------o-  |  -o------o------o------o--
          Ethernet 1            |   Ethernet 2
          IP network "development" |   IP network "accounting"
                                 |
                                 |
                                 |       H       I       J
                                 |       |       |       |
                      --o-----o------o------o--
                          Ethernet 3
                          IP network "factory"
```

                   Figure 7.   Three IP Networks; One internet

   Except for computer D, each computer has a TCP/IP protocol stack like
   that in Figure 1. Computer D is the IP-router; it is connected to
   all 3 networks and therefore has 3 IP addresses and 3 Ethernet
   addresses. Computer D has a TCP/IP protocol stack similar to that in
   Figure 3, except that it has 3 ARP modules and 3 Ethernet drivers
   instead of 2. Please note that computer D has only one IP module.

   The network manager has assigned a unique number, called an IP
   network number, to each of the Ethernets. The IP network numbers are
   not shown in this diagram, just the network names.

   When computer A sends an IP packet to computer B, the process is
   identical to the single network example above. Any communication
   between computers located on a single IP network matches the direct
   routing example discussed previously.

   When computer D and A communicate, it is direct communication. When
   computer D and E communicate, it is direct communication. When
   computer D and H communicate, it is direct communication. This is
   because each of these pairs of computers is on the same IP network.

   However, when computer A communicates with a computer on the far side
   of the IP-router, communication is no longer direct. A must use D to
   forward the IP packet to the next IP network. This communication is
   called "indirect".

   This routing of IP packets is done by IP modules and happens
   transparently to TCP, UDP, and the network applications.

   If A sends an IP packet to E, the source IP address and the source
   Ethernet address are A's. The destination IP address is E's, but
   because A's IP module sends the IP packet to D for forwarding, the
   destination Ethernet address is D's.

```
           -------------------------------------------
           |address              source  destination|
           -------------------------------------------
           |IP header            A       E           |
           |Ethernet header      A       D           |
           -------------------------------------------
```
      TABLE 6.   Addresses in an Ethernet frame for an IP packet
                       from A to E (before D)

   D's IP module receives the IP packet and upon examining the
   destination IP address, says "This is not my IP address," and sends
   the IP packet directly to E.

```
           -------------------------------------------
           |address              source  destination|
           -------------------------------------------
           |IP header            A       E           |
           |Ethernet header      D       E           |
           -------------------------------------------
```
      TABLE 7.   Addresses in an Ethernet frame for an IP packet
                       from A to E (after D)

   In summary, for direct communication, both the source IP address and
   the source Ethernet address is the sender's, and the destination IP
   address and the destination Ethernet address is the recipient's. For
   indirect communication, the IP address and Ethernet addresses do not
   pair up in this way.

   This example internet is a very simple one. Real networks are often
   complicated by many factors, resulting in multiple IP-routers and
   several types of physical networks. This example internet might have
   come about because the network manager wanted to split a large
   Ethernet in order to localize Ethernet broadcast traffic.

5.3 IP Module Routing Rules

   This overview of routing has shown what happens, but not how it
   happens. Now let's examine the rules, or algorithm, used by the IP
   module.

      For an outgoing IP packet, entering IP from an upper layer, IP must
      decide whether to send the IP packet directly or indirectly, and IP
      must choose a lower network interface. These choices are made by
      consulting the route table.

      For an incoming IP packet, entering IP from a lower interface, IP
      must decide whether to forward the IP packet or pass it to an upper
      layer. If the IP packet is being forwarded, it is treated as an

      outgoing IP packet.

      When an incoming IP packet arrives it is never forwarded back out
      through the same network interface.

   These decisions are made before the IP packet is handed to the lower
   interface and before the ARP table is consulted.

5.4 IP Address

   The network manager assigns IP addresses to computers according to
   the IP network to which the computer is attached. One part of a 4-
   byte IP address is the IP network number, the other part is the IP
   computer number (or host number). For the computer in table 1, with
   an IP address of 223.1.2.1, the network number is 223.1.2 and the
   host number is number 1.

   The portion of the address that is used for network number and for
   host number is defined by the upper bits in the 4-byte address. All
   example IP addresses in this tutorial are of type class C, meaning
   that the upper 3 bits indicate that 21 bits are the network number
   and 8 bits are the host number. This allows 2,097,152 class C
   networks up to 254 hosts on each network.

   The IP address space is administered by the NIC (Network Information
   Center). All internets that are connected to the single world-wide
   Internet must use network numbers assigned by the NIC. If you are
   setting up your own internet and you are not intending to connect it
   to the Internet, you should still obtain your network numbers from
   the NIC. If you pick your own number, you run the risk of confusion
   and chaos in the eventuality that your internet is connected to
   another internet.

5.5 Names

   People refer to computers by names, not numbers. A computer called
   alpha might have the IP address of 223.1.2.1. For small networks,
   this name-to-address translation data is often kept on each computer
   in the "hosts" file. For larger networks, this translation data file
   is stored on a server and accessed across the network when needed. A
   few lines from that file might look like this:

   223.1.2.1   alpha
   223.1.2.2   beta
   223.1.2.3   gamma
   223.1.2.4   delta
   223.1.3.2   epsilon
   223.1.4.2   iota

The IP address is the first column and the computer name is the
second column.

In most cases, you can install identical "hosts" files on all
computers. You may notice that "delta" has only one entry in this
file even though it has 3 IP addresses. Delta can be reached with
any of its IP addresses; it does not matter which one is used. When
delta receives an IP packet and looks at the destination address, it
will recognize any of its own IP addresses.

IP networks are also given names. If you have 3 IP networks, your
"networks" file for documenting these names might look something like
this:

```
223.1.2     development
223.1.3     accounting
223.1.4     factory
```

The IP network number is in the first column and its name is in the
second column.

From this example you can see that alpha is computer number 1 on the
development network, beta is computer number 2 on the development
network and so on. You might also say that alpha is development.1,
Beta is development.2, and so on.

The above hosts file is adequate for the users, but the network
manager will probably replace the line for delta with:

```
223.1.2.4     devnetrouter     delta
223.1.3.1     facnetrouter
223.1.4.1     accnetrouter
```

These three new lines for the hosts file give each of delta's IP
addresses a meaningful name. In fact, the first IP address listed
has 2 names; "delta" and "devnetrouter" are synonyms. In practice
"delta" is the general-purpose name of the computer and the other 3
names are only used when administering the IP route table.

These files are used by network administration commands and network
applications to provide meaningful names. They are not required for
operation of an internet, but they do make it easier for us.

5.6 IP Route Table

How does IP know which lower network interface to use when sending
out a IP packet? IP looks it up in the route table using a search
key of the IP network number extracted from the IP destination

address.

The route table contains one row for each route. The primary columns
in the route table are: IP network number, direct/indirect flag,
router IP address, and interface number. This table is referred to
by IP for each outgoing IP packet.

On most computers the route table can be modified with the "route"
command. The content of the route table is defined by the network
manager, because the network manager assigns the IP addresses to the
computers.

5.7 Direct Routing Details

To explain how it is used, let us visit in detail the routing
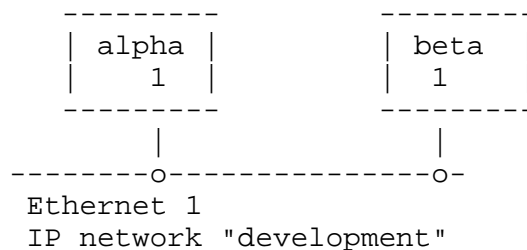situations we have reviewed previously.

```
                    ---------            ---------
                   | alpha  |           | beta   |
                   |   1    |           |   1    |
                    ---------            ---------
                        |                    |
                --------o--------------o-
                    Ethernet 1
                    IP network "development"
```

Figure 8. Close-up View of One IP Network

The route table inside alpha looks like this:

```
    ----------------------------------------------------------------
   |network        direct/indirect flag  router   interface number|
    ----------------------------------------------------------------
   |development  direct                 <blank>  1                 |
    ----------------------------------------------------------------
```
TABLE 8.   Example Simple Route Table

This view can be seen on some UNIX systems with the "netstat -r"
command. With this simple network, all computers have identical
routing tables.

For discussion, the table is printed again without the network number
translated to its network name.

```
      -----------------------------------------------------------------
      |network         direct/indirect flag  router    interface number|
      -----------------------------------------------------------------
      |223.1.2         direct                <blank>  1                |
      -----------------------------------------------------------------
```
             TABLE 9. Example Simple Route Table with Numbers

5.8 Direct Scenario

   Alpha is sending an IP packet to beta. The IP packet is in alpha's
   IP module and the destination IP address is beta or 223.1.2.2. IP
   extracts the network portion of this IP address and scans the first
   column of the table looking for a match. With this network a match
   is found on the first entry.

   The other information in this entry indicates that computers on this
   network can be reached directly through interface number 1. An ARP
   table translation is done on beta's IP address then the Ethernet
   frame is sent directly to beta via interface number 1.

   If an application tries to send data to an IP address that is not on
   the development network, IP will be unable to find a match in the
   route table. IP then discards the IP packet. Some computers provide
   a "Network not reachable" error message.

5.9 Indirect Routing Details

   Now, let's take a closer look at the more complicated routing
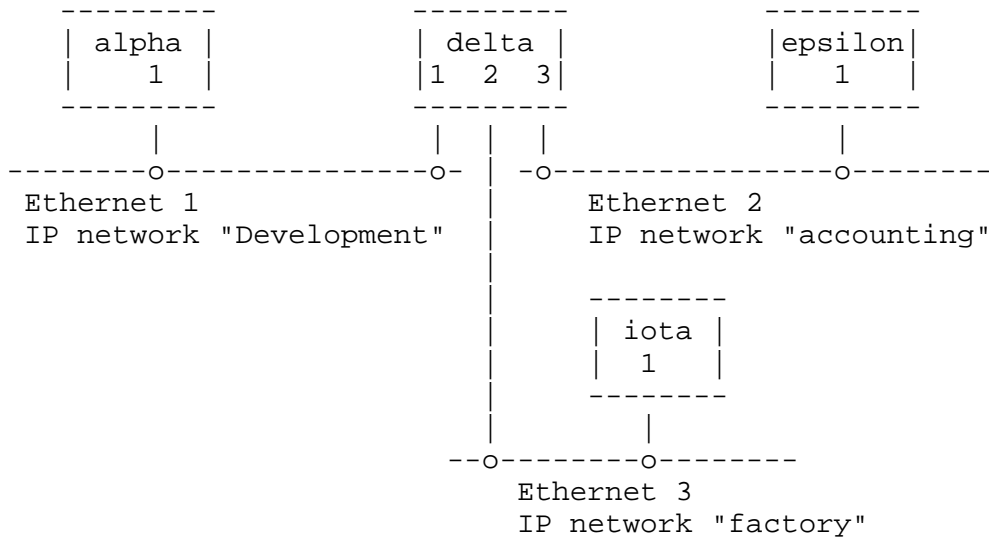   scenario that we examined previously.

```
         ---------             ---------             ---------
        | alpha |             | delta |             |epsilon|
        |   1   |             |1  2  3|             |   1   |
         ---------             ---------             ---------
            |                   |  |  |                 |
    --------o---------------o- | -o---------------o--------
      Ethernet 1             |    Ethernet 2
      IP network "Development"|    IP network "accounting"
                             |
                             |    --------
                             |   | iota |
                             |   |   1  |
                             |    --------
                             |       |
              --o--------o--------
                 Ethernet 3
                 IP network "factory"
```

                Figure 9. Close-up View of Three IP Networks

   The route table inside alpha looks like this:

```
    -------------------------------------------------------------------
    |network        direct/indirect flag  router            interface number|
    -------------------------------------------------------------------
    |development  direct                   <blank>           1              |
    |accounting   indirect                 devnetrouter      1              |
    |factory      indirect                 devnetrouter      1              |
    -------------------------------------------------------------------
                   TABLE 10.   Alpha Route Table
```

   For discussion the table is printed again using numbers instead of
   names.

```
    -------------------------------------------------------------------
    |network        direct/indirect flag  router            interface number|
    -------------------------------------------------------------------
    |223.1.2      direct                   <blank>           1              |
    |223.1.3      indirect                 223.1.2.4         1              |
    |223.1.4      indirect                 223.1.2.4         1              |
    -------------------------------------------------------------------
                 TABLE 11.   Alpha Route Table with Numbers
```

   The router in Alpha's route table is the IP address of delta's
   connection to the development network.

5.10 Indirect Scenario

   Alpha is sending an IP packet to epsilon. The IP packet is in
   alpha's IP module and the destination IP address is epsilon
   (223.1.3.2). IP extracts the network portion of this IP address
   (223.1.3) and scans the first column of the table looking for a
   match. A match is found on the second entry.

   This entry indicates that computers on the 223.1.3 network can be
   reached through the IP-router devnetrouter. Alpha's IP module then
   does an ARP table translation for devnetrouter's IP address and sends
   the IP packet directly to devnetrouter through Alpha's interface
   number 1. The IP packet still contains the destination address of
   epsilon.

   The IP packet arrives at delta's development network interface and is
   passed up to delta's IP module. The destination IP address is
   examined and because it does not match any of delta's own IP
   addresses, delta decides to forward the IP packet.

   Delta's IP module extracts the network portion of the destination IP
   address (223.1.3) and scans its route table for a matching network
   field. Delta's route table looks like this:

```
-----------------------------------------------------------------------
|network        direct/indirect flag  router            interface number|
-----------------------------------------------------------------------
|development  direct                   <blank>          1                |
|factory      direct                   <blank>          3                |
|accounting   direct                   <blank>          2                |
-----------------------------------------------------------------------
```
                    TABLE 12. Delta's Route Table

   Below is delta's table printed again, without the translation to
   names.

```
-----------------------------------------------------------------------
|network        direct/indirect flag  router            interface number|
-----------------------------------------------------------------------
|223.1.2        direct                <blank>          1                |
|223.1.3        direct                <blank>          3                |
|223.1.4        direct                <blank>          2                |
-----------------------------------------------------------------------
```
                TABLE 13. Delta's Route Table with Numbers

   The match is found on the second entry. IP then sends the IP packet
   directly to epsilon through interface number 3. The IP packet
   contains the IP destination address of epsilon and the Ethernet

destination address of epsilon.

The IP packet arrives at epsilon and is passed up to epsilon's IP
module. The destination IP address is examined and found to match
with epsilon's IP address, so the IP packet is passed to the upper
protocol layer.

5.11 Routing Summary

When a IP packet travels through a large internet it may go through
many IP-routers before it reaches its destination. The path it takes
is not determined by a central source but is a result of consulting
each of the routing tables used in the journey. Each computer
defines only the next hop in the journey and relies on that computer
to send the IP packet on its way.

5.12 Managing the Routes

Maintaining correct routing tables on all computers in a large
internet is a difficult task; network configuration is being modified
constantly by the network managers to meet changing needs. Mistakes
in routing tables can block communication in ways that are
excruciatingly tedious to diagnose.

Keeping a simple network configuration goes a long way towards making
a reliable internet. For instance, the most straightforward method
of assigning IP networks to Ethernet is to assign a single IP network
number to each Ethernet.

Help is also available from certain protocols and network
applications. ICMP (Internet Control Message Protocol) can report
some routing problems. For small networks the route table is filled
manually on each computer by the network administrator. For larger
networks the network administrator automates this manual operation
with a routing protocol to distribute routes throughout a network.

When a computer is moved from one IP network to another, its IP
address must change. When a computer is removed from an IP network
its old address becomes invalid. These changes require frequent
updates to the "hosts" file. This flat file can become difficult to
maintain for even medium-size networks. The Domain Name System helps
solve these problems.

6. User Datagram Protocol

UDP is one of the two main protocols to reside on top of IP. It
offers service to the user's network applications. Example network
applications that use UDP are: Network File System (NFS) and Simple

Network Management Protocol (SNMP). The service is little more than
an interface to IP.

UDP is a connectionless datagram delivery service that does not
guarantee delivery. UDP does not maintain an end-to-end connection
with the remote UDP module; it merely pushes the datagram out on the
net and accepts incoming datagrams off the net.

UDP adds two values to what is provided by IP. One is the
multiplexing of information between applications based on port
number. The other is a checksum to check the integrity of the data.

6.1 Ports

How does a client on one computer reach the server on another?

The path of communication between an application and UDP is through
UDP ports. These ports are numbered, beginning with zero. An
application that is offering service (the server) waits for messages
to come in on a specific port dedicated to that service. The server
waits patiently for any client to request service.

For instance, the SNMP server, called an SNMP agent, always waits on
port 161. There can be only one SNMP agent per computer because
there is only one UDP port number 161. This port number is well
known; it is a fixed number, an internet assigned number. If an SNMP
client wants service, it sends its request to port number 161 of UDP
on the destination computer.

When an application sends data out through UDP it arrives at the far
end as a single unit. For example, if an application does 5 writes
to the UDP port, the application at the far end will do 5 reads from
the UDP port. Also, the size of each write matches the size of each
read.

UDP preserves the message boundary defined by the application. It
never joins two application messages together, or divides a single
application message into parts.

6.2 Checksum

An incoming IP packet with an IP header type field indicating "UDP"
is passed up to the UDP module by IP. When the UDP module receives
the UDP datagram from IP it examines the UDP checksum. If the
checksum is zero, it means that checksum was not calculated by the
sender and can be ignored. Thus the sending computer's UDP module
may or may not generate checksums. If Ethernet is the only network
between the 2 UDP modules communicating, then you may not need

checksumming. However, it is recommended that checksum generation
always be enabled because at some point in the future a route table
change may send the data across less reliable media.

If the checksum is valid (or zero), the destination port number is
examined and if an application is bound to that port, an application
message is queued for the application to read. Otherwise the UDP
datagram is discarded. If the incoming UDP datagrams arrive faster
than the application can read them and if the queue fills to a
maximum value, UDP datagrams are discarded by UDP. UDP will continue
to discard UDP datagrams until there is space in the queue.

7. Transmission Control Protocol

   TCP provides a different service than UDP. TCP offers a connection-
   oriented byte stream, instead of a connectionless datagram delivery
   service. TCP guarantees delivery, whereas UDP does not.

   TCP is used by network applications that require guaranteed delivery
   and cannot be bothered with doing time-outs and retransmissions. The
   two most typical network applications that use TCP are File Transfer
   Protocol (FTP) and the TELNET. Other popular TCP network
   applications include X-Window System, rcp (remote copy), and the r-
   series commands. TCP's greater capability is not without cost: it
   requires more CPU and network bandwidth. The internals of the TCP
   module are much more complicated than those in a UDP module.

   Similar to UDP, network applications connect to TCP ports. Well-
   defined port numbers are dedicated to specific applications. For
   instance, the TELNET server uses port number 23. The TELNET client
   can find the server simply by connecting to port 23 of TCP on the
   specified computer.

   When the application first starts using TCP, the TCP module on the
   client's computer and the TCP module on the server's computer start
   communicating with each other. These two end-point TCP modules
   contain state information that defines a virtual circuit. This
   virtual circuit consumes resources in both TCP end-points. The
   virtual circuit is full duplex; data can go in both directions
   simultaneously. The application writes data to the TCP port, the
   data traverses the network and is read by the application at the far
   end.

   TCP packetizes the byte stream at will; it does not retain the
   boundaries between writes. For example, if an application does 5
   writes to the TCP port, the application at the far end might do 10
   reads to get all the data. Or it might get all the data with a
   single read. There is no correlation between the number and size of

writes at one end to the number and size of reads at the other end.

TCP is a sliding window protocol with time-out and retransmits.
Outgoing data must be acknowledged by the far-end TCP.
Acknowledgements can be piggybacked on data. Both receiving ends can
flow control the far end, thus preventing a buffer overrun.

As with all sliding window protocols, the protocol has a window size.
The window size determines the amount of data that can be transmitted
before an acknowledgement is required. For TCP, this amount is not a
number of TCP segments but a number of bytes.

8. Network Applications

Why do both TCP and UDP exist, instead of just one or the other?

They supply different services. Most applications are implemented to
use only one or the other. You, the programmer, choose the protocol
that best meets your needs. If you need a reliable stream delivery
service, TCP might be best. If you need a datagram service, UDP
might be best. If you need efficiency over long-haul circuits, TCP
might be best. If you need efficiency over fast networks with short
latency, UDP might be best. If your needs do not fall nicely into
these categories, then the "best" choice is unclear. However,
applications can make up for deficiencies in the choice. For
instance if you choose UDP and you need reliability, then the
application must provide reliability. If you choose TCP and you need
a record oriented service, then the application must insert markers
in the byte stream to delimit records.

What network applications are available?

There are far too many to list. The number is growing continually.
Some of the applications have existed since the beginning of internet
technology: TELNET and FTP. Others are relatively new: X-Windows and
SNMP. The following is a brief description of the applications
mentioned in this tutorial.

8.1 TELNET

TELNET provides a remote login capability on TCP. The operation and
appearance is similar to keyboard dialing through a telephone switch.
On the command line the user types "telnet delta" and receives a
login prompt from the computer called "delta".

TELNET works well; it is an old application and has widespread
interoperability. Implementations of TELNET usually work between
different operating systems. For instance, a TELNET client may be on

VAX/VMS and the server on UNIX System V.

8.2 FTP

   File Transfer Protocol (FTP), as old as TELNET, also uses TCP and has
   widespread interoperability. The operation and appearance is as if
   you TELNETed to the remote computer. But instead of typing your
   usual commands, you have to make do with a short list of commands for
   directory listings and the like. FTP commands allow you to copy
   files between computers.

8.3 rsh

   Remote shell (rsh or remsh) is one of an entire family of remote UNIX
   style commands. The UNIX copy command, cp, becomes rcp. The UNIX
   "who is logged in" command, who, becomes rwho. The list continues
   and is referred to collectively to as the "r" series commands or the
   "r*" (r star) commands.

   The r* commands mainly work between UNIX systems and are designed for
   interaction between trusted hosts. Little consideration is given to
   security, but they provide a convenient user environment.

   To execute the "cc file.c" command on a remote computer called delta,
   type "rsh delta cc file.c". To copy the "file.c" file to delta, type
   "rcp file.c delta:". To login to delta, type "rlogin delta", and if
   you administered the computers in a certain way, you will not be
   challenged with a password prompt.

8.4 NFS

   Network File System, first developed by Sun Microsystems Inc, uses
   UDP and is excellent for mounting UNIX file systems on multiple
   computers. A diskless workstation can access its server's hard disk
   as if the disk were local to the workstation. A single disk copy of
   a database on mainframe "alpha" can also be used by mainframe "beta"
   if the database's file system is NFS mounted on "beta".

   NFS adds significant load to a network and has poor utility across
   slow links, but the benefits are strong. The NFS client is
   implemented in the kernel, allowing all applications and commands to
   use the NFS mounted disk as if it were local disk.

8.5 SNMP

   Simple Network Management Protocol (SNMP) uses UDP and is designed
   for use by central network management stations. It is a well known
   fact that if given enough data, a network manager can detect and

diagnose network problems. The central station uses SNMP to collect
this data from other computers on the network. SNMP defines the
format for the data; it is left to the central station or network
manager to interpret the data.

8.6 X-Window

The X Window System uses the X Window protocol on TCP to draw windows
on a workstation's bitmap display. X Window is much more than a
utility for drawing windows; it is entire philosophy for designing a
user interface.

9. Other Information

Much information about internet technology was not included in this
tutorial. This section lists information that is considered the next
level of detail for the reader who wishes to learn more.

    o administration commands: arp, route, and netstat
    o ARP: permanent entry, publish entry, time-out entry, spoofing
    o IP route table: host entry, default gateway, subnets
    o IP: time-to-live counter, fragmentation, ICMP
    o RIP, routing loops
    o Domain Name System

10. References

    [1] Comer, D., "Internetworking with TCP/IP Principles, Protocols,
        and Architecture", Prentice Hall, Englewood Cliffs, New Jersey,
        U.S.A., 1988.

    [2] Feinler, E., et al, DDN Protocol Handbook, Volume 2 and 3, DDN
        Network Information Center, SRI International, 333 Ravenswood
        Avenue, Room EJ291, Menlow Park, California, U.S.A., 1985.

    [3] Spider Systems, Ltd., "Packets and Protocols", Spider Systems
        Ltd., Stanwell Street, Edinburgh, U.K. EH6 5NG, 1990.

11. Relation to other RFCs

This RFC is a tutorial and it does not UPDATE or OBSOLETE any other
RFC.

12. Security Considerations

There are security considerations within the TCP/IP protocol suite.
To some people these considerations are serious problems, to others
they are not; it depends on the user requirements.

This tutorial does not discuss these issues, but if you want to learn
more you should start with the topic of ARP-spoofing, then use the
"Security Considerations" section of RFC 1122 to lead you to more
information.

13. Authors' Addresses

Theodore John Socolofsky
Spider Systems Limited
Spider Park
Stanwell Street
Edinburgh EH6 5NG
United Kingdom

Phone:
   from UK   031-554-9424
   from USA 011-44-31-554-9424
Fax:
   from UK   031-554-0649
   from USA 011-44-31-554-0649

EMail: TEDS@SPIDER.CO.UK

Claudia Jeanne Kale
12 Gosford Place
Edinburgh EH6 4BJ
United Kingdom

Phone:
   from UK   031-554-7432
   from USA 011-44-31-554-7432

EMail: CLAUDIAK@SPIDER.CO.UK