IP Networking

An introduction & some debugging hints

Adrian Gschwend

netlabs.org - Open Source Software

Warpstock Europe 2009

Internet Networking

- Started in late sixties (ARPANET)
- Military grade network (fault tolerant)
- Packet switching (unlike circuit switching)
- *Stable* since early eighties
- Documented in RFCs (Request for Comments)

Networking Stacks

- Early implementation in Unix systems
- Free implementation in BSD Unix
- Still the base for many open and closed source IP stacks
- Well documented in *TCP/IP Illustrated Volume II* (W. Richard Stevens)

eComStation IP Stack

- IPv4 only
- BSD derived, behaves mostly like it
- NDIS stack for drivers (old version)

IPv4 addresses

- Each node in the network has a IP address and a netmask e.g. 192.168.0.4 with netmask 255.255.255.0
- The IP address space is 32 bit long (2^{32} or a bit less than 4.3 billion IPs)
- The netmask divides an IP address into a network and a client portion
- The division will be made by a bitwise AND operation

Subnetworking

Calculate the wildcard range:

```
192.168.0.4     = 11000000.10101000.00000000.00000100
255.255.255.0   = 11111111.11111111.11111111.00000000
Wildcard        = 00000000.00000000.00000000.11111111
=====================================================
Range           = 11000000.10101000.00000000.00000000
```

The first 24 bit are the network portion, the last 8 bit are the client portion. Thus we have an 8 bit range (= 256) of addresses in the network.

CIDR Notation

CIDR notation of an IP address puts a forward slash (/) and a decimal number, which represents the size of the prefix (network portion), after the IP address:

- 192.168.0.0/24 could be written 192.168.0.0/255.255.255.0
- 192.168.0.0/22 could be written 192.168.0.0/255.255.252.0

As you can see CIDR notation is shorter than the dot-decimal notation.

Data Link Layer

- IP needs an interface to the physical layer, to be able to communicate through a network
- The Data Link Layer is such an interface
- A common technology these days is Ethernet
- Ethernet connects IP with the physical layer

The Layers

#	Data Unit	Layer	Function	Example
4	Segments	Transport	end-to-end connections and reliability	TCP, UDP
3	Packets	Network	Path determining and logical addressing	IP, ICMP
2	Frames	Data Link	Physical addressing	ARP
1	Bits	Physical	Media, signal and binary transmissions	RS-232, T1 802.11*, DSL

IP routing

- IP uses a routing table and knows that it needs to reach a certain gateway which is able to reach Google
- Routing table complexity ranges from as simple as only having a default gateway up to routers with >200’000 entries in their routing table
- Such routing decisions are handled by the operating system and thus this process is completly transparent for applications and users

Route decisions

- If the destined receiver is within the same network portion, it is assumed that the node is local
- If the destined receiver is within a different network portion, a lookup in the routing table will be made and the packets are sent to the router

```
user@host:~$ /sbin/ip route get 72.14.215.99
72.14.215.99 via 192.168.0.1 dev eth0  src 192.168.0.4
    cache  mtu 1500 advmss 1460 hoplimit 64
```

(Linux only)

IP - Conclusion

- IP is used to send single packets from one node to another node
- IP has its own addressing schema to address nodes
- IP always relies on a *local transport medium*, the data link layer, such as Ethernet
- The data link layer has it's own rules when it comes to addressing and address resolution
- The separation of the data link layer makes IP independent of a specific transport media - thus, an IP packet may travel over Ethernet, ADSL, ATM and other technologies on its way to the destination node

Address Resolution Protocol – ARP

The Address Resolution Protocol (ARP) is used to find a hardware address for a particular network layer address. Due to the overwhelming prevalence of the TCP/IP protocol family and Ethernet it is primarily used to translate IP addresses into MAC addresses. But it will also be used for other networking technologies like IP over ATM or Token Ring.

ARP in `tcpdump`

ARP transmits ARP requests to the networks broadcast address:

```
host:~# tcpdump -nv -i eth1 "arp"
tcpdump: listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
10:58:12.053496 arp who-has 192.168.0.1 tell 192.168.0.3
10:58:12.054843 arp reply 192.168.0.1 is-at 00:00:24:c2:3b:a9
```

The ARP cache

You can use the command `arp` to get the content of your local ARP cache.

- `arp -a` (display all arp entries)
- `arp -an` condensed BSD style output

Talk with a local computer

Issue a `ping` to a computer within the same local LAN and watch that in the sniffer:

```
10:42:10.969652 arp who—has 192.168.0.3 tell 192.168.0.4
10:42:10.970521 arp reply 192.168.0.3 is—at 00:0d:93:00:4d:c7
10:42:10.970549 IP 192.168.0.4 > 192.168.0.3: ICMP echo request, id 41511, seq 1
, length 64
10:42:10.971232 IP 192.168.0.3 > 192.168.0.4: ICMP echo reply, id 41511, seq 1,
length 64
```

What we have seen

- 192.168.0.4 asks for the MAC address of 192.168.0.3
- 00:0d:93:00:4d:c7 is the lookup result
- The two machines exchange IP packets
- ICMP was transmitted too later

Talk with the Internet

Issue a ping to **www.google.ch** and watch that in the sniffer:

```
10:52:01.178539 arp who-has 192.168.0.1 tell 192.168.0.4
10:52:01.179901 arp reply 192.168.0.1 is-at 00:00:24:c5:3a:a8
10:52:01.182029 IP 192.168.0.4 > 72.14.215.99: ICMP echo request, id 58408, seq
1, length 64
10:52:01.274295 IP 72.14.215.99 > 192.168.0.4: ICMP echo reply, id 58408, seq 1,
 length 64
```

What we have seen

- 192.168.0.4 asks for the MAC address of 192.168.0.1
- 192.168.0.1 is our default gateway
- Thus, on Ethernet layer we address our default gateway
- But on IP layer we still address the Google web server

“Pinging” with ARP

Run a sniffer while `arping` is running to see what it does:

```
host:~# arping -c 2 -I eth0 192.168.0.1
ARPING 192.168.0.1 from 192.168.0.3 eth0
Unicast reply from 192.168.0.1 [00:00:24:c2:3b:a9]  2.111ms
Unicast reply from 192.168.0.1 [00:00:24:c2:3b:a9]  1.630ms
Sent 2 probes (1 broadcast(s))
Received 2 response(s)
```

I could not find `arping` on eComStation.

`arping`

- `arping` sends out a first ARP request to the broadcast address
- After that, it sends the ARP requests to the wanted host only
- By measuring the time between ARP request and answer the utility is capable of calculating the round-trip time (like `ping` does)
- Since `arping` works on the data link layer it allows to “ping” hosts that are configured to not answer to traditional ping
- This is especially useful since answering ARP requests cannot be turned off without loosing network connectivity!

Limitations of ARP

ARP is limited to the local network only. Therefore it can only be used to lookup addresses from all hosts within the same network segment as you are.

Internet Control Message Protocol (ICMP)

- ICMP is another core protocol of the Internet
- It has been standardized in 1980 in RFC792
- ICMP is an integral part of IP
- It is primarily used by operating systems to exchange error messages
- It is usually not used directly by user network applications, with some notable exceptions being the `ping` tool and `traceroute`

Common ICMP types

- `Echo` (type 8) and `Echo Reply` (0) are used by `ping`
- `Time Exceeded` (type 11) is used by `traceroute`
- `Destination Unreachable` (type 3) is used if hosts or ports are not reachable

More interesting ICMP types

There are three more interesting ICMP types:

- `Redirect` (type 5) - alter a hosts routing table
- `Address Mask` (type 17 request, 18 reply) - request subnet mask from a host
- `Timestamp` (type 13) - request a timestamp from a host

Purpose of upper layers

- With IP we're only able to address hosts – what if we need to address services running on a host?
- With IP we're only able to exchange single packets – what if we need a "session management"?

Transmission Control Protocol (TCP)

- Research began in 1973
- TCP has been standardized in 1981 in RFC793
- Since then there have been a lot of enhancements developed
- TCP builds upon IP - thus often the term *TCP/IP* protocol is used
- Since TCP is not the only protocol that builds upon IP, TCP/IP is rather a slang term than a technical description

Properties of TCP

TCP has convenient properties:

- TCP allows to create “virtual channels” between two nodes - both nodes can communicate bi-directionally
- Each packet gets a *sequence number* - TCP assures that data will be assembled at the correct order on the receiving side
- The receiving side always has to acknowledge received packets - if the sender does not receive an acknowledge within a reasonable time frame a re-transmit occurs
- Each packet gets a checksum - the receiver checks this and is able to detect data corruption during transmit

Ports

- TCP introduces port numbers as an additional addressing characteristic
- A port number is a unsigned 16-bit number
- This means, ports are in the range of 0 - 65535
- Ports are used to distinguish different services or applications running on the same node
- A TCP connection is always identified by four tokens:
  - source IP
  - source port
  - destination IP
  - destination port

netstat

netstat is a command-line tool that displays:

- network connections (incoming and outgoing)
- routing tables
- various network statistics

The tool exists for various operating systems and works with different command-line switches depending on the platform you use. Consult your man page or help system for details.

Established connections

The operating system tracks all currently open connections:

```
user@host:~$ netstat -n --tcp
Active Internet connections (w/o servers)
Proto Local Address       Foreign Address     State
tcp   192.168.0.4:35284   192.168.0.85:5223   ESTABLISHED
tcp   192.168.0.4:53440   10.42.137.61:5223   ESTABLISHED
tcp   192.168.0.4:46043   10.50.1.25:143      ESTABLISHED
tcp   192.168.0.4:47722   172.16.31.236:22    ESTABLISHED
```

On eComStation: `netstat -s`

Listen sockets

The operating system tracks all listening sockets:

```
user@host:~$ netstat -an --tcp
Active Internet connections (servers and established)
Proto Local Address   Foreign Address   State
tcp   0.0.0.0:8010    0.0.0.0:*         LISTEN
tcp   0.0.0.0:6000    0.0.0.0:*         LISTEN
tcp   127.0.0.1:631   0.0.0.0:*         LISTEN
```

A *socket* is defined by its protocol, the local IP & port and the distant IP & port.

On eComStation: `netstat -tl`

Connections

- TCP has a well defined procedure to establish and close connections
  - Handshake to establish connection (SYN, SYN/ACK, ACK)
  - Connection ending (FIN)
- This connection management makes it easy to implement stateful firewalls
- An operating system can (virtually) handle as much connections as needed, as long as each connection is uniquely identifieable through the four tokens

User Datagram Protocol (UDP)

- UDP is another layer 4 protocol
- Research on UDP begun on 1977, when a more lightweight companion to TCP was needed
- It has been standardized in 1980 in RFC768

Properties of UDP

- In contrast to TCP, UDP is much simpler
- UDP only does addressing and no error checking or session management
- UDP does not make sure that sent data are received in the right order, nor does it detect when packets are lost
- This makes it preferred choice for some applications like DNS, VOIP or streaming where latency is more important than thorough error checking
- Additionally, UDP is also prefered by some VPN solutions because tunneling TCP in TCP may have unwanted side actions

Things to check I

- IP address: `ifconfig` (everyone else), `ipconfig` (Windows)
- Resolve hostname: `host`, `nslookup`, `dig`, `ping`
- Ping host/default gateway: `ping`, `arping`
- Check ARP cache: `arp -an`
- Check DNS servers:`/etc/resolv.conf` (Unix systems), `\mptn\etc\resolv2` (eComStation)
- Check routing table: `netstat -rn` (BSD stacks, Linux, Windows), `route` (Linux only)

Things to check II

- Check routing: `traceroute` (Unix), `tracerte` (eComStation), `tracert` (Windows), `tcptraceroute`, `mtr`
- Check link status: `ifconfig` (BSD stacks), `mii-tool` (Linux), not available on eComStation due to NDIS limitations
- Check physical cabling: Gigabit is sensitive!

Possible issues

- DNS fails: IP address, DNS servers, routing table
- No IP: ARP cache, link status, cabling
- No Internet: check routing
- TCP/UDP issues: `netcat`, `netio` (many platforms), `iperf` (Unix platforms)
- Everything else: use `tcpdump`

Questions