IP: TCP, UDP, and ICMP

1. Try http://www.tcpipguide.com/ (previously, recommended the 3Com Technical Library, but it appears to me that this site is at least accessible)

2. USAH: Chapter 13 (TCP/IP), Chapter 14 (Routing) and Chapter 20 (Network Management and Debugging)

3. For an interesting bit of historical interest, take a look at RFC 681 at http://www.faqs.org/rfcs/rfc681.html –

such a proposal if acted on might have kept sockets and ports in Unix filename space from 1975.

IP: TCP, UDP, and ICMP

4. Many protocols can co-exist:

   (a) ISO/OSI – Deprecated : 7 layer approach. GOSIP (Government Open Systems Interconnection Profile) was a flop (http://www.itl.nist.gov/fipspubs/fip146-2.htm); here's the official announcement from 1995 repealing the 1990 FIPS146-1 procurement requirement:

What does "deprecate" mean?!

From the Wikipedia (http://www.wikipedia.org/wiki/Dep

Deprecation

In computer software standards and documentation,
deprecation is the gradual phasing-out of a software
or programming language feature.

A feature or method marked as deprecated is one
which is considered obsolete, and whose use is discouraged.

IP: TCP, UDP, and ICMP

FIPS 146-1 adopted the Government Open Systems Interconnection Profile (GOSIP) which defines a common set of Open Systems Interconnection (OSI) protocols that enable systems developed by different vendors to interoperate and the users of different applications on those systems to exchange information.
This change modifies FIPS 146-1 by removing the requirement that Federal agencies specify GOSIP protocols when they acquire networking products and services and communications systems and services.

IP: TCP, UDP, and ICMP

5. Visualizing packets - a tool to capture and display packets is very informative and instructional. Such a tool is **tshark** (previously known as **tethereal**)

IP: TCP, UDP, and ICMP

6. As a system administrator, one of your strongest debugging tools is **tethereal**. This allows you to actually see at a low level exact packet information.

Description of IP

1. 4 layer approach

2. Some layers can be viewed as combinations of multiple ISO/OSI layers

Description of IP

3. The four protocol that system administrators interact with

   (a) ARP - Address Resolution Protocol
   (b) ICMP - Internet Control Message Protocol

Description of IP

(c) UDP – User Datagram Protocol
(d) TCP – Transmission Control Protocol

4. Two main transport layer protocols are TCP and UDP

Description of IP

5. Physical network types

   (a) Ethernet (http://www.ieee802.org/3)
   (b) 802.11 wireless (http://www.ieee802.org/11)

Description of IP

(c) ATM (http://www.ietf.org/rfc/rfc1932.txt) – still exists, but is no longer really interesting to system administrators)

(d) Even IP over SCSI! (http://www.ietf.org/rfc/rfc2143.txt – experimental RFC)

Ethernet is old, but still the most important version

- [OBSOLETE] Thicknet (10Base5)
- [OBSOLETE] Thinnet (10Base2)
- Twisted Pair (10BaseT/100BaseT/1000BaseT)

Ethernet is old, but still the most important version

☞ Ethernet addresses - unique 48-bit (6 byte) MAC (Media Access Control) values examples: 00:0b:db:3f:66:27, 00:30:48:2a:29:fd (if you are doing NAT, these are the addresses that are "spoofed" by a router if your IP is locked to a particular MAC. "Spoof" means in this case that it is using that MAC address although it is not the one assigned to that port at the

factory.)

Ethernet is old, but still the most important version

- Ethernet address ranges are controlled at a manufacturer level; you can generally identify a manufacturer from the Ethernet number it is using; the current list is at http://standards.ieee.org/regauth/oui/oui.txt

- For instance, the block 00205C is owned by InterNet Systems of Florida, Inc. in Crestview, Florida.

IP interfaces

The logical view that a system administrator has of network connectivity is via an "interface".

You can see the interfaces on a machine with **ifconfig -a**

IP interfaces

Linux:

```
eth0      Link encap:Ethernet  HWaddr 00:0B:DB:3F:66:27
          inet addr:128.186.120.8  Bcast:128.186.120.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:329465 errors:0 dropped:0 overruns:0 frame:0
          TX packets:33862 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:86856566 (82.8 Mb)  TX bytes:4174751 (3.9 Mb)
          Base address:0xecc0 Memory:ff8e0000-ff900000
```

IP interfaces

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:221671 errors:0 dropped:0 overruns:0 frame:0
          TX packets:221671 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:23151065 (22.0 Mb)  TX bytes:23151065 (22.0 Mb)
```

IP interfaces

On Solaris 10:

```
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
bge0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
       inet 128.186.120.53 netmask ffffff00 broadcast 128.186.120.255
```

IP interfaces

On Windows, **ipconfig /all** does the same.

ARP – Address Resolution Protocol

ARP lets you map IP to and from MAC addresses.

Here's an ARP table from a Linux machine (**arp -a**):

```
aegis.cs.fsu.edu (128.186.120.1) at 00:0B:BE:F7:51:88 [ether] on eth0
csdc03.cs.fsu.edu (128.186.120.179) at 00:30:48:2A:29:FD [ether] on eth0
```

ARP – Address Resolution Protocol

ARP table from a Solaris 10 machine:

ARP – Address Resolution Protocol

```
% /usr/sbin/arp -a
Net to Media Table: IPv4
Device   IP Address               Mask      Flags   Phys Addr
------ -------------------- --------------- ----- ---------------
bge0   aegis.cs.fsu.edu     255.255.255.255       00:0b:be:f7:51:88
bge0   mail                 255.255.255.255       00:30:48:27:18:3c
bge0   sophie.cs.fsu.edu    255.255.255.255       00:0b:db:3f:66:27
bge0   titanic.cs.fsu.edu   255.255.255.255       00:30:48:76:22:de
bge0   omicron              255.255.255.255       00:03:ba:2f:c3:45
bge0   brain.cs.fsu.edu     255.255.255.255       00:b0:d0:7b:8b:6d
bge0   csdc02.cs.fsu.edu    255.255.255.255       00:30:48:27:43:2b
bge0   csdc03.cs.fsu.edu    255.255.255.255       00:30:48:2a:29:fd
bge0   tempest.cs.fsu.edu   255.255.255.255       00:90:27:e0:01:15
bge0   m114-8.cs.fsu.edu    255.255.255.255       00:04:75:e7:2f:d7
```

```
bge0    azaroman.cs.fsu.edu  255.255.255.255       00:0b:db:3f:65:73
bge0    ivy.cs.fsu.edu       255.255.255.255       00:0b:db:7e:ab:48
```

arp -a also works on W2K3.

```
Interface: 128.186.121.35
    Internet Address      Physical Address      Type
    128.186.121.10        08-00-20-1d-f0-37     dynamic
    128.186.121.36        00-a0-24-8e-31-06     dynamic
    128.186.121.41        08-00-20-7d-4f-49     dynamic
    128.186.121.83        00-c0-f0-16-4d-13     dynamic
```

You can also do an "ARP ping":

```
[root@localhost root]# /usr/sbin/arping csdc02.cs.fsu.edu
ARPING 128.186.120.178 from 128.186.120.8 eth0
Unicast reply from 128.186.120.178 [00:30:48:27:43:2B]  2.029ms
```

```
Unicast reply from 128.186.120.178 [00:30:48:27:43:2B]  1.092ms
Unicast reply from 128.186.120.178 [00:30:48:27:43:2B]  0.987ms
Unicast reply from 128.186.120.178 [00:30:48:27:43:2B]  0.978ms
```

tshark: a first glance

Name service is typically done via UDP, not TCP, although both are supported.

The best fundamental RFC for name service is RFC1034 (http://www.ietf.org/rfc/rfc1034.txt).

tshark: a first glance

Here's a simple nameserver lookup:

```
Capturing on eth0
1 Frame 1 (74 bytes on wire, 74 bytes captured)
    Arrival Time: Jan 31, 2006 10:27:59.500427000
    Time delta from previous packet: 0.000000000 seconds
    Time since reference or first frame: 0.000000000 seconds
    Frame Number: 1
    Packet Length: 74 bytes
    Capture Length: 74 bytes
    Protocols in frame: eth:ip:udp:dns
Ethernet II, Src: DellEsgP_3f:66:27 (00:0b:db:3f:66:27),
              Dst: Supermic_2a:29:fd (00:30:48:2a:29:fd)
    Destination: Supermic_2a:29:fd (00:30:48:2a:29:fd)
```

```
    Source: DellEsgP_3f:66:27 (00:0b:db:3f:66:27)
    Type: IP (0x0800)
Internet Protocol, Src: 128.186.120.8 (128.186.120.8),
                  Dst: 128.186.120.179 (128.186.120.179)
    Version: 4
    Header length: 20 bytes
     [  ...  ]
    Total Length: 60
     [  ...  ]
    Protocol: UDP (0x11)
     [  ...  ]
    Source: 128.186.120.8 (128.186.120.8)
    Destination: 128.186.120.179 (128.186.120.179)
User Datagram Protocol, Src Port: 32778 (32778), Dst Port: domain (53)
    Source port: 32778 (32778)
    Destination port: domain (53)
    Length: 40
    Checksum: 0x09a4 [correct]
Domain Name System (query)
    Transaction ID: 0xdc56
```

```
    Flags: 0x0100 (Standard query)
        0... .... .... .... = Response: Message is a query
        .000 0... .... .... = Opcode: Standard query (0)
        .... ..0. .... .... = Truncated: Message is not truncated
        .... ...1 .... .... = Recursion desired: Do query recursively
        .... .... .0.. .... = Z: reserved (0)
        .... .... ...0 .... = Non-authenticated data OK: Non-authenticated
                                   data is unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    Queries
        www.cs.fsu.edu: type A, class IN
            Name: www.cs.fsu.edu
            Type: A (Host address)
            Class: IN (0x0001)

2 Frame 2 (90 bytes on wire, 90 bytes captured)
    Arrival Time: Jan 31, 2006 10:27:59.500918000
```

```
    Time delta from previous packet: 0.000491000 seconds
    Time since reference or first frame: 0.000491000 seconds
    Frame Number: 2
    Packet Length: 90 bytes
    Capture Length: 90 bytes
    Protocols in frame: eth:ip:udp:dns
Ethernet II, Src: Supermic_2a:29:fd (00:30:48:2a:29:fd),
              Dst: DellEsgP_3f:66:27 (00:0b:db:3f:66:27)
    Destination: DellEsgP_3f:66:27 (00:0b:db:3f:66:27)
    Source: Supermic_2a:29:fd (00:30:48:2a:29:fd)
    Type: IP (0x0800)
Internet Protocol, Src: 128.186.120.179 (128.186.120.179),
                   Dst: 128.186.120.8 (128.186.120.8)
    Version: 4
    Header length: 20 bytes
    [  ...  ]
    Total Length: 76
    [  ...  ]
    Protocol: UDP (0x11)
    [  ...  ]
```

```
    Source: 128.186.120.179 (128.186.120.179)
    Destination: 128.186.120.8 (128.186.120.8)
User Datagram Protocol, Src Port: domain (53), Dst Port: 32778 (32778)
    Source port: domain (53)
    Destination port: 32778 (32778)
    Length: 56
    Checksum: 0xdf2c [correct]
Domain Name System (response)
    Transaction ID: 0xdc56
    Flags: 0x8580 (Standard query response, No error)
        1... .... .... .... = Response: Message is a response
        .000 0... .... .... = Opcode: Standard query (0)
        .... .1.. .... .... = Authoritative: Server is an authority for domain
        .... ..0. .... .... = Truncated: Message is not truncated
        .... ...1 .... .... = Recursion desired: Do query recursively
        .... .... 1... .... = Recursion available: Server can do recursive queries
        .... .... .0.. .... = Z: reserved (0)
        .... .... ..0. .... = Answer authenticated: Answer/authority portion
                                    was not authenticated by the server
        .... .... .... 0000 = Reply code: No error (0)
```

```
Questions: 1
Answer RRs: 1
Authority RRs: 0
Additional RRs: 0
Queries
    www.cs.fsu.edu: type A, class IN
        Name: www.cs.fsu.edu
        Type: A (Host address)
        Class: IN (0x0001)
Answers
    www.cs.fsu.edu: type A, class IN, addr 192.168.23.10
        Name: www.cs.fsu.edu
        Type: A (Host address)
        Class: IN (0x0001)
        Time to live: 1 hour
        Data length: 4
        Addr: 192.168.23.10
```

tshark continued: a look at mail exchange

Here's another type of record, an MX (mail exchange) record:

```
tethereal -V port 53
Capturing on eth0
1 Frame 1 (70 bytes on wire, 70 bytes captured)
    Arrival Time: Jan 31, 2006 10:31:00.104730000
    Time delta from previous packet: 0.000000000 seconds
    Time since reference or first frame: 0.000000000 seconds
    Frame Number: 1
    Packet Length: 70 bytes
    Capture Length: 70 bytes
    Protocols in frame: eth:ip:udp:dns
```

```
Ethernet II, Src: DellEsgP_3f:66:27 (00:0b:db:3f:66:27),
              Dst: Supermic_2a:29:fd (00:30:48:2a:29:fd)
    Destination: Supermic_2a:29:fd (00:30:48:2a:29:fd)
    Source: DellEsgP_3f:66:27 (00:0b:db:3f:66:27)
    Type: IP (0x0800)
Internet Protocol, Src: 128.186.120.8 (128.186.120.8),
                   Dst: 128.186.120.179 (128.186.120.179)
    Version: 4
    Header length: 20 bytes
    [  ...  ]
    Protocol: UDP (0x11)
    Header checksum: 0x4885 [correct]
        Good: True
        Bad : False
    Source: 128.186.120.8 (128.186.120.8)
    Destination: 128.186.120.179 (128.186.120.179)
User Datagram Protocol, Src Port: 32778 (32778), Dst Port: domain (53)
    Source port: 32778 (32778)
    Destination port: domain (53)
    Length: 36
```

```
    Checksum: 0xf824 [correct]
Domain Name System (query)
    Transaction ID: 0x68be
    Flags: 0x0100 (Standard query)
        0... .... .... .... = Response: Message is a query
        .000 0... .... .... = Opcode: Standard query (0)
        .... ..0. .... .... = Truncated: Message is not truncated
        .... ...1 .... .... = Recursion desired: Do query recursively
        .... .... .0.. .... = Z: reserved (0)
        .... .... ...0 .... = Non-authenticated data OK: Non-authenticated
                                    data is unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    Queries
        cs.fsu.edu: type MX, class IN
            Name: cs.fsu.edu
            Type: MX (Mail exchange)
            Class: IN (0x0001)
```

```
2 Frame 2 (107 bytes on wire, 107 bytes captured)
    Arrival Time: Jan 31, 2006 10:31:00.105676000
    Time delta from previous packet: 0.000946000 seconds
    Time since reference or first frame: 0.000946000 seconds
    Frame Number: 2
    Packet Length: 107 bytes
    Capture Length: 107 bytes
    Protocols in frame: eth:ip:udp:dns
Ethernet II, Src: Supermic_2a:29:fd (00:30:48:2a:29:fd),
              Dst: DellEsgP_3f:66:27 (00:0b:db:3f:66:27)
    Destination: DellEsgP_3f:66:27 (00:0b:db:3f:66:27)
    Source: Supermic_2a:29:fd (00:30:48:2a:29:fd)
    Type: IP (0x0800)
Internet Protocol, Src: 128.186.120.179 (128.186.120.179),
                   Dst: 128.186.120.8 (128.186.120.8)
    Version: 4
    Header length: 20 bytes
    [  ...  ]
    Protocol: UDP (0x11)
```

```
    Header checksum: 0x746c [correct]
        Good: True
        Bad : False
    Source: 128.186.120.179 (128.186.120.179)
    Destination: 128.186.120.8 (128.186.120.8)
User Datagram Protocol, Src Port: domain (53), Dst Port: 32778 (32778)
    Source port: domain (53)
    Destination port: 32778 (32778)
    Length: 73
    Checksum: 0xc6ba [correct]
Domain Name System (response)
    Transaction ID: 0x68be
    Flags: 0x8580 (Standard query response, No error)
        1... .... .... .... = Response: Message is a response
        .000 0... .... .... = Opcode: Standard query (0)
        .... .1.. .... .... = Authoritative: Server is an authority for domain
        .... ..0. .... .... = Truncated: Message is not truncated
        .... ...1 .... .... = Recursion desired: Do query recursively
        .... .... 1... .... = Recursion available: Server
                                  can do recursive queries
```

```
    .... .... .0.. .... = Z: reserved (0)
    .... .... ..0. .... = Answer authenticated: Answer/authority
                          portion was not authenticated by the server
    .... .... .... 0000 = Reply code: No error (0)
Questions: 1
Answer RRs: 1
Authority RRs: 0
Additional RRs: 1
Queries
    cs.fsu.edu: type MX, class IN
        Name: cs.fsu.edu
        Type: MX (Mail exchange)
        Class: IN (0x0001)
Answers
    cs.fsu.edu: type MX, class IN, preference 10, mx mail.cs.fsu.edu
        Name: cs.fsu.edu
        Type: MX (Mail exchange)
        Class: IN (0x0001)
        Time to live: 1 hour
        Data length: 9
```

```
        Preference: 10
        Mail exchange: mail.cs.fsu.edu
Additional records
    mail.cs.fsu.edu: type A, class IN, addr 128.186.120.4
        Name: mail.cs.fsu.edu
        Type: A (Host address)
        Class: IN (0x0001)
        Time to live: 1 hour
        Data length: 4
        Addr: 128.186.120.4
```

tshark: UDP can be complex, also

Here's a more complex lookup:

```
Capturing on eth0
1 Frame 1 (73 bytes on wire, 73 bytes captured)
    Arrival Time: Jan 31, 2006 10:19:50.034677000
    Time delta from previous packet: 0.000000000 seconds
    Time since reference or first frame: 0.000000000 seconds
    Frame Number: 1
    Packet Length: 73 bytes
    Capture Length: 73 bytes
    Protocols in frame: eth:ip:udp:dns
Ethernet II, Src: DellEsgP_3f:66:27 (00:0b:db:3f:66:27),
              Dst: Supermic_2a:29:fd (00:30:48:2a:29:fd)
    Destination: Supermic_2a:29:fd (00:30:48:2a:29:fd)
```

```
    Source: DellEsgP_3f:66:27 (00:0b:db:3f:66:27)
    Type: IP (0x0800)
Internet Protocol, Src: 128.186.120.8 (128.186.120.8),
                  Dst: 128.186.120.179 (128.186.120.179)
    Version: 4
    Header length: 20 bytes
    [  ...  ]
    Protocol: UDP (0x11)
    Header checksum: 0xdb64 [correct]
        Good: True
        Bad : False
    Source: 128.186.120.8 (128.186.120.8)
    Destination: 128.186.120.179 (128.186.120.179)
User Datagram Protocol, Src Port: 32778 (32778), Dst Port: domain (53)
    Source port: 32778 (32778)
    Destination port: domain (53)
    Length: 39
    Checksum: 0xce29 [correct]
Domain Name System (query)
    Transaction ID: 0xf7f5
```

```
    Flags: 0x0100 (Standard query)
        0... .... .... .... = Response: Message is a query
        .000 0... .... .... = Opcode: Standard query (0)
        .... ..0. .... .... = Truncated: Message is not truncated
        .... ...1 .... .... = Recursion desired: Do query recursively
        .... .... .0.. .... = Z: reserved (0)
        .... .... ...0 .... = Non-authenticated data OK: Non-authenticated
                                  data is unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    Queries
        www.yahoo.com: type A, class IN
            Name: www.yahoo.com
            Type: A (Host address)
            Class: IN (0x0001)


2 Frame 2 (539 bytes on wire, 539 bytes captured)
```

```
    Arrival Time: Jan 31, 2006 10:19:50.036833000
    Time delta from previous packet: 0.002156000 seconds
    Time since reference or first frame: 0.002156000 seconds
    Frame Number: 2
    Packet Length: 539 bytes
    Capture Length: 539 bytes
    Protocols in frame: eth:ip:udp:dns
Ethernet II, Src: Supermic_2a:29:fd (00:30:48:2a:29:fd),
              Dst: DellEsgP_3f:66:27 (00:0b:db:3f:66:27)
    Destination: DellEsgP_3f:66:27 (00:0b:db:3f:66:27)
    Source: Supermic_2a:29:fd (00:30:48:2a:29:fd)
    Type: IP (0x0800)
Internet Protocol, Src: 128.186.120.179 (128.186.120.179),
                   Dst: 128.186.120.8 (128.186.120.8)
    Version: 4
    Header length: 20 bytes
    [  ...  ]
    Protocol: UDP (0x11)
    Header checksum: 0xa538 [correct]
        Good: True
```

```
        Bad : False
    Source: 128.186.120.179 (128.186.120.179)
    Destination: 128.186.120.8 (128.186.120.8)
User Datagram Protocol, Src Port: domain (53), Dst Port: 32778 (32778)
    Source port: domain (53)
    Destination port: 32778 (32778)
    Length: 505
    Checksum: 0xeafe [correct]
Domain Name System (response)
    Transaction ID: 0xf7f5
    Flags: 0x8180 (Standard query response, No error)
        1... .... .... .... = Response: Message is a response
        .000 0... .... .... = Opcode: Standard query (0)
        .... .0.. .... .... = Authoritative: Server is not an authority for domain
        .... ..0. .... .... = Truncated: Message is not truncated
        .... ...1 .... .... = Recursion desired: Do query recursively
        .... .... 1... .... = Recursion available: Server can do recursive queries
        .... .... .0.. .... = Z: reserved (0)
        .... .... ..0. .... = Answer authenticated: Answer/authority portion was
                                   not authenticated by the server
```

```
    .... .... .... 0000 = Reply code: No error (0)
Questions: 1
Answer RRs: 9
Authority RRs: 10
Additional RRs: 7
Queries
    www.yahoo.com: type A, class IN
        Name: www.yahoo.com
        Type: A (Host address)
        Class: IN (0x0001)
Answers
    www.yahoo.com: type CNAME, class IN, cname www.yahoo.akadns.net
        Name: www.yahoo.com
        Type: CNAME (Canonical name for an alias)
        Class: IN (0x0001)
        Time to live: 1 minute
        Data length: 22
        Primary name: www.yahoo.akadns.net
    www.yahoo.akadns.net: type A, class IN, addr 68.142.226.52
        Name: www.yahoo.akadns.net
```

```
    Type: A (Host address)
    Class: IN (0x0001)
    Time to live: 48 seconds
    Data length: 4
    Addr: 68.142.226.52
www.yahoo.akadns.net: type A, class IN, addr 68.142.226.55
    Name: www.yahoo.akadns.net
    Type: A (Host address)
    Class: IN (0x0001)
    Time to live: 48 seconds
    Data length: 4
    Addr: 68.142.226.55
www.yahoo.akadns.net: type A, class IN, addr 68.142.226.34
    Name: www.yahoo.akadns.net
    Type: A (Host address)
    Class: IN (0x0001)
    Time to live: 48 seconds
    Data length: 4
    Addr: 68.142.226.34
www.yahoo.akadns.net: type A, class IN, addr 68.142.226.35
```

```
    Name: www.yahoo.akadns.net
    Type: A (Host address)
    Class: IN (0x0001)
    Time to live: 48 seconds
    Data length: 4
    Addr: 68.142.226.35
www.yahoo.akadns.net: type A, class IN, addr 68.142.226.37
    Name: www.yahoo.akadns.net
    Type: A (Host address)
    Class: IN (0x0001)
    Time to live: 48 seconds
    Data length: 4
    Addr: 68.142.226.37
www.yahoo.akadns.net: type A, class IN, addr 68.142.226.44
    Name: www.yahoo.akadns.net
    Type: A (Host address)
    Class: IN (0x0001)
    Time to live: 48 seconds
    Data length: 4
    Addr: 68.142.226.44
```

```
        www.yahoo.akadns.net: type A, class IN, addr 68.142.226.45
            Name: www.yahoo.akadns.net
            Type: A (Host address)
            Class: IN (0x0001)
            Time to live: 48 seconds
            Data length: 4
            Addr: 68.142.226.45
        www.yahoo.akadns.net: type A, class IN, addr 68.142.226.50
            Name: www.yahoo.akadns.net
            Type: A (Host address)
            Class: IN (0x0001)
            Time to live: 48 seconds
            Data length: 4
            Addr: 68.142.226.50
    Authoritative nameservers
        akadns.net: type NS, class IN, ns use1.akadns.net
            Name: akadns.net
            Type: NS (Authoritative name server)
            Class: IN (0x0001)
            Time to live: 10 hours, 55 minutes, 5 seconds
```

```
    Data length: 7
    Name server: use1.akadns.net
akadns.net: type NS, class IN, ns use9.akadns.net
    Name: akadns.net
    Type: NS (Authoritative name server)
    Class: IN (0x0001)
    Time to live: 10 hours, 55 minutes, 5 seconds
    Data length: 7
    Name server: use9.akadns.net
akadns.net: type NS, class IN, ns usw5.akadns.net
    Name: akadns.net
    Type: NS (Authoritative name server)
    Class: IN (0x0001)
    Time to live: 10 hours, 55 minutes, 5 seconds
    Data length: 7
    Name server: usw5.akadns.net
akadns.net: type NS, class IN, ns usw6.akadns.net
    Name: akadns.net
    Type: NS (Authoritative name server)
    Class: IN (0x0001)
```

```
    Time to live: 10 hours, 55 minutes, 5 seconds
    Data length: 7
    Name server: usw6.akadns.net
akadns.net: type NS, class IN, ns asia4.akadns.net
    Name: akadns.net
    Type: NS (Authoritative name server)
    Class: IN (0x0001)
    Time to live: 10 hours, 55 minutes, 5 seconds
    Data length: 8
    Name server: asia4.akadns.net
akadns.net: type NS, class IN, ns asia9.akadns.net
    Name: akadns.net
    Type: NS (Authoritative name server)
    Class: IN (0x0001)
    Time to live: 10 hours, 55 minutes, 5 seconds
    Data length: 8
    Name server: asia9.akadns.net
akadns.net: type NS, class IN, ns eur4.akadns.net
    Name: akadns.net
    Type: NS (Authoritative name server)
```

```
    Class: IN (0x0001)
    Time to live: 10 hours, 55 minutes, 5 seconds
    Data length: 7
    Name server: eur4.akadns.net
akadns.net: type NS, class IN, ns eur7.akadns.net
    Name: akadns.net
    Type: NS (Authoritative name server)
    Class: IN (0x0001)
    Time to live: 10 hours, 55 minutes, 5 seconds
    Data length: 7
    Name server: eur7.akadns.net
akadns.net: type NS, class IN, ns eur8.akadns.net
    Name: akadns.net
    Type: NS (Authoritative name server)
    Class: IN (0x0001)
    Time to live: 10 hours, 55 minutes, 5 seconds
    Data length: 7
    Name server: eur8.akadns.net
akadns.net: type NS, class IN, ns usc4.akadns.net
    Name: akadns.net
```

```
        Type: NS (Authoritative name server)
        Class: IN (0x0001)
        Time to live: 10 hours, 55 minutes, 5 seconds
        Data length: 7
        Name server: usc4.akadns.net
Additional records
    eur4.akadns.net: type A, class IN, addr 195.219.3.169
        Name: eur4.akadns.net
        Type: A (Host address)
        Class: IN (0x0001)
        Time to live: 1 day, 8 hours, 20 minutes, 19 seconds
        Data length: 4
        Addr: 195.219.3.169
    eur7.akadns.net: type A, class IN, addr 193.108.94.88
        Name: eur7.akadns.net
        Type: A (Host address)
        Class: IN (0x0001)
        Time to live: 18 minutes, 34 seconds
        Data length: 4
        Addr: 193.108.94.88
```

```
eur8.akadns.net: type A, class IN, addr 62.4.69.96
    Name: eur8.akadns.net
    Type: A (Host address)
    Class: IN (0x0001)
    Time to live: 18 minutes, 34 seconds
    Data length: 4
    Addr: 62.4.69.96
usc4.akadns.net: type A, class IN, addr 69.45.78.3
    Name: usc4.akadns.net
    Type: A (Host address)
    Class: IN (0x0001)
    Time to live: 1 day, 12 hours, 53 minutes, 38 seconds
    Data length: 4
    Addr: 69.45.78.3
use1.akadns.net: type A, class IN, addr 67.72.17.134
    Name: use1.akadns.net
    Type: A (Host address)
    Class: IN (0x0001)
    Time to live: 7 hours, 19 minutes, 34 seconds
    Data length: 4
```

```
    Addr: 67.72.17.134
use9.akadns.net: type A, class IN, addr 81.52.250.134
    Name: use9.akadns.net
    Type: A (Host address)
    Class: IN (0x0001)
    Time to live: 20 hours, 59 minutes, 38 seconds
    Data length: 4
    Addr: 81.52.250.134
usw5.akadns.net: type A, class IN, addr 63.241.73.200
    Name: usw5.akadns.net
    Type: A (Host address)
    Class: IN (0x0001)
    Time to live: 1 day, 15 hours, 13 minutes, 44 seconds
    Data length: 4
    Addr: 63.241.73.200
```

Characteristics of IP to bear in mind

- IP addresses assigned to NIC, not computer
- Interfaces don't have to be physical devices: Virtual interfaces "eth0:0", "eth0:1", etc.
- Another non-physical interface: Loopback device
- A computer can have multiple NICs

TCP protocol

TCP is a bit more complex than UDP, which just throws a packet on the wire. In an environmental where speed is desirable, and small packetsize is not a detriment, and there is no particular need for sequencing, UDP can be quite useful.

TCP protocol

TCP tries to be fast, but it also provides sequencing and losslessness, which fits in the general paradigm of a file as just a sequence of bytes.

TCP protocol

Let's look at a TCP connection over port 25, the SMTP MTA port:

```
[root@localhost root]# tshark -V port 25
Capturing on eth0
1 Frame 1 (74 bytes on wire, 74 bytes captured)
    Arrival Time: Jan 31, 2006 11:48:35.009104000
    Time delta from previous packet: 0.000000000 seconds
    Time since reference or first frame: 0.000000000 seconds
    Frame Number: 1
    Packet Length: 74 bytes
    Capture Length: 74 bytes
    Protocols in frame: eth:ip:tcp
```

```
Ethernet II, Src: DellEsgP_3f:66:27 (00:0b:db:3f:66:27),
                  Dst: Supermic_27:18:3c (00:30:48:27:18:3c)
    Destination: Supermic_27:18:3c (00:30:48:27:18:3c)
    Source: DellEsgP_3f:66:27 (00:0b:db:3f:66:27)
    Type: IP (0x0800)
Internet Protocol, Src: 128.186.120.8 (128.186.120.8),
                   Dst: 128.186.120.4 (128.186.120.4)
    Version: 4
    Header length: 20 bytes
    [   ...   ]
    Protocol: TCP (0x06)
    Header checksum: 0x39ee [correct]
        Good: True
        Bad : False
    Source: 128.186.120.8 (128.186.120.8)
    Destination: 128.186.120.4 (128.186.120.4)
Transmission Control Protocol, Src Port: 35433 (35433),
                               Dst Port: smtp (25), Seq: 0, Ack: 0, Len: 0
    Source port: 35433 (35433)
    Destination port: smtp (25)
```

```
    Sequence number: 0    (relative sequence number)
    Header length: 40 bytes
    Flags: 0x0002 (SYN)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...0 .... = Acknowledgment: Not set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..1. = Syn: Set
        .... ...0 = Fin: Not set
    Window size: 5840
    Checksum: 0x2105 [correct]
    Options: (20 bytes)
        Maximum segment size: 1460 bytes
        [  ...  ]


2 Frame 2 (74 bytes on wire, 74 bytes captured)
    Arrival Time: Jan 31, 2006 11:48:35.009722000
```

```
    Time delta from previous packet: 0.000618000 seconds
    Time since reference or first frame: 0.000618000 seconds
    Frame Number: 2
    Packet Length: 74 bytes
    Capture Length: 74 bytes
    Protocols in frame: eth:ip:tcp
Ethernet II, Src: Supermic_27:18:3c (00:30:48:27:18:3c),
    Dst: DellEsgP_3f:66:27 (00:0b:db:3f:66:27)
    Destination: DellEsgP_3f:66:27 (00:0b:db:3f:66:27)
    Source: Supermic_27:18:3c (00:30:48:27:18:3c)
    Type: IP (0x0800)
Internet Protocol, Src: 128.186.120.4 (128.186.120.4),
    Dst: 128.186.120.8 (128.186.120.8)
    Version: 4
    Header length: 20 bytes
    [  ...  ]
    Protocol: TCP (0x06)
    Header checksum: 0x493b [correct]
        Good: True
        Bad : False
```

```
    Source: 128.186.120.4 (128.186.120.4)
    Destination: 128.186.120.8 (128.186.120.8)
Transmission Control Protocol, Src Port: smtp (25),
                               Dst Port: 35433 (35433), Seq: 0, Ack: 1, Len: 0
    Source port: smtp (25)
    Destination port: 35433 (35433)
    Sequence number: 0    (relative sequence number)
    Acknowledgement number: 1    (relative ack number)
    Header length: 40 bytes
    Flags: 0x0012 (SYN, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..1. = Syn: Set
        .... ...0 = Fin: Not set
    Window size: 5792
    Checksum: 0x2559 [correct]
```

```
    Options: (20 bytes)
        Maximum segment size: 1460 bytes
        [  ...  ]
    SEQ/ACK analysis
        This is an ACK to the segment in frame: 1
        The RTT to ACK the segment was: 0.000618000 seconds

3 Frame 3 (66 bytes on wire, 66 bytes captured)
    Arrival Time: Jan 31, 2006 11:48:35.009786000
    Time delta from previous packet: 0.000064000 seconds
    Time since reference or first frame: 0.000682000 seconds
    Frame Number: 3
    Packet Length: 66 bytes
    Capture Length: 66 bytes
    Protocols in frame: eth:ip:tcp
Ethernet II, Src: DellEsgP_3f:66:27 (00:0b:db:3f:66:27),
              Dst: Supermic_27:18:3c (00:30:48:27:18:3c)
    Destination: Supermic_27:18:3c (00:30:48:27:18:3c)
    Source: DellEsgP_3f:66:27 (00:0b:db:3f:66:27)
    Type: IP (0x0800)
```

```
Internet Protocol, Src: 128.186.120.8 (128.186.120.8),
                   Dst: 128.186.120.4 (128.186.120.4)
    Version: 4
    Header length: 20 bytes
    [  ....  ]
    Protocol: TCP (0x06)
    Header checksum: 0x39f5 [correct]
        Good: True
        Bad : False
    Source: 128.186.120.8 (128.186.120.8)
    Destination: 128.186.120.4 (128.186.120.4)
Transmission Control Protocol, Src Port: 35433 (35433),
                               Dst Port: smtp (25), Seq: 1, Ack: 1, Len: 0
    Source port: 35433 (35433)
    Destination port: smtp (25)
    Sequence number: 1    (relative sequence number)
    Acknowledgement number: 1    (relative ack number)
    Header length: 32 bytes
    Flags: 0x0010 (ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
```

```
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 5840
    [  ...  ]
    SEQ/ACK analysis
        This is an ACK to the segment in frame: 2
        The RTT to ACK the segment was: 0.000064000 seconds

4 Frame 4 (101 bytes on wire, 101 bytes captured)
    Arrival Time: Jan 31, 2006 11:48:35.023964000
    Time delta from previous packet: 0.014178000 seconds
    Time since reference or first frame: 0.014860000 seconds
    Frame Number: 4
    Packet Length: 101 bytes
    Capture Length: 101 bytes
```

```
    Protocols in frame: eth:ip:tcp:smtp
Ethernet II, Src: Supermic_27:18:3c (00:30:48:27:18:3c),
              Dst: DellEsgP_3f:66:27 (00:0b:db:3f:66:27)
    Destination: DellEsgP_3f:66:27 (00:0b:db:3f:66:27)
    Source: Supermic_27:18:3c (00:30:48:27:18:3c)
    Type: IP (0x0800)
Internet Protocol, Src: 128.186.120.4 (128.186.120.4),
                   Dst: 128.186.120.8 (128.186.120.8)
    Version: 4
    Header length: 20 bytes
    [  ...  ]
    Protocol: TCP (0x06)
    Header checksum: 0x03ce [correct]
        Good: True
        Bad : False
    Source: 128.186.120.4 (128.186.120.4)
    Destination: 128.186.120.8 (128.186.120.8)
Transmission Control Protocol, Src Port: smtp (25),
                               Dst Port: 35433 (35433), Seq: 1, Ack: 1, Len: 35
    Source port: smtp (25)
```

```
Destination port: 35433 (35433)
Sequence number: 1    (relative sequence number)
Next sequence number: 36    (relative sequence number)
Acknowledgement number: 1    (relative ack number)
Header length: 32 bytes
Flags: 0x0018 (PSH, ACK)
    0... .... = Congestion Window Reduced (CWR): Not set
    .0.. .... = ECN-Echo: Not set
    ..0. .... = Urgent: Not set
    ...1 .... = Acknowledgment: Set
    .... 1... = Push: Set
    .... .0.. = Reset: Not set
    .... ..0. = Syn: Not set
    .... ...0 = Fin: Not set
Window size: 5792 (scaled)
Checksum: 0x44e4 [correct]
Options: (12 bytes)
    NOP
    NOP
    Time stamp: tsval 3481428865, tsecr 9604399
```

```
Simple Mail Transfer Protocol
    Response: 220 mail.cs.fsu.edu ESMTP Postfix\r\n
        Response code: 220
        Response parameter: mail.cs.fsu.edu ESMTP Postfix

5 Frame 5 (66 bytes on wire, 66 bytes captured)
    Arrival Time: Jan 31, 2006 11:48:35.024014000
    Time delta from previous packet: 0.000050000 seconds
    Time since reference or first frame: 0.014910000 seconds
    Frame Number: 5
    Packet Length: 66 bytes
    Capture Length: 66 bytes
    Protocols in frame: eth:ip:tcp
Ethernet II, Src: DellEsgP_3f:66:27 (00:0b:db:3f:66:27),
             Dst: Supermic_27:18:3c (00:30:48:27:18:3c)
    Destination: Supermic_27:18:3c (00:30:48:27:18:3c)
    Source: DellEsgP_3f:66:27 (00:0b:db:3f:66:27)
    Type: IP (0x0800)
Internet Protocol, Src: 128.186.120.8 (128.186.120.8),
                  Dst: 128.186.120.4 (128.186.120.4)
```

```
    Version: 4
    Header length: 20 bytes
    [  ...  ]
    Protocol: TCP (0x06)
    Header checksum: 0x39f4 [correct]
        Good: True
        Bad : False
    Source: 128.186.120.8 (128.186.120.8)
    Destination: 128.186.120.4 (128.186.120.4)
Transmission Control Protocol, Src Port: 35433 (35433),
                               Dst Port: smtp (25), Seq: 1, Ack: 36, Len: 0
    Source port: 35433 (35433)
    Destination port: smtp (25)
    Sequence number: 1    (relative sequence number)
    Acknowledgement number: 36    (relative ack number)
    Header length: 32 bytes
    Flags: 0x0010 (ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
```

```
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 5840
    Checksum: 0x53be [correct]
    Options: (12 bytes)
        NOP
        NOP
        Time stamp: tsval 9604400, tsecr 3481428865
    SEQ/ACK analysis
        This is an ACK to the segment in frame: 4
        The RTT to ACK the segment was: 0.000050000 seconds
```

TCP: important points for system administrators

- ☞ Did the SYN packet go out?
- ☞ Did it get an ACK/SYN back?

TCP: important points for system administrators

☞ Did the maximum segment size (MSS) look reasonable (1460 is good, occasionally will see much smaller, which is not so great for high volume web servers.)

TCP: important points for system administrators

☞ Did a PUSH happen, and did it have expected data over the correct port?

☞ **strace** and **tshark** are two of the system administrator's best tools.

Other useful tools

ping – can do both UDP pings and ICMP pings

traceroute – **may** be useful to see where a network blockage might be

Other useful tools

tcpdump – similiar to **tshark** (in fact, **tshark** uses some of the libraries from **tcpdump**) (libpcap)

netstat – lets you see network status. **netstat -rn** is particularly useful since it can let you see the routing table

Other useful tools

route – lets you manipulate routing tables

Routing Theory

- ☞ Why do we need routing?
- ☞ Machines on same network don't need it
- ☞ Two disparate physical nets do need it

Routing Theory

- ☞ Routers/Gateways (often counted as slightly different, but we will use the terms interchangeably)
- ☞ Main types of routing

Routing Theory

- Static routes
  - Entered manually; gathered via DHCP; gather with "router discovery" (e.g. RFC 1256)

Routing Theory

- Every machine should have at least one: the default route
- Method for adding (**route** command)
- Adding an imaginary route to met.fsu.edu through xi would be:

Routing Theory

```
route add -net 128.186.5.0 netmask 255.255.255.0 gw 128.186.121.41 (Linux)
route add net 128.186.5.0 128.186.121.41 (SunOS/Solaris)
```

Routing Theory

- Dynamic
  - Uses routing daemons, **routed** or **gated**

Routing Theory

- ☞ Kernel routing table (**netstat -rn**)
- ☞ How does routing work? Do we have routes to everywhere?

Dynamic routing

- ☞ Distance vector, e.g. RIP
- ☞ Link state, e.g. OSPF
- ☞ Outside our area, e.g. BGP

Dynamic routing

☞ This has been an active area for networkers (with lots and lots of protocols such as RIP-2, IGRP, EIGRP, IS-IS, MOSPF, DVMRP, PIM, and so on), but increasing irrelevant for system administrators, who are largely using "static" routing.

"Static" routing

☞ There are at least three different ways to implement "static" routes:

"Static" routing

1. Put it in "very" statically with the **route** program; at boottime, either `/etc/sysconfig/network` (Linux) or `/etc/defaultrouter` (Solaris) is checked for an entry.
2. Use DHCP to pick up the information.

"Static" routing

3. Use router discovery via RFC1256 (http://www.ietf.org/rf
   ICMP:

   ```
   This document specifies an alternative router discovery method using a
   pair of ICMP [10] messages, for use on multicast links.  It
   eliminates the need for manual configuration of router addresses and
   is independent of any specific routing protocol.
   ```

"Static" routing

On Solaris, **in.routed** also understands ICMP router discovery. On Linux, it is still done via a separate **rdisc** daemon.

Fitting it all together

- ☞ System administrators typical use tools on machines to debug network problems
- ☞ **ping** (ICMP) is a good candidate to discover if a host is up or down, and to see if network connectivity has been lost to a net

Fitting it all together

- **traceroute** is also useful program to see exactly how packets are traversing the network
- Finally, **tcpdump**/**tshark** are also useful to make sure traffic is proceeding well

Firewalls

In the past, firewalls were not nearly as important as they are today, particularly for interior machines. While it has been generally recognized for a long time that firewalls were important for outward facing machines, with the proliferation of malware, it is now conventional wisdom that one should enable firewalls on virtually all non-isolated machines.

Firewalls

On the Linux side, this generally means running **iptables**.

Firewalls

The configuration for iptables is generally found in `/etc/sysconfig/iptables`:

```
# Firewall configuration written by redhat-config-securitylevel
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
```

```
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Firewalls

For instance, if you wanted to let this machine serve SMTP, you could add a rule:

```
...
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp --destination-port 25 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
...
```

Ubiquitous computing: Linux as a router appliance

☞ While it isn't quite embedded processing as it were, you will find Linux distributions that focus on providing a router appliance. From there, you can get into many more interesting issues such as VLANs, CIDR addressing, and a more in-depth study of routing protocols.

Ubiquitous computing: Linux as a router appliance

☞ Free Cisco (http://www.freesco.org/)

Ubiquitous computing: Linux as a router appliance

- LEAF (http://leaf.sourceforge.net/)
- Linksys firewalls: the WRT54G model is a popular Linux platform (most people doing this should get a 16 megabyte or bigger version); DD-WRT is pretty good