Computer Networks - Xarxes de Computadors

Outline
Course Syllabus
Unit 1: Introduction
Unit 2: IP Networks
Unit 3: Point to Point Protocols - TCP
Unit 4: LANs
Unit 5: Data Transmission

Unit 2: IP Networks

Outline
IP layer service
IP addresses
Subnetting
Routing tables
ARP protocol
IP header
ICMP protocol
DHCP protocol
NAT
DNS
Routing algorithms
Security in IP

IP Layer Service
Internet Protocol (IP) goal is routing datagrams.
IP main design goal was interconnecting hosts attached to LANs/WANs of different technologies.
IP was designed independently of any existing network technology.
IP characteristics are:
Connectionless
Stateless
Best effort

High Performance Routers

"There is a major upgrade going on at service providers upgrading their core networks," Chris Howes, director of service provider marketing at Juniper, said.

"The next generation core network is all about meshing the ability to support any service. 11,000 is delivering No. 1 in scale. It is an access control switch. It is not fancy. All the metrics matter. Important for a service provider."

The keys to the performance throughput on the Juniper T700 are the 100Gbps capable vlinks that can support all the major serviceability features that carriers may need. Among these features support for OC-749 (45.76Gbps), OC-192 (10Gbps) and OC-48c (10 Gigabit Ethernet).

Juniper (www.juniper.net)

Cisco (www.cisco.com)
Unit 2: IP Networks

**Outline**
- IP layer service: ICMP protocol
- **IP addresses**: DHCP protocol
- Subnetting: NAT
- Routing tables: DNS
- ARP protocol: Routing algorithms
- IP header: Security in IP

**IP Addresses (RFC 791)**

![IP Datagram Header Diagram]

**IP Addresses**
32 bits (4 bytes).
Dotted point notation: Four bytes in decimal, e.g. 147.83.24.28
netid identifies the network.
hostid identifies the host within the network.
An IP address identifies an *interface*: an attachment point to the network.
All IP addresses in Internet must be different. To achieve this goal, Internet Assigned Numbers Authority, IANA (http://www.iana.net) assign address blocs to Regional Internet Registries, RIR:
APNIC: ASIA http://www.apnic.net.
RIR assign addresses to ISPs, and ISPs to their customers.

**IP Addresses - Classes**
The highest bits identify the class.
The size of netid/hostid varies in classes A/B/C.
D Class is for multicast addresses (e.g. 224.0.0.2: “all routers”)
E Class are reserved addresses.

<table>
<thead>
<tr>
<th>Classe</th>
<th>netid (bytes)</th>
<th>hostid (bytes)</th>
<th>Codification</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>3</td>
<td>0xxxx...x</td>
<td>0.0.0.0 ~ 127.255.255.255</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>2</td>
<td>10xxxx...x</td>
<td>128.0.0.0 ~ 191.255.255.255</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>1</td>
<td>110xxxx...x</td>
<td>192.0.0.0 ~ 223.255.255.255</td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>-</td>
<td>1110xxxx...x</td>
<td>224.0.0.0 ~ 239.255.255.255</td>
</tr>
<tr>
<td>E</td>
<td>-</td>
<td>-</td>
<td>1111xxxx...x</td>
<td>240.0.0.0 ~ 255.255.255.255</td>
</tr>
</tbody>
</table>
Unit 2: IP Networks

IP Addresses – Special Addresses
Special addresses cannot be used for a physical interface.
Each network has two special addresses: network and broadcast addresses.

<table>
<thead>
<tr>
<th>netid</th>
<th>hostid</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxx</td>
<td>all ‘0’</td>
<td>Identifies a network. It is used in routing tables.</td>
</tr>
<tr>
<td>xxx</td>
<td>all ‘1’</td>
<td>Broadcast in the net. xxx</td>
</tr>
<tr>
<td>all ‘0’</td>
<td>all ‘0’</td>
<td>Identifies “this host” in “this net.” Used as source address in configuration protocols, e.g. DHCP.</td>
</tr>
<tr>
<td>all ‘1’</td>
<td>all ‘1’</td>
<td>broadcast in “this net.” Used as destination address in configuration protocols, e.g. DHCP.</td>
</tr>
<tr>
<td>127</td>
<td>xxx</td>
<td>host loopback: interprocess communication with TCP/IP.</td>
</tr>
</tbody>
</table>

Example:

![Network Diagram]

Unit 2: IP Networks

IP Addresses – Private Addresses (RFC 1918)
Most commercial OSs include the TCP/IP stack.
TCP/IP is used to network many kind of electronic devices:

Addresses assigned to RIRs by IANA are called public, global or registered.
What if we arbitrarily assign a registered address to a host? It may be filtered by our ISP or cause trouble to the right host using that address.
Private addresses has been reserved for devices not using public addresses. These addresses are not assigned to any RIR (are not unique). There are addresses in each class:
1 class A network: 10.0.0.0
16 class B networks: 172.16.0.0 ~ 172.31.0.0
256 class C networks: 192.168.0.0 ~ 192.168.255.0

Unit 2: IP Networks

Outline
- IP layer service
- ICMP protocol
- IP addresses
- DHCP protocol
- Subnetting
- NAT
- Routing tables
- DNS
- ARP protocol
- Routing algorithms
- IP header
- Security in IP

Subnetting (RFC 950)
Initially the netid was given by the address class: A with 2^24 addresses, B with 2^16 addresses and C with 2^8 addresses.
What if we want to divide the network?

Subnetting allows adding bits from the hostid to the netid (called subnetid bits).
Example: For the ISP the network prefix is 24 bits. For the internal router the network prefix is 26 bits. The 2 extra bits allows 4 “subnetworks”.
A mask is used to identify the size of the netid+subnetid prefix.
Mask notations:
dotted, as 255.255.255.192
giving the number of bits as 210.50.30.0/26
Unit 2: IP Networks

IP Addresses – Subnetting Example
We want to subnet the address 210.50.30.0/24 in 4 subnets

\[ B = 210.50.30 \]

<table>
<thead>
<tr>
<th>subnet</th>
<th>subnetid</th>
<th>IP net. addr.</th>
<th>range</th>
<th>broadcast</th>
<th>available</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>00</td>
<td>B.0/26</td>
<td>B.0 ~ B.63</td>
<td>B.63</td>
<td>(2^5 - 2 = 62)</td>
</tr>
<tr>
<td>S2</td>
<td>01</td>
<td>B.64/26</td>
<td>B.64 ~ B.127</td>
<td>B.127</td>
<td>(2^5 - 2 = 62)</td>
</tr>
<tr>
<td>S3</td>
<td>10</td>
<td>B.128/26</td>
<td>B.128 ~ B.191</td>
<td>B.191</td>
<td>(2^5 - 2 = 62)</td>
</tr>
<tr>
<td>S4</td>
<td>11</td>
<td>B.192/26</td>
<td>B.192 ~ B.255</td>
<td>B.255</td>
<td>(2^5 - 2 = 62)</td>
</tr>
</tbody>
</table>

Unit 2: IP Networks

IP Addresses – Classless Inter-Domain Routing, CIDR (RFC 1519)
Initially, Internet backbone Inter-Domain routing tables did not use masks: netid was derived from the IP address class.
When the number of networks in Internet started growing exponentially, routing tables size started exploding.
In order to reduce routing tables size, CIDR proposed a “rational” geographical-based distribution of IP addresses to be able to “summarize routes”, and use masks instead of classes.
Summarization example:

\[
\begin{align*}
200.1.10.0/24 & \quad 200.1.10.0/24 \rightarrow 200.1.10.0/23
\end{align*}
\]

Unit 2: IP Networks

IP Addresses – Variable Length Subnet Mask (VLSM)
Subnetworks of different sizes.
Example, subnetworking a class C address:
We have 1 byte for subnetid + hostid.
Subnetid is green, chosen subnets addresses are underlined.

<table>
<thead>
<tr>
<th>subnet</th>
<th>subnetid</th>
<th>IP net. addr.</th>
<th>range</th>
<th>broadcast</th>
<th>available</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>0</td>
<td>B.0/25</td>
<td>B.0 ~ B.127</td>
<td>B.127</td>
<td>(2^7 - 2 = 126)</td>
</tr>
<tr>
<td>S2</td>
<td>10</td>
<td>B.128/26</td>
<td>B.128 ~ B.191</td>
<td>B.191</td>
<td>(2^8 - 2 = 62)</td>
</tr>
<tr>
<td>S3</td>
<td>1100</td>
<td>B.192/28</td>
<td>B.192 ~ B.207</td>
<td>B.207</td>
<td>(2^4 - 2 = 14)</td>
</tr>
<tr>
<td>S4</td>
<td>1101</td>
<td>B.208/28</td>
<td>B.208 ~ B.223</td>
<td>B.223</td>
<td>(2^4 - 2 = 14)</td>
</tr>
<tr>
<td>S5</td>
<td>1110</td>
<td>B.224/28</td>
<td>B.224 ~ B.239</td>
<td>B.239</td>
<td>(2^4 - 2 = 14)</td>
</tr>
<tr>
<td>S6</td>
<td>1111</td>
<td>B.240/28</td>
<td>B.240 ~ B.255</td>
<td>B.255</td>
<td>(2^4 - 2 = 14)</td>
</tr>
</tbody>
</table>

Outline
IP layer service
IP addresses
Subnetting
Routing tables
ARP protocol
IP header
ICMP protocol
DHCP protocol
NAT
DNS
Routing algorithms
Security in IP
Routing Table

ip_output() kernel function consults the routing table for each datagram.

Routing can be:

Direct: The destination is directly connected to an interface.
Indirect: Otherwise.

Default route: Is an entry where to send all datagrams with a destination address to a network not present in the routing table. The default route address is 0.0.0.0/0.

Hosts usually have two entries: The network where they are connected and a default route.

if (Datagram Destination == address of any of the interfaces) {
    send the datagram to upper layers
}

for each routing table entry ordered from longest to shortest mask

if (Datagram Destination IP address & mask) == Destination table entry {
    return (gateway, interface);
}

3. Forward the datagram

if (it is a direct routing) {
    send the datagram to the Datagram Destination IP address
} else {
    if it is an indirect routing /*
        send the datagram to the gateway IP address
    */
}
Unit 2: IP Networks

Outline
IP layer service  ICMP protocol
IP addresses  DHCP protocol
Subnetting  NAT
Routing tables  DNS
ARP protocol  Routing algorithms
IP header  Security in IP

Address Resolution Protocol, ARP (RFC 826)

To send the datagram, IP layer may have to pass a "physical address" to the NIC driver.
ARP translate IP addresses to "physical addresses" (used by the physical network).
If needed, IP calls ARP module to obtain the "physical addresses" before the NIC driver call.
Physical addresses are also called MAC or hardware addresses.

Ethernet example:

ARP tables:
A: /sbin/arp -n
Address 147.83.34.123 Unicast ether 00:14:F1:CC:59:00 Flags Mask ether C
B: /sbin/arp -n
Address 147.83.34.123 unicast ether 00:0c:49:d5:96:d8 Flags Mask ether C
C: /sbin/arp -n
Address 147.83.34.123 unicast ether 00:0c:49:d5:96:d8 Flags Mask ether C

ARP tables:
A: /sbin/arp -n
Address 147.83.34.123 Unicast ether 00:14:F1:CC:59:00 Flags Mask ether C
B: /sbin/arp -n
Address 147.83.34.123 unicast ether 00:0c:49:d5:96:d8 Flags Mask ether C
C: /sbin/arp -n
Address 147.83.34.123 unicast ether 00:0c:49:d5:96:d8 Flags Mask ether C

"Completed" flag
Unit 2: IP Networks

Address Resolution Protocol – Message format (ethernet)

ARP messages are encapsulated directly in a data-link frame.

```
<table>
<thead>
<tr>
<th>Hardware Type (14)</th>
<th>Protocol Type (14)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware Length (8)</td>
<td>Protocol Length (8)</td>
</tr>
<tr>
<td>Operation (16)</td>
<td></td>
</tr>
<tr>
<td>Hardware Address (48)</td>
<td>Protocol Address (32)</td>
</tr>
<tr>
<td>Hardware Address (48)</td>
<td>Target Hardware Address (48)</td>
</tr>
<tr>
<td>Hardware Address (48)</td>
<td>Target Protocol Address (32)</td>
</tr>
</tbody>
</table>
```

ARP tables:
```
A # /sbin/arp -i eth0 -s 10.0.0.5 00:00:39:7e:06:3b pub  
A # /sbin/arp -n  
Address  Hardware Address  Flags Mask  Iface  
10.0.0.5  00:00:39:7e:06:3b  C  etht0  
10.0.0.5  00:00:39:7f:1e:14  a0  etht0  
B # /sbin/arp -n  
Address  Hardware Address  Flags Mask  Iface  
10.0.0.5  00:00:39:7e:04:3b  C  etht0  
```

Routing table of host A:
```
A # route -n  
Destination   Gateway  Genmask   Flags Metric Ref    Use Iface  
10.0.0.5        0.0.0.0         255.255.255.255 U     0      0        0 ppp0  
10.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0  
```

Unit 2: IP Networks

Address Resolution Protocol – Proxy ARP

```
2 unicast  
1 broadcast: 20:02:25.681331 arp who-has 10.0.0.20 tell 10.0.0.20  
```

Goals:
- Detect duplicated IP addresses.
- Update ARP table MAC addresses after an IP or NIC change.

Outline
- IP layer service
- ICMP protocol
- IP addresses
- DHCP protocol
- Sub netting
- NAT
- Routing tables
- DNS
- ARP protocol
- Routing algorithms
- IP header
- Security in IP
**Unit 2: IP Networks**

**IP Header (RFC 791)**

Version: 4
IP Header Length (IHL): Header size in 32 bit words.
Type of Service: (ToS): x$x$x$x$x$x$x$x
Total Length: Datagram size in bytes.
Identification/Flags/Fragment Offset: used in fragmentation.
Time to Live (TTL): rt(--TTL==0) { discard ; }.
Protocol: Encapsulated protocol (/etc/protocols in unix).
Header Checksum: Header error detection.
Source and Destination Addresses: End nodes addresses.

```
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Version</td>
<td>IHL</td>
<td>Type of Service</td>
<td>Total Length</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Identification</td>
<td>Flags</td>
<td>Fragment Offset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Source Address</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Destination Address</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Options</td>
<td>Padding</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

20 bytes

---

**IP Fragmentation**

Router: Fragmentation may be needed when two networks with different Maximum Transfer Unit (MTU) are connected.
Host: Fragmentation may be needed using UDP. TCP segments are ≤ MTU.
Datagrams are reconstructed at the destination.
Fields:
Identification (16 bits): identify fragments from the same datagram.
Flags (3 bits):
D, don't fragment. Used in MTU path discovery
M, More fragments: Set to 0 only in the last fragment
Offset (13 bits): Position of the fragment first byte in the original datagram in 8 byte words (indexed at 0).

---

**Unit 2: IP Networks**

**IP Fragmentation - Example**

Original datagram = 4464 bytes (4Mbps Token Ring): 20 header + 4444 payload.

Fragment size = $\frac{1500 - 20}{8} = 185$ 8-byte-words (1480 bytes)

1st fragment: offset = 0, M = 1. 0–1479 payload bytes.
2nd fragment: offset = 185, M = 1. 1480–2959 payload bytes.
4th fragment: offset = 555, M = 0. 4440–4443 payload bytes.

---

**MTU Path Discovery**

Used in modern TCP implementations.
TCP by default choses the maximum segment size, to avoid headers overhead (TCP payload / (TCP payload + Σ TCP,IP,Data-link,Physical headers))
Goal: avoid fragmentation:
Unit 2: IP Networks

Outline
IP layer service  ICMP protocol
IP addresses      DHCP protocol
Subnetting        NAT
Routing tables    DNS
ARP protocol      Routing algorithms
IP header         Security in IP

Internet Control Message Protocol, ICMP (RFC 792)
Used for attention and error messages.
Can be generated by IP, TCP/UDP, and application layers.
Are encapsulated into an IP datagram.
Can be: (i) query, (ii) error.
An ICMP error message cannot generate another ICMP error message (to avoid loops).

ICMP general format message (RFC 792)
Query type messages have an identifier field, for request-reply correspondence.
Error messages have a field where the first 8 bytes of the datagram payload causing the error are copied. These bytes capture the TCP/UDP ports.

Common ICMP messages

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>query/error</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>query</td>
<td>echo reply</td>
<td>Reply on an echo request</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>error</td>
<td>network unreachable</td>
<td>Network not in the RT</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>error</td>
<td>host unreachable</td>
<td>ARP cannot solve the address.</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>error</td>
<td>protocol unreachable</td>
<td>IP cannot deliver the payload.</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>error</td>
<td>port unreachable</td>
<td>TCP/UDP cannot deliver the payload.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>error</td>
<td>fragmentation needed but DF set</td>
<td>MTU path discovery</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>error</td>
<td>source quench</td>
<td>Sent by a congested router.</td>
</tr>
<tr>
<td>0</td>
<td>8</td>
<td>query</td>
<td>echo request</td>
<td>Request for reply</td>
</tr>
<tr>
<td>0</td>
<td>11</td>
<td>error</td>
<td>TTL=0 during transit</td>
<td>Sent by a router when --TTL=0</td>
</tr>
</tbody>
</table>
Unit 2: IP Networks

Outline
IP layer service
IP addresses
Subnetting
Routing tables
ARP protocol
IP header
ICMP protocol
DHCP protocol
NAT
DNS
Routing algorithms
Security in IP

Dynamic Host Configuration Protocol, DHCP (RFC 2131)

Implements and can interoperate with previous BOOTP protocol.

Used for automatic network configuration:
Assign IP address and mask,
Default route,
Hostname,
DNS domain,
Configure DNS servers,
etc.

IP address configuration can be:
Dynamic: During a leasing time.
Automatic: Unlimited leasing time.
Manual: IP addresses are assigned to specific MAC addresses.

DHCP – Protocol Messages (RFC 2131)

DHCPDISCOVER - Client broadcasts to locate available servers.
DHCPOFFER - Server to client in response to DHCPDISCOVER with offer of configuration parameters.
DHCPREQUEST - Client message to servers either (a) requesting offered parameters from one server and implicitly declining offers from all others, (b) confirming correctness of previously allocated address after, e.g., system reboot, or (c) extending the lease on a particular network address.
DHCPACK - Server to client with configuration parameters, including committed network address.
DHCPNAK - Server to client indicating client's notion of network address is incorrect (e.g., client has moved to new subnet) or client's lease as expired.
DHCPDECLINE - Client to server indicating network address is already in use.
DHCPRELEASE - Client to server relinquishing network address and cancelling remaining lease.
DHCPINFORM - Client to server, asking only for local configuration parameters; client already has externally configured network address.

DHCP – Message Fields (RFC 2131)

<table>
<thead>
<tr>
<th>FIELD</th>
<th>OCTETS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>op</td>
<td>1</td>
<td>Message op code / message type. 1 = BOOTREQUEST, 2 = BOOTREPLY.</td>
</tr>
<tr>
<td>htype</td>
<td>1</td>
<td>Hardware address type.</td>
</tr>
<tr>
<td>hlen</td>
<td>1</td>
<td>Hardware address length.</td>
</tr>
<tr>
<td>hops</td>
<td>1</td>
<td>Client sets to zero, optionally used by relay agents when booting via a relay agent.</td>
</tr>
<tr>
<td>xid</td>
<td>4</td>
<td>Transaction ID, a random number chosen by the client, used by the client and server to associate messages and responses between a client and a server.</td>
</tr>
<tr>
<td>secs</td>
<td>2</td>
<td>Filled in by client, seconds elapsed since client began address acquisition or renewal process.</td>
</tr>
<tr>
<td>flags</td>
<td>2</td>
<td>Flags.</td>
</tr>
<tr>
<td>ciaddr</td>
<td>4</td>
<td>Client IP address; only filled in if client is in BOUND, REBIND or REBINDING state and can respond to RARP requests.</td>
</tr>
<tr>
<td>yiaddr</td>
<td>4</td>
<td>&quot;your&quot; (client) IP address. Set by the server in a DHCPOFFER message.</td>
</tr>
<tr>
<td>siaddr</td>
<td>4</td>
<td>IP address of next server to use in bootstrap; returned in DHCPOFFER, DHCPACK by server.</td>
</tr>
<tr>
<td>giaddr</td>
<td>4</td>
<td>Relay agent IP address, used in booting via a relay agent.</td>
</tr>
<tr>
<td>chaddr</td>
<td>16</td>
<td>Client hardware address.</td>
</tr>
<tr>
<td>name</td>
<td>64</td>
<td>Optional server host name, null terminated string.</td>
</tr>
<tr>
<td>file</td>
<td>128</td>
<td>Boot file name, null terminated string; &quot;generic&quot; name or null in DHCPOFFER, fully qualified directory-path name in DHCPOFFER.</td>
</tr>
<tr>
<td>options</td>
<td>var</td>
<td>Optional parameters field.</td>
</tr>
</tbody>
</table>
Unit 2: IP Networks

DHCP – Client-server interaction (RFC 2131)

UDP, server port = 67, client port = 68.

The client can directly send DHCPREQUEST:
After rebooting if it remembers and wishes to reuse a previously allocated network address.
Extending the lease on a particular network address.

DHCP – Example: tcpdump/dhcpdump capture

```
linux # tcpdump -lenx -s 1500 -i eth0 port bootps or port bootpc | dhcpdump
```

```
DHCPREQUEST
```

```
DHCPDISCOVER
```

```
DHCPACK
```

Can be unicast or broadcast, if requested by the client.

```
DHCPACK
```

```
DHCPREQUEST
```

```
DHCPDISCOVER
```

Can be unicast or broadcast, if requested by the client.

```
DHCPACK
```

```
DHCPREQUEST
```

```
DHCPDISCOVER
```

```
DHCPACK
```

Network Address Translation, NAT (RFCs 1631, 2663 3022)

Typical scenario: Private addresses (internal addresses) are translated to public addresses (external addresses).
A NAT table is used for address mapping.
Advantages:
- Save public addresses.
- Security.
- Administration, e.g. changing ISP does not imply changing private network addressing.

Outline

IP layer service
IP addresses
DHCP protocol
Subnetting
NAT
Routing tables
DNS
ARP protocol
Routing algorithms
IP header
Security in IP
Unit 2: IP Networks

NAT – Types of translations

NOTE: NAT is a technique, not a protocol. Implementations and terminology may change from one manufacturer to another.

Basic NAT:
A different external address is used for each internal address → a different public IP address is needed for each hosts accessing Internet.
Each NAT table entry has the tuple: (internal address, external address).
Each host requires one NAT table entry.
Port and Address Translation, PAT:
The same external address can be used for each internal address → a unique public IP address can be used for all hosts accessing Internet.
Each NAT table entry has the tuple: (int. address/port, ext. address/port)
Each connection requires one NAT table entry.
The NAT table entries can be:
Static: Manually added.
Dynamic: Entries are automatically added when an internal connection is initiated. 
External addresses are chosen from a pool.
Table entries have a timeout.

DNAT

What if we want external connections to internal servers? (DNAT in linux-iptables terminology).
The address translation is exactly the same as NAT, but, the connection is initiated from an external client.
Typically, some static configuration is needed to configure the server IP/port.

iptables -t NAT -A POSTROUTING -j SNAT --to-source 80.102.191.191

Information of established connections is recorded by the “connection tracking” module. Connection information is used as “NAT table”.

Driver
FORWARDING table
Local Processes
Routing
INPUT
OUTPUT
INPUT
OUTPUT
Post-routing NAT (SNAT)
Routing
Connection tracking table
Outgoing packets
Incoming packets
Linux routing chains.

iptables -t NAT -A POSTROUTING -j SNAT --to-source 80.102.191.191

iptables -t NAT -A PREROUTING -p tcp –dport 22 -j DNAT --to-destination 192.168.1.101

snat=84.120.112.212 sport=1730 dport=1755 src=84.120.112.222 dst=80.102.191.191

dst=217.125.101.197 sport=5770 dport=4941
tcp 6 SYN_RECV src=84.120.112.222 dst=217.125.101.197 [ASSURED]
sync-4662 dport=4598 [ASSURED]

192.168.1.101

Protocol-name, protocol-number, timeout(seconds), [tcp-state], received IP/port src/dst, expected

Assured packets in both directions.
Unit 2: IP Networks

NAT – ADSL commercial router example
NAT outgoing packets to 80.102.191.191
DNAT incoming packets, port 22 (ssh) to 192.168.1.100

```
# telnet 192.168.1.1
Trying 192.168.0.1...Connected to 192.168.1.1. =>nat[nat]=>list
```

<table>
<thead>
<tr>
<th>NAT</th>
<th>Index</th>
<th>Port</th>
<th>Inside-address:Port</th>
<th>Outside-address:Port</th>
<th>Protocol</th>
<th>Flg</th>
<th>Expire</th>
<th>State</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNAT</td>
<td>5</td>
<td>6</td>
<td>192.168.1.100:22</td>
<td>80.102.191.191:423</td>
<td>TCP</td>
<td>0</td>
<td>0</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>DNAT</td>
<td>11</td>
<td>6</td>
<td>192.168.1.100:423</td>
<td>80.102.191.123:22</td>
<td>TCP</td>
<td>0</td>
<td>0</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>SNAT</td>
<td>12</td>
<td>6</td>
<td>192.168.1.100:22</td>
<td>80.102.191.100:423</td>
<td>TCP</td>
<td>0</td>
<td>0</td>
<td>Open</td>
<td>1</td>
</tr>
</tbody>
</table>

Unit 2: IP Networks

Outline

- IP layer service
- IP addresses
- Subnetting
- Routing tables
- ARP protocol
- IP header

ICMP protocol
DHCP protocol
NAT
DNS
Routing algorithms
Security in IP

Domain Name System DNS (RFC 1034, 1035)

Allows users to use names instead of IP addresses: e.g. rogent.ac.upc.edu instead of 147.83.31.7, www.upc.edu instead of 147.83.194.21, etc.

Names consists of a node-name and a domain-name: rogent.ac.upc.edu, www.upc.edu

DNS consists of a worldwide distributed data base, organized hierarchically.

DNS data base entries are referred to as Resource Records (RR).

The information associated with a name is composed of 1 or more RRs.

Names are case insensitive (e.g. www.upc.edu and WWW.UPC.EDU are equivalent).

DNS – Domain Hierarchy

The Internet Corporation for Assigned Names and Numbers (ICANN) is responsible for managing and coordinating the DNS.

ICANN delegates Top Level Domains (TLD) administration to registrars: http://www.internic.net

Domains delegate the administration of subdomains.
Unit 2: IP Networks

DNS – Data Base Organization
Access to DNS data base is done using Name Servers (NS).
NSs may hold permanent and cached RRs. Cached RRs are removed after a timeout.
Each subdomain has an authority which consists of a primary and backup NSs.
In this context, subdomains are referred to as zones, and delegated subdomains subzones.
An authority has the complete information of a zone:
Names and addresses of all nodes within the zone.
Names and addresses of all subzone authorities.
Root Servers are the entry point to the domain hierarchy.
Root Servers distributed around the world and have the TLD addresses:
www.root-servers.org
Root server addresses are needed in a NS configuration.

Unit 2: IP Networks

DNS – Unix example: The resolver
The applications use the calls (resolver library):
struct hostent *gethostbyname(const char *name);
struct hostent *gethostbyaddr(const void *addr, int len, int type);
The resolver first looks the /etc/hosts file:
# hosts This file describes a number of hostname-to-address mappings for the TCP/IP subsystem. It is mostly used at boot time, when no name servers are running.
On small systems, this file can be used instead of a "named" name server.
# Syntax.
# IP-Address Full-Qualified-Hostname Short-Hostname
127.0.0.1 localhost
10.0.1.1 massanilla.ar.upc.edu massanilla

Otherwise a name server is contacted using /etc/resolv.conf file:
search ar.upc.edu
nameserver 147.83.32.3
nameserver 147.83.33.4

Unit 2: IP Networks

DNS – Protocol
Client-server paradigm
UDP/TCP. Short messages uses UDP.
well-known port: 53

http://www.foo.org

Unit 2: IP Networks

DNS – Unix example: Basic NS configuration
Unix NS implementation is BIND (Berkeley Internet Name Domain),
named is the BIND NS daemon.
BIND basic configuration files:
/etc/named.conf global configuration
/var/lib/named/root.hint root servers addresses
/var/lib/named/*.db zone files
Unit 2: IP Networks

DNS – Unix example: zone file

```
#cat /etc/bind/named.conf
zone "example.com" { type master; file "/etc/bind/db.example.com";}
```

DNS – Resolution

NSs cache name resolutions.

A cached RR is returned without looking for in the NS authority.

The same name may be associated with several IP addresses (e.g. load balancing).

The addresses of a common domain may not belong to the same IP network (e.g. Content Distribution Networks).

Example using dig:

```
dig www.microsoft.com
```

```
; <<>> DiG 9.3.2 <<>> www.microsoft.com
;; global options:  printcmd
;; Got answer:
;; Got no errors across queries:
181 IN CNAME lb1.www.ms.akadns.net. www.microsoft.com
215 IN CNAME g.www.ms.akadns.net. www.microsoft.com
```

```
Name server lb1.www.ms.akadns.net.
```

```
Address: 207.46.19.60
```

```
Address: 207.46.18.30
```

```
Address: 207.46.198.30
```

```
Address: 207.46.225.60
```

```
;; Query time: 42 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; MSG SIZE  rcvd: 203
```
Unit 2: IP Networks

DNS – Messages: Header
Identification: 16 random bits used to match query/response
Flags. Some of them:
Query-Response, QR: 0 for query, 1 for response.
Recursion Desired, RD: When set, indicates that recursion is desired.
The other fields indicate the number of Questions, Answer, Authority and Additional fields of the message.

Unit 2: IP Networks

DNS – Messages: Question
QName: Indicates the name to be resolved.
QType: Indicates the question type:
Address, A.
Name Server, NS.
Pointer, PTR: For an inverse resolution.
Mail Exchange, MX: Domain Mail Server address.
Qclass: For Internet addresses is 1.
Unit 2: IP Networks

DNS – Messages: Example

```
# tcpdump -s1500 -vvvni eth0 port 53

tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 200 bytes
11:17:30.769328 IP (UDP, length: 55) 147.83.30.137.1042 > 147.83.30.70.53: 36388+ A? ns.uu.net. (27)
11:17:30.771324 IP (UDP, length: 145) 147.83.30.70.53 > 147.83.30.137.1042: 36388
q: A? ns.uu.net.: 1/2/2 ns.uu.net. A 137.39.1.3
ar: auth00.ns.uu.net. A 198.6.1.65, auth60.ns.uu.net. A 198.6.1.181 (117)
```

Query message:
36388: Identifier.
+ : Recursion-Desired is set.
A? : Qtype = A.

na uu.net.: Name to resolve.
Response message:
36388: Identifier.
q: A? ns.uu.net.: Repeat the Question field.
1/2/2: 1 Answers, 2 Authorities, 2 Additional follows.
na uu.net.: The answer (RR of type A, address: 137.39.1.3).
n: ns uu.net.: NS auth00.ns uu.net., ns uu.net. NS auth60.ns uu.net.
ar: auth00.ns uu.net. A 198.6.1.65, auth60.ns uu.net. A 198.6.1.181 (117)
Unit 2: IP Networks
Routing algorithms - Autonomous Systems (AS)

AS definition (RFC 1930): “An AS is a connected group of one or more IP prefixes run by one or more network operators which has a SINGLE and CLEARLY DEFINED routing policy”.

Each AS is identified by a 16 bits AS Number (ASN) assigned by IANA. ASs facilitate Internet routing by introducing a two-level hierarchy: “IGP and EGP domains”.

“IGP domain”: Each domain is identified by a ASN. AS paths are used instead of metrics. Advertised AS paths depend on the routing preferences between ASes.

“EGP domain”: metrics are used to find the set of “best paths” between IGP networks.

Routing Information Protocol, RIP (RFC 2453)

The metric (distance) to a destination is the number of hops (i.e., transmissions) to reach the destination: 1 if the destination is attached to a directly connected network, 2 if 1 additional router is needed ...

Routers send updates every 30 seconds to the neighbors.

RIP messages use UDP, src./dst. port = 520, broadcast dst. IP addr.

RIP updates include destinations and metrics tuples.

A neighbor is considered down if no RIP messages are seen during 180 seconds.

Infinite metric is 16.

Two versions of RIP: Version 2 allows variable masks ans uses the multicast dst. address 244.0.0.9 (all RIPv2 routers).

The routing algorithm is known as “distance-vector” or “Bellman-Ford algorithm”.

RIP – Routing Table (RT) Update Example

When an update message from Rj is received:

Increase the message metrics.

Add new destinations.

Change entries with other routers with larger metrics.

Update metrics using Rj’s gateway.

RIP – Count to Infinity

Depending on the route update message order, convergence problems may arise:

Evolution of D=N4 entry when R3 fails:
Unit 2: IP Networks

RIP – Count to Infinity Solutions
Split horizon: When the router sends the update, removes the entries having a gateway in the interface where the update is sent:

![Diagram](image)

Split horizon with Poisoned Reverse: Consists of adding the entries having a gateway with $M=16$.
Triggered updates: Consists of sending the update before the 30 seconds timer expires when a metric change in the routing table.
Hold down timer (CISCO): When a route becomes unreachable (metric = 16), the entry is placed in holddown during 280 seconds. During this time, the entry is not updated.

Unit 2: IP Networks

Open Shortest Path First, OSPF (RFC 2328)
IETF standard for high performance IGP routing protocol.
Link State protocol: Routers monitor neighbor routers and networks and send this information to all OSPF routers (Link State Advertisements, LSA). LSA are encapsulated into IP datagrams with multicast destination address 224.0.0.5, and routed using flooding.
LSA are only sent when changes in the neighborhood occur, or when a LSA Request is received.
Neighbor routers are monitored using a hello protocol.
OSPF routers maintain a LS database with the information received with LSA. The Shortest Path First algorithm (Dijkstra algorithm) is used to optimal build routing table entries.
The metric is computed taking into account link bitrates, delays etc.
The infinite metric is the maximum metric value.
There is no count to infinity.

Unit 2: IP Networks

Border Gateway Protocol, BGP (RFC 1771, 1772)
BGP is the routing protocol used among ASs in Internet:

![Diagram](image)

"EGP domain": Each domain is identified by a ASN. All paths are used instead of metrics. Advertised AS paths depend on the routing preferences between ASs.

GDP domain": metrics are used to find the set of "best paths" between BGP networks.

Stub: Only carries local traffic and is connected to only one AS.
Multihomed: Only carries local traffic and is connected to more than one AS.
Transit: Route traffic from other ASs.
Unit 2: IP Networks

Outline
IP layer service  ICMP protocol
IP addresses  DHCP protocol
Subnetting  NAT
Routing tables  DNS
ARP protocol  Routing algorithms
IP header  Security in IP

Security in IP
Goals:
Confidentiality: Who can access.
Integrity: Who can modify the data.
Availability: Access guarantee.
Vulnerabilities:
Technological: Protocols (e.g. ftp and telnet send messages in “clear text”) and networking devices (routers...)
Configuration: Servers, passwords, ...
Missing security policies: Secure servers, encryption, firewalls, ...

Security in IP – Attacks
Reconnaissance: Previous to an attack.
Available IP addresses.
Available servers and ports.
Types of OSs, versions, devices...
Eavesdropping
Access: Unauthorized access to an account or service.
Denial of Service: Disables or corruptions networks, systems, or services.
Worms, viruses, trojan horses...: Malicious software that replicate itself.

Security in IP – Basic Solutions
Firewalls.
Virtual Private Networks (VPN).
Unit 2: IP Networks

Security in IP – Firewalls

Firewall: System or group of systems that enforces an access control policy to a network.

There are many firewall types: From simple packet filtering based on IP/TCP/UDP header rules, to state-full connection tracking and application-based filtering, defense against network attacks, ...

---

Unit 2: IP Networks

Security in IP – Basic Firewall Configuration

NAT
Access Control List, ACL

<table>
<thead>
<tr>
<th>Protocol</th>
<th>IP-src</th>
<th>IP-dst</th>
<th>Port-src</th>
<th>Port-dst</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>any</td>
<td>200.200.10.10/32</td>
<td>any</td>
<td>80</td>
<td>accept</td>
</tr>
<tr>
<td>TCP</td>
<td>any</td>
<td>any</td>
<td>&lt; 1024</td>
<td>≥ 1024</td>
<td>accept</td>
</tr>
<tr>
<td>ICMP</td>
<td>any</td>
<td>any</td>
<td>any</td>
<td>any</td>
<td>accept</td>
</tr>
<tr>
<td>IP</td>
<td>any</td>
<td>any</td>
<td>any</td>
<td>any</td>
<td>deny</td>
</tr>
</tbody>
</table>

---

Unit 2: IP Networks

Security in IP – VPN Security

Authentication
Cryptography
Tunneling

Example: creating a tunnel in Linux:
`ip tunnel add tun0 mode gre remote 180.0.0.30 local 180.0.0.25 ttl 256`

<table>
<thead>
<tr>
<th>Destination</th>
<th>Gateway</th>
<th>Gateway</th>
<th>Destination</th>
<th>Gateway</th>
<th>Gateway</th>
<th>Destination</th>
<th>Gateway</th>
<th>Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.1.0</td>
<td>0.0.0.0</td>
<td>0.0.0.0</td>
<td>10.0.1.0</td>
<td>0.0.0.0</td>
<td>0.0.0.0</td>
<td>10.0.1.0</td>
<td>0.0.0.0</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>160.0.0.1</td>
<td>0.0.0.0</td>
<td>0.0.0.0</td>
<td>255.255.255</td>
<td>0.0.0.0</td>
<td>0.0.0.0</td>
<td>255.255.255</td>
<td>0.0.0.0</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>0.0.0.0</td>
<td>192.168.0.1</td>
<td>160.0.0.1</td>
<td>192.168.0.0</td>
<td>160.0.0.1</td>
<td>192.168.0.0</td>
<td>192.168.0.0</td>
<td>192.168.0.0</td>
<td>192.168.0.0</td>
</tr>
<tr>
<td>10.0.1.0</td>
<td>192.168.0.1</td>
<td>255.255.255</td>
<td>10.0.1.0</td>
<td>192.168.0.1</td>
<td>255.255.255</td>
<td>10.0.1.0</td>
<td>192.168.0.1</td>
<td>255.255.255</td>
</tr>
</tbody>
</table>

---

Unit 2: IP Networks

Security in IP – Virtual Private Network, VPN

Provides connectivity for remote users over a public infrastructure, as they would have over a private network.

- Conventional Private Network
  - More cost.
  - Less flexible.
  - WAN management.

- VPN
  - Less cost.
  - More flexible.
  - Simple management.
  - Internet availability.
Unit 2: IP Networks

Security in IP – VPN Tunneling Problems

Fragmentation inside the tunnel will use the external header, thus, the exit router of the tunnel may reassemble fragmented datagrams.

ICMP messages sent inside the tunnel are addressed to the tunnel entry.

MTU path discovery may fail.

Solution: the router entry maintains a “tunnel state”, e.g. the tunnel MTU, and generate ICMP messages that would be generated inside the tunnel. Furthermore, the tunnel entry router typically fragment the datagrams, if needed, before encapsulation, to avoid the exit router having to reassemble fragmented datagrams.

![Diagram of tunneling](image)


Generic Routing Encapsulation, GRE (RFC 1701): There is an additional GRE header: different protocol encap (not only IP).

Point-to-Point Tunneling Protocol (RFC 2637): Add the ppp functionalities.

IPsec (RFC 2401): Standards to introduce authentication and encryption and tunneling to IP layer.