Computer Networking

IP

Andrzej Duda
Fabrice Theoleyre

Network Layer

Chapter goals:

- understand principles behind network layer services:
  - ✓ addressing
  - ✓ packet forwarding
  - ✓ routing (how a router works)
- instantiation and implementation in the Internet

Overview:

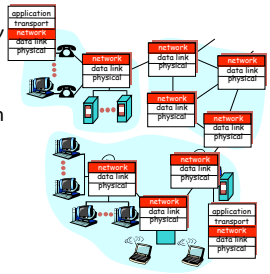
- network layer services
- IP addresses
- packet forwarding principles
- details of IP
- overview of DHCP, ICMP, ARP
- routing protocols
- routers
- IPsec and VPN

Network layer functions

- transport packet from sending to receiving hosts
- network layer protocols in *every* host, router

three important functions:

- *path determination:* route taken by packets from source to dest. *Routing algorithms*
- *switching:* move packets from router's input to appropriate router output
- *call setup:* some network architectures require router call setup along path before data flows



Network service model

- The *network service model* defines edge-to-edge channel
- The most important abstraction provided by network layer:
  - ✓ **network-layer connection-oriented service:** virtual circuit (X. 25, Frame Relay, ATM, MPLS)
  - ✓ **network-layer connectionless service**: datagram (IP, IPX)

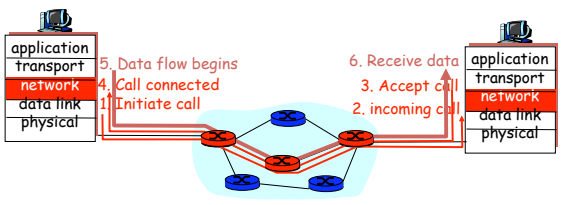
Virtual circuits

"source-to-dest path behaves much like telephone circuit"
- ✓ performance-wise
- ✓ network actions along source-to-dest path

- call setup, teardown for each call *before* data can flow
- each packet carries VC identifier (not destination host ID)
- *every* router on source-dest path maintains "state" for each passing connection
  - ✓ transport-layer connection only involved two end systems
- link, router resources (bandwidth, buffers) may be *allocated* to VC
  - ✓ to get circuit-like performance

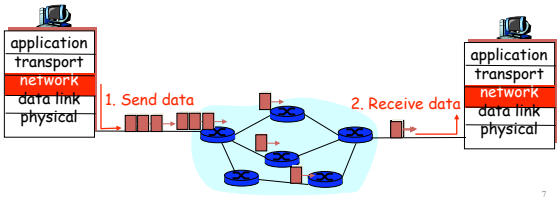
Virtual circuits: signaling protocols

- used to setup, maintain teardown VC
- used in ATM, Frame-Relay, X.25
- not used in today's Internet
  - ✓ but MPLS at 2.5 (between Link and Network Layer)



Datagram networks: the Internet model

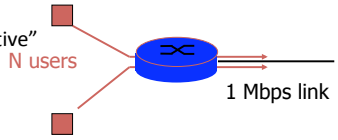
- no call setup at network layer
- routers: no state about end-to-end connections
  - ✓ no network-level concept of "connection"
- packets typically routed using destination host ID
  - ✓ packets between same source-dest pair may take different paths



Packet switching vs. Circuit switching

Packet switching allows more users to use network!

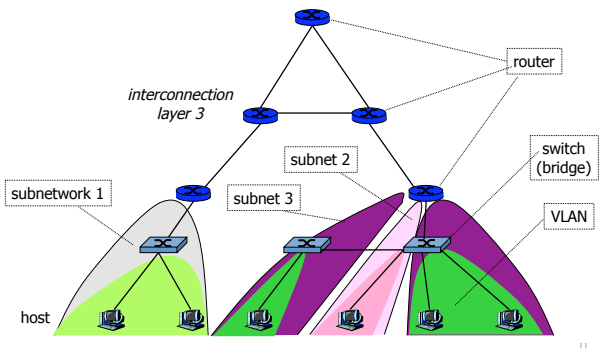
- Eg. 1 Mbit link
- each user:
  - ✓ 100 Kb/s when "active"
  - ✓ active 10% of time
- circuit-switching:
  - ✓ 10 users
- packet switching:
  - ✓ with 35 users, probability > 10 active less that .004



IP principles

- Elements
  - ✓ host = end system; router = intermediate system; subnetwork = a collection of hosts that can communicate directly without routers
- Routers are between subnetworks only:
  - ✓ a subnetwork = a collection of systems with a common prefix
- Packet forwarding
  - ✓ direct: inside a subnetwork hosts communicate directly without routers, router delivers packets to hosts
  - ✓ indirect: between subnetworks one or several routers are used
- Host either sends a packet to the destination using its LAN, or it passes it to the router for forwarding

Interconnection structure - layer 3



Interconnection at layer 3

- Routers
  - ✓ interconnect subnetworks
  - ✓ logically separate groups of hosts
  - ✓ managed by one entity
- Forwarding based on IP address
  - ✓ structured address space
  - ✓ routing tables: aggregation of entries
  - ✓ works if no loops - routing protocols
  - ✓ scalable inside one administrative domain

Internet and intranet

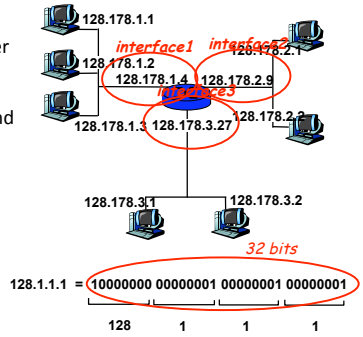
- An intranet
  - ✓ a collection of end and intermediate systems interconnected using the TCP/IP architecture
  - ✓ normally inside one organization
- The Internet
  - ✓ the global collection of all hosts and routers interconnected using the TCP/IP architecture
  - ✓ coordinated allocation of addresses and implementation requirements by the Internet Society
- Intranets are often connected to the Internet by firewalls
  - ✓ routers that act as protocol gateways (address and port translation, application level relay)

IP addresses

- Unique addresses in the world, decentralized allocation
- An IP address is 32 bits, noted in dotted decimal notation: **`192.78.32.2`**
- An IP address has a prefix and a host part:
  - ✓ **`prefix:host`**
- Two ways of specifying prefix
  - ✓ subnet mask identifies the prefix by bitwise & operation
  - ✓ CIDR: bit length of the prefix
- Prefix identifies a subnetwork
  - ✓ used for locating a subnetwork - routing

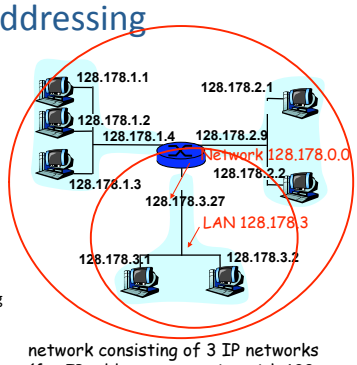
IP Addressing: introduction

- IP address: 32-bit identifier for host, router *interface*
- *interface:* connection between host, router and physical link
  - ✓ router's typically have multiple interfaces
  - ✓ host may have multiple interfaces
  - ✓ IP addresses associated with interface, not host, router



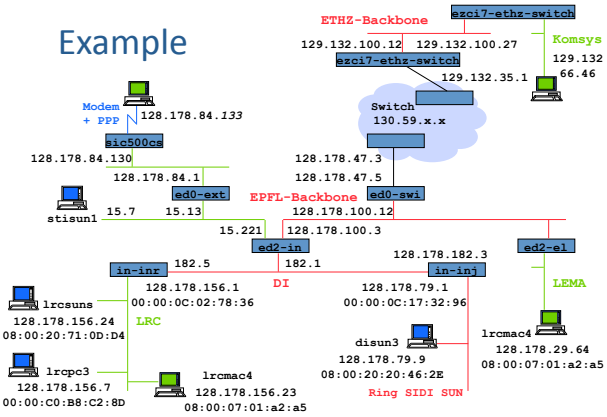
IP Addressing

- IP address:
  - ✓ network (or prefix) part (high order bits)
  - ✓ host part (low order bits)
- *What's a subnetwork?* (from IP address perspective)
  - ✓ device interfaces with same network part of IP address
  - ✓ can physically reach each other without intervening router

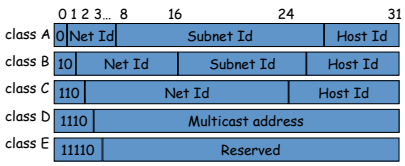


network consisting of 3 IP networks (for IP addresses starting with 128, first 24 bits are network address)

Example



IP Address Classes



Examples: 128.178.x.x = EPFL host; 129.132.x.x = ETHZ host
9.x.x.x = IBM host 18.x.x.x = MIT host

Class	*Range*
A	0.0.0.0 to 127.255.255.255
B	128.0.0.0 to 191.255.255.255
C	192.0.0.0 to 223.255.255.255
D	224.0.0.0 to 239.255.255.255
E	240.0.0.0 to 247.255.255.255

- Class B addresses are close to exhausted; new addresses are taken from class C, allocated as continuous blocks

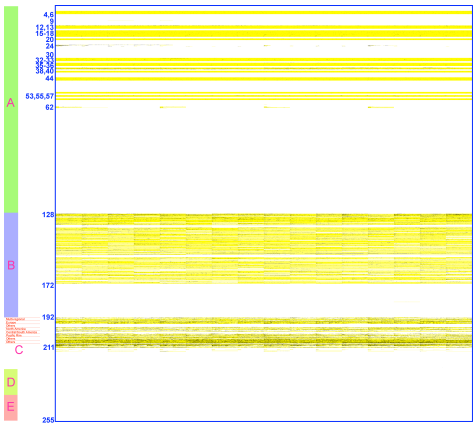
Special case IP addresses

```
1. 0.0.0.0                  this host, on this network
2. 0.hostId                 specified host on this net
                                (initialization phase)
3. 255.255.255.255          limited broadcast
                                (not forwarded by routers)
4. subnetId.all 1's         broadcast on this subnet
5. subnetId.all 0's         BSD used it for broadcast
                            on this subnet (obsolete)
6. 127.x.x.x                    loopback

7. 10/8                     reserved networks for
   172.16/12                    internal use (Intranet)
   192.168/16
```

- 1,2: source IP@ only; 3,4,5: destination IP@ only

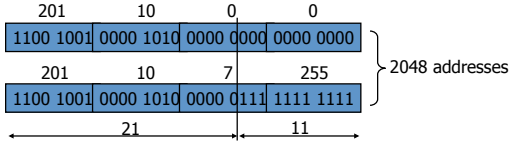
Used addresses in Internet



CIDR: IP Address Hierarchies

- The prefix of an IP address is itself structured in order to support aggregation
  - ✓ For example: 128.178.x.y represents an EPFL host
    128.178.156 / 24 represents the LRC subnet at EPFL
    128.178/15 represents EPFL
  - ✓ Used between routers by routing algorithms
  - ✓ This way of doing is called classless and was first introduced in inter domain routing under the name of CIDR (Classless Interdomain Routing)
- Notation: 128.178.0.0/16 means : the prefix made of the 16 first bits of the string
- It is equivalent to: 128.178.0.0 with netmask=255.255.0.0
- In the past, the class based addresses, with networks of class A, B or C was used; now only the distinction between class D and non-class D is relevant.

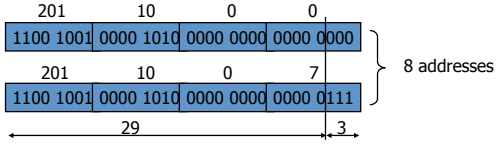
CIDR



201.10.0.0/21: 201.10.0.0 - 201.10.0.255
201.10.1.0 - 201.10.1.255
...
201.10.7.0 - 201.10.7.255

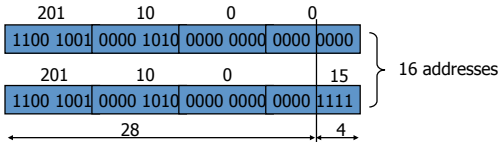
1 C class network: 256 addresses
256 × 8 = 2048 addresses

Choosing prefix length



- prefix = 201.10.0.0/29
  - ✓ 8 addresses
  - ✓ 1 broadcast address: 201.10.0.7
  - ✓ 1 network address: 201.10.0.0
  - ✓ only 6 addresses can be used for hosts

Choosing prefix length

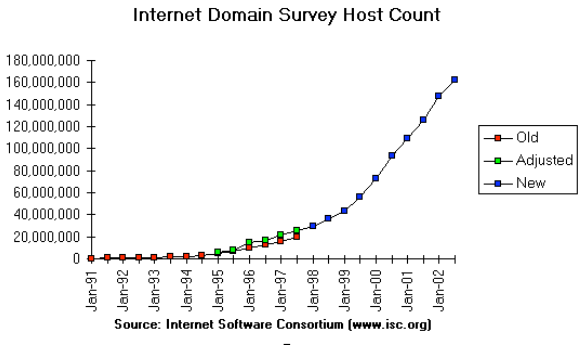


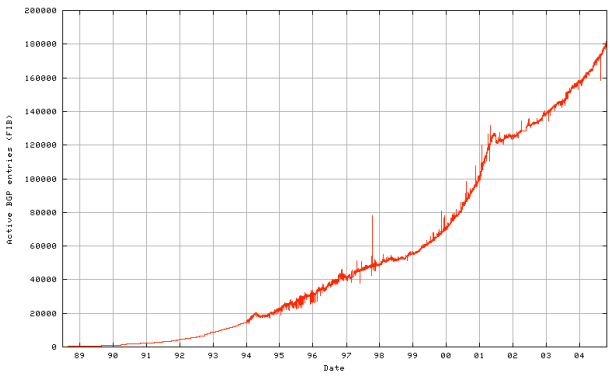
- prefix = 201.10.0.0/28
  - ✓ 201.10.0.16/28, 201.10.0.32/28, 201.10.0.48/28…
  - ✓ 16 addresses
  - ✓ 2 broadcast addresses: 201.10.0.0, 201.10.0.15
  - ✓ only 14 addresses can be used for hosts

Address allocation

- World coverage
  - ✓ Europe and the Middle East (RIPE NCC)
  - ✓ Africa (ARIN & RIPE NCC)
  - ✓ North America (ARIN)
  - ✓ Latin America including the Caribbean (ARIN)
  - ✓ Asia-Pacific (APNIC)
- Current allocations of Class C
  - ✓ `193-195/8, 212-213/8, 217/8` for RIPE
  - ✓ `199-201/8, 204-209/8, 216/8` for ARIN
  - ✓ `202-203/8, 210-211/8, 218/8` for APNIC
- Simplifies routing
  - ✓ short prefix aggregates many subnetworks
  - ✓ routing decision is taken based on the short prefix

Number of hosts

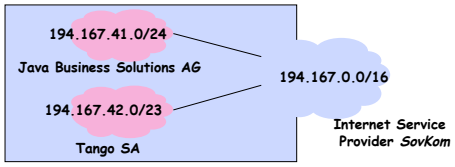




IP Addresses and subnet mask

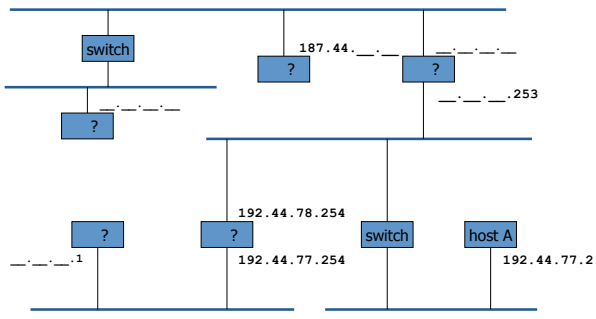
- subnet mask at ETHZ = 255.255.255.192
- CIDR 129.132/26
- question: subnet prefix and host parts of spr13.tik.ee.ethz.ch = 129.132.119.77 ?
  - ✓ 129.132.119.77 : 10000001.10000100.01110111.01001101
  - ✓ 255.255.255.192: 11111111.11111111.11111111.11000000
- answer:
  - ✓ subnet prefix = 129.132.119.64 (64=01000000)
  - ✓ host = 13=001101 (6 bits)

IP Addresses



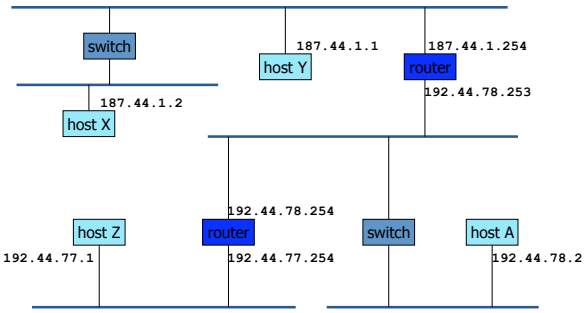
- Sovkom
  - ✓ has received IP addresses **194.167.0.0** to **194.167.255.255**
  - ✓ total: 2^{16} addr., but .0 and .255 are not usable
- **Java Business Solutions AG**
  - ✓ has received IP addresses **194.167.41.0** to **194.167.41.255**
  - ✓ total: 2^8 –2 addresses
- **Tango SA**
  - ✓ has received IP addresses **194.167.43.255** to **194.167.42.0**
  - ✓ total: 2^9 –2 addresses

Example



- Can host A have this address?

Example



- Host A is on subnetwork `192.44.78`

IP Principles

- Homogeneous addressing
  - ✓ an IP address is unique across the whole network (= the world in general)
  - ✓ IP address is the address of the interface
  - ✓ communication between IP hosts requires knowledge of IP addresses
- Routing:
  - ✓ inside a subnetwork: hosts communicate directly without routers
  - ✓ between subnetworks: one or several routers are used
  - ✓ a subnetwork = a collection of systems with a common prefix

IP packet forwarding algorithm

- Rule for sending packets (hosts, routers)
  - ✓ if the destination IP address has the same prefix as one of my interfaces, send directly to that interface
  - ✓ otherwise send to a router as given by the IP routing table

At `lrcsuns`: Next Hop Table

destination@	subnetMask	nextHop
DEFAULT		128.178.156.1

Physical Interface Tables

IP	subnetMask
128.178.156.24	255.255.255.0

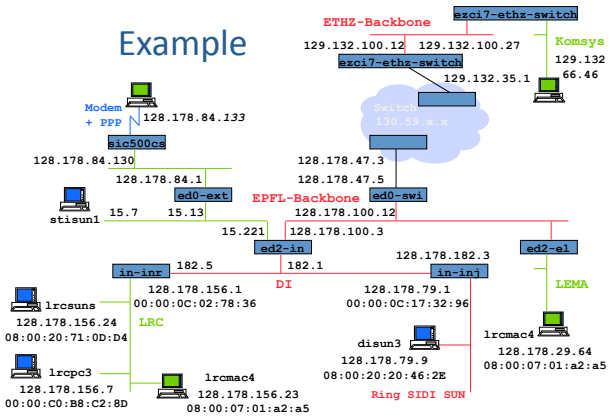
At `in-inj`: Next Hop Table

destination@	subnetMask	nextHop
128.178.156.0 DEFAULT	255.255.255.0	128.178.182.5 128.178.182.1

Physical Interface Tables

IP	subnetMask
128.178.79.1 128.178.182.3	255.255.255.0 255.255.255.0

Example



IP packet forwarding algorithm

destAddr = packet dest. address, destinationAddr = address in routing table

Case 1: a **host route** exists for destAddr
  for every entry in routing table
    if (destinationAddr = destAddr)
    then send to nextHop IPaddr; leave

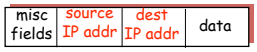
Case 2: destAddr is on a **directly connected network** (= on-link):
  for every physical interface IP address A and subnet mask SM
    if(A & SM = destAddr & SM)
    then send directly to destAddr; leave

Case 3: a **network route** exists for destAddr
  for every entry in routing table and subnet mask SM
    if (destinationAddr & SM = destAddr & SM)
    then send to nextHop IP addr; leave

Case 4: use **default route**
  for every entry in routing table
    if (destinationAddr=DEFAULT) then send to nextHop IPaddr; leave

Getting a datagram from source to dest.

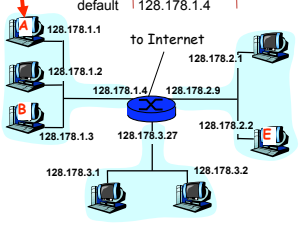
IP datagram:



- datagram remains unchanged, as it travels source to destination
- addr fields of interest here

routing table in A

Dest. Net.	next router	Nhops
128.178.1		1
128.178.2	128.178.1.4	2
128.178.3	128.178.1.4	2
default	128.178.1.4	

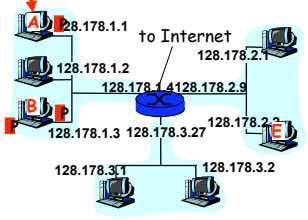


Getting a datagram from source to dest.: same subnetwork

misc fields	128.178.1.1	128.178.1.3	data

- Starting at A, given IP datagram addressed to B:
- look up net. address of B
- find B is on same net. as A
- link layer will send datagram directly to B inside link-layer frame
  - ✓ B and A are directly connected

Dest. Net.	next router	Nhops
128.178.1		1
128.178.2	128.178.1.4	2
128.178.3	128.178.1.4	2
default	128.178.1.4	



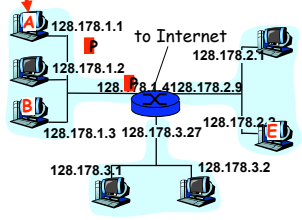
Getting a datagram from source to dest.: different subnetworks

misc fields	128.178.1.1	128.178.2.3	data

Starting at A, dest. E:

- look up network address of E
- E on *different* network
  - A, E not directly attached
- routing table: next hop router to E is 128.178.1.4
- link layer sends datagram to router 128.178.1.4 inside link -layer frame
- datagram arrives at 128.178.1.4
- continued.....

Dest. Net.	next router	Nhops
128.178.1		1
128.178.2	128.178.1.4	2
128.178.3	128.178.1.4	2
default	128.178.1.4	



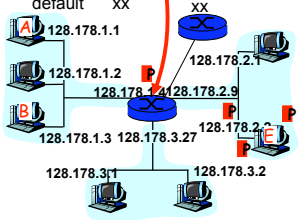
Getting a datagram from source to dest.: different subnetworks

misc fields	128.178.1.1	128.178.2.3	data

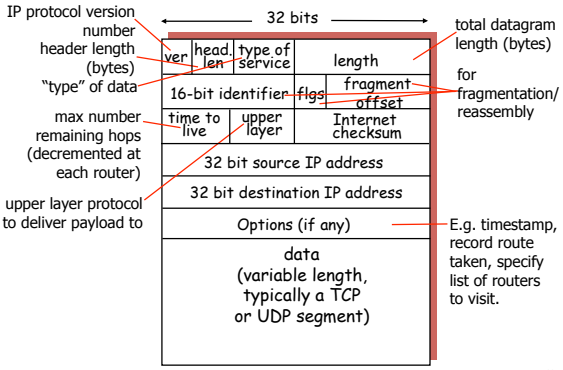
Arriving at 128.178.1.4, destined for 128.178.2.2

- look up network address of E
- E on *same* network as router's interface 128.178.2.9
  - router, E directly attached
- link layer sends datagram to 128.178.2.2 inside link-layer frame via interface 128.178.2.9
- datagram arrives at 128.178.2.2!!! (hooray!)

Dest. network	next router	Nhops	interface
128.178.1	-	1	128.178.1.4
128.178.2	-	1	128.178.2.9
128.178.3	-	1	128.178.3.27
default	xx	xx	



IP datagram format



IP header

- Version
  - ✓ IPv4, future IPv6
- Header size
  - ✓ options - variable size
  - ✓ in 32 bit words
- Type of service
  - ✓ priority : 0 - normal, 7 - control packets
  - ✓ short delay (telnet), high throughput (ftp), high reliability (SNMP), low cost (NNTP)
- Redefined in DiffServ (Differentiated Services)
  - ✓ 1 byte codepoint determining QoS class
    - Expedited Forwarding (EF) - minimize delay and jitter
    - Assured Forwarding (AF) - four classes and three drop-precedences (12 codepoints)

IP header

- Packet size
  - ✓ in bytes including header
  - ✓ <= 64 Kbytes; limited in practice by link-level MTU (Maximum Transmission Unit)
- Id
  - ✓ unique identifier for re-assembling
- Flags
  - ✓ M : more ; set in fragments
  - ✓ F : prohibits fragmentation

IP header

- Offset
  - ✓ position of a fragment in multiples of 8 bytes
- TTL (Time-to-live)
  - ✓ in seconds
  - ✓ now: number of hops
  - ✓ router : --, if 0, drop (send ICMP packet to source)
- Protocol
  - ✓ identifier of protocol (1 - ICMP, 6 - TCP, 17 - UDP)
- Checksum
  - ✓ only on the header
  - ✓ Recomputed per hop

IP header

- Options
  - ✓ strict source routing
    - all routers
  - ✓ loose source routing
    - some routers
  - ✓ record route
  - ✓ timestamp route
  - ✓ router alert
    - used by IGMP or RSVP for processing a packet

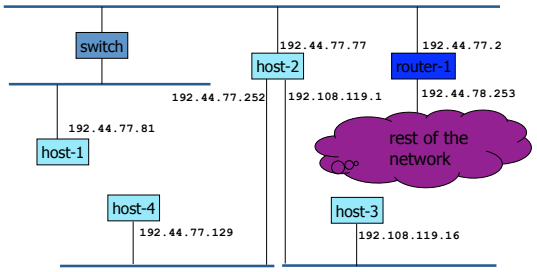
Configuration of a Unix host

```
/usr/etc/ifconfig interface [ address_family ]
        [ address [ dest_address ] ] [ netmask mask ]
        [ broadcast address ] [ up ] [ down ] [ trailers ]
        [ -trailers ] [ arp ] [ -arp ] [ private ]
        [ -private ] [ metric n ] [ auto-revarp ]

host-1# ifconfig le0 192.44.77.81 255.255.255.128

host-1# ifconfig -a
le0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING>
    inet 192.44.77.81 netmask ffffff80 broadcast 192.44.77.0
    ether 8:0:20:1c:74:84
lo0: flags=849<UP,LOOPBACK,RUNNING>
    inet 127.0.0.1 netmask ff000000
```


Example interconnection



Routing tables

```
host-1 (192.44.77.81) :
>netstat -n -r
Routing tables
Destination      Gateway       Flags    Refcnt Use   Interface
192.108.119.16   192.44.77.77  UGHD     1      1683  le0
127.0.0.1        127.0.0.1     UH       2      12971 lo0
default          192.44.77.2   UG       3      16977 le0
192.44.77.0      192.44.77.81  U        13     5780  le0

U - up
G - gateway (next router)
H - host route
D - route from ICMP Redirect
```


Routing tables

```
host-2 (192.44.77.77) :
>rsh host-2 netstat -n -r
Routing tables
Destination    Gateway        Flags  Refcnt  Use      Interface
127.0.0.1      127.0.0.1      UH     3       351344   lo0
default        192.44.77.2    UG     3       17388997 le0
192.44.77.128  192.44.77.252  U      26      504768   le2
192.44.77.0    192.44.77.77   U      24      10702069 le0
192.108.119.0  192.108.119.1  U      2       249777   le1
```

Modifying routing tables

```
/usr/etc/route [ -fn ] add|delete [ host|net ]
  destination [gateway [ metric ] ]
host-1# netstat -r
Routing tables
Destination     Gateway      Flags    Refcnt Use
  Interface
localhost       localhost    UH       2      13569    lo0
192.44.77.0     host-1       U        18     13272    le0
host-1# ping 133.11.11.11
sendto: Network is unreachable
host-1# route add 0.0.0.0 router-1 1
add net 0.0.0.0 gateway router-1
```


Modifying routing tables

```
host-1# netstat -r
Routing tables
Destination     Gateway      Flags    Refcnt Use
  Interface
localhost       localhost    UH       2      13591    lo0
default         router-1     UG       0      0        le0
192.44.77.0     host-1       U        16     13566    le0
host-1# ping 133.11.11.11
133.11.11.11 is alive
```

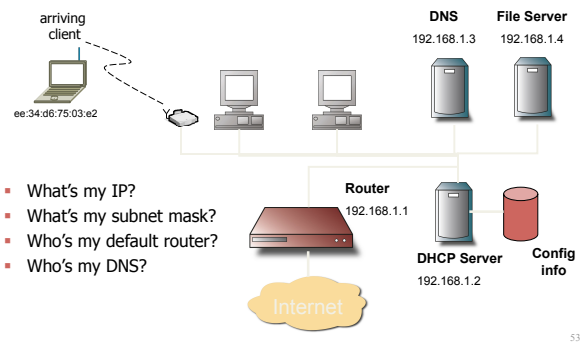

DHCP

- DHCP
  - ✓ Dynamic Host Configuration Protocol (RFC 2131)
- Goal: allow host to dynamically obtain its IP address from network server when it joins network
  - ✓ Support for mobile users who want to join network
  - ✓ Allows reuse of addresses (hold address only while connected)
- Uses UDP port 67 (to server) and 68 (to client)
  - ✓ Why UDP?
  - ✓ IP source address 0, broadcast 255.255.255.255

DHCP

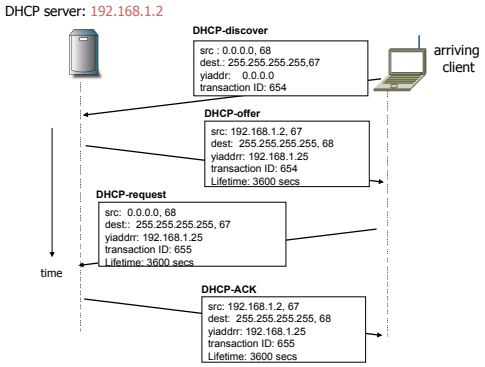
- Dynamic addresses
  - ✓ 2 databases
    - Static DB - Matches IP's and Physical Addresses
    - Dynamic DB - Pool of IP's leased out
- Temporary addresses
  - ✓ Addresses leased from Dynamic DB are temporary
  - ✓ Each lease has an expiration which the client must obey
  - ✓ Can renew its lease on address in use

DHCP



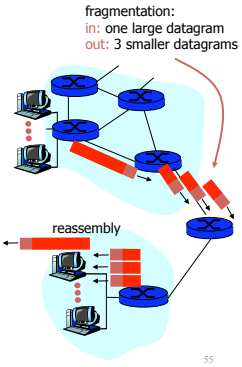
- What's my IP?
- What's my subnet mask?
- Who's my default router?
- Who's my DNS?

DHCP client-server scenario



IP Fragmentation & Reassembly

- network links have MTU (max.transfer size) - largest possible link-level frame.
  - ✓ different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - ✓ one datagram becomes several datagrams
  - ✓ "reassembled" only at final destination
  - ✓ IP header bits used to identify, order related fragments
- fragmentation is in principle avoided with TCP and UDP using small segments



MTU: Maximum Transfer Unit

Data links have different maximum packet length

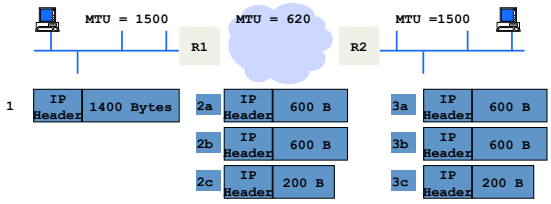
- MTU (maximum transmission unit) = maximum packet size usable for an IP packet
- value of short MTU ? of long MTU ?

Link technology	MTU
Ethernet	1500
802.3 with LLC/SNAP	1492
FDDI	4352
X.25	576
Frame Relay	1600
ATM with AAL5	9180
Hyperchannel	65535
PPP	296 to 1500

```
lrcsuns$ ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
        inet 127.0.0.1 netmask ff000000
le0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
        inet 128.178.156.24 netmask ffffff00 broadcast 128.178.156.255
        ether 8:0:20:71:d:d4
```


IP fragmentation

- IP hosts or routers may have IP datagrams larger than MTU
- fragmentation is performed when IP datagram too large
- re-assembly is only at destination
- fragmentation is in principle avoided with TCP



IP fragmentation

- IP datagram is *fragmented* if MTU of interface < datagram total length
- all fragments are self-contained IP packets
- fragmentation controlled by fields: Identification, Flag and Fragment Offset
- IP *datagram* = original ; IP *packet* = fragments or complete datagram

	1	2a	2b	2c
Length	1420	620	620	220
Identification	567	567	567	567
More Fragment flag	0	1	1	0
Offset	0	0	75	150
8 * Offset	0	0	600	1200

```
Fragment data size (here  600) is always a multiple of 8
Identification given by source
```


TCP, UDP and fragmentation

- The UDP service interface accepts a datagram up to 64 KB
  - ✓ UDP datagram passed to the IP service interface as one SDU
  - ✓ is fragmented at the source if resulting IP datagram is too large
- The TCP service interface is stream oriented
  - ✓ packetization is done by TCP
  - ✓ several calls to the TCP service interface may be grouped into one TCP segment (many small pieces)
  - ✓ or: one call may cause several segments to be created (one large piece)
  - ✓ TCP always creates a segment that fits in one IP packet: no fragmentation at source
  - ✓ fragmentation may occur in a router, if IPv4 is used, and if PMTU discovery is not implemented

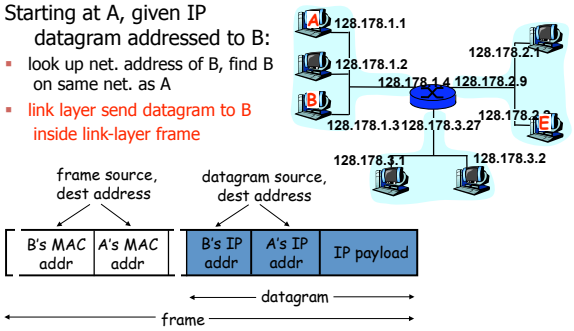
LAN Addresses and ARP

- 32-bit IP address:
  - ✓ network-layer address
  - ✓ used to get datagram to destination network (recall IP network definition)
- LAN (or MAC or physical) address:
  - ✓ used to get datagram from one interface to another physically-connected interface (same network)
  - ✓ 48 bit MAC address (for most LANs) burned in the adapter ROM
- Why different addresses at IP and MAC?
  - ✓ LANs not only for IP (LAN addresses are neutral)
  - ✓ if IP addresses used, they should be stored in a RAM and reconfigured when host moves
  - ✓ independency of layers

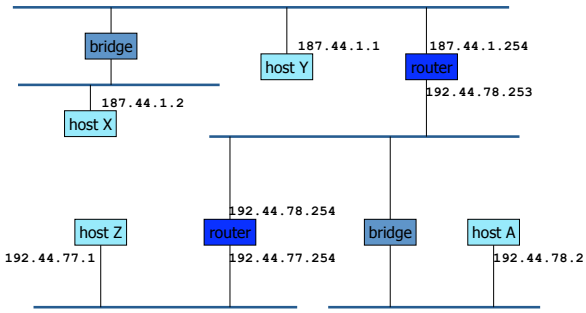
MAC Address resolution

Starting at A, given IP datagram addressed to B:

- look up net. address of B, find B on same net. as A
- link layer send datagram to B inside link-layer frame



Example



- Host A is on subnetwork `192.44.78`

Packet delivery

Packet sent by `187.44.1.2 to 187.44.1.1`

MAC-host-Y	MAC-host-X	187.44.1.1	187.44.1.2	payload
Ethernet header		IP header		

X needs to know MAC address of Y (ARP)

Packet sent by `187.44.1.2 to 192.44.78.2`

MAC-router	MAC-host-X	192.44.78.2	187.44.1.2	payload
Ethernet header		IP header		

MAC-host-A	MAC-router	192.44.78.2	187.44.1.2	payload
Ethernet header		IP header		

X needs to know MAC address of router (X knows the IP address of router - configuration)

Router needs to know MAC address of A

ARP: Address Resolution Protocol

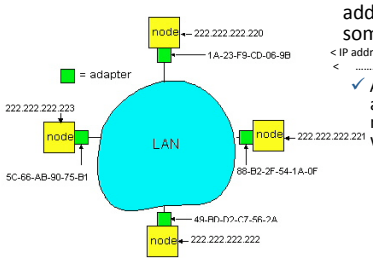
ARP is used to determine the MAC address of B given B's IP address

- Each IP node (Host, Router) on LAN implements ARP protocol and has ARP table
- ARP Table: IP/MAC address mappings for some LAN nodes

< IP address; MAC address>

< ..... >

  - ✓ ARP table is a cache: after an interval (typically 20 min) the address mapping will be forgotten

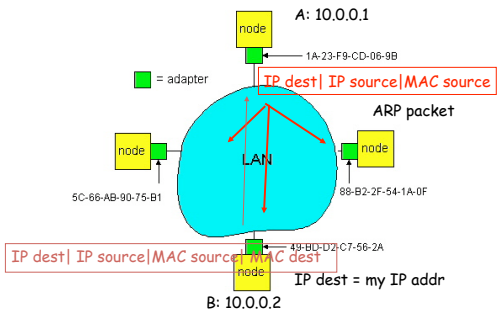


ARP protocol

- A knows B's IP address, wants to learn physical address of B
- A broadcasts ARP query pkt, containing B's IP address
  - ✓ all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) physical layer address
- A caches (saves) IP-to-physical address pairs until information becomes old (times out)
  - ✓ soft state: information that times out (goes away) unless refreshed

ARP protocol

IP address MAC address TTL
10.0.0.2 49:BD:D2:07:56:2A 6:00:00



ARP frame

- Request (broadcast)

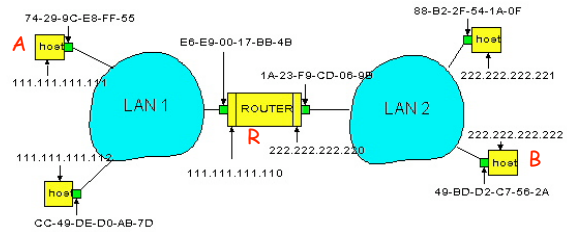
sender Ethernet address
sender IP address
target Ethernet address ???
target IP address

- Reply (unicast)

sender Ethernet address
sender IP address
target Ethernet address
target IP address

Routing to another LAN

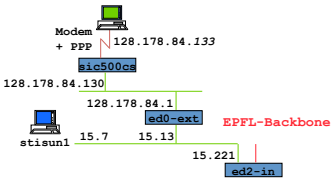
- walkthrough: routing from A to B via R



- In routing table at source Host, find router 111.111.111.110
- In ARP table at source, find MAC address E6-E9-00-17-BB-4B, etc

Proxy ARP

- Proxy ARP: a host answers ARP requests on behalf of others
  - ✓ example: **`sic500cs`** for PPP connected computers
  - ✓ manual configuration of **`sic500cs`**



ICMP: Internet Control Message Protocol

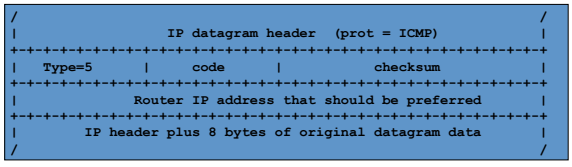
- Used by hosts, routers, gateways to communication network-level information
  - ✓ error reporting: unreachable host, network, port, protocol
  - ✓ echo request/reply (used by ping)
- Network-layer "above" IP:
  - ✓ ICMP msgs carried in IP datagrams
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

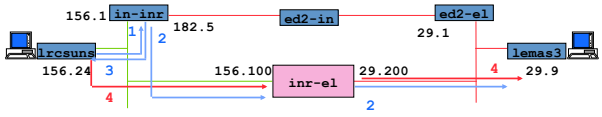
ICMP Redirect

- Sent by router to source host to inform source that destination is directly connected
  - ✓ host updates the routing table
  - ✓ ICMP redirect can be used to update the router table (eg. **`in-inj`** route to LRC?)

ICMP Redirect Format



ICMP Redirect example



	dest IP addr	srce IP addr	prot	data part
1:	128.178.29.9	128.178.156.24	udp	xxxxxxx
2:	128.178.29.9	128.178.156.24	udp	xxxxxxx
3:	128.178.156.24	128.178.156.1 cksum	icmp	type=redir code=host 128.178.156.100 xxxxxxx (28 bytes of
4:	128.178.29.9	128.178.156.24	udp	.....

ICMP Redirect example (cont'd)

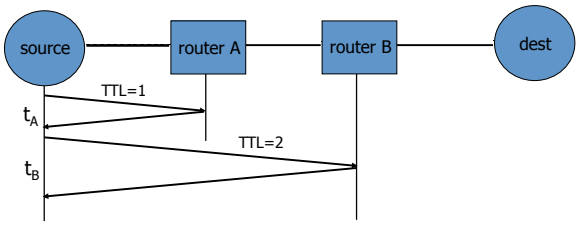
After 4

```
lrcsuns$ netstat -nr
Routing Table:
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
127.0.0.1            127.0.0.1            UH        0  11239  lo0
128.178.29.9         128.178.156.100      UGHD      0     19
128.178.156.0        128.178.156.24       U         3  38896  le0
224.0.0.0            128.178.156.24       U         3      0  le0
default              128.178.156.1        UG        0  85883
```


Tools that use ICMP

- ping
  - ICMP Echo request
  - wait for Echo reply
  - measure RTT
- traceroute
  - IP packet with TTL = 1
  - wait for ICMP TTL expired
  - IP packet with TTL = 2
  - wait for ICMP TTL expired
  - ...

Traceroute



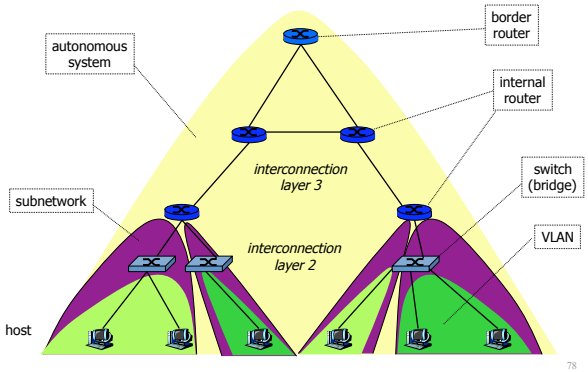
Routing and Packet forwarding

- Packet forwarding - data plane
  - forward every packet to the next hop
  - in real time
- Routing - control plane
  - computation of routing tables or data structures for unicast and multicast
  - normally only between routers
  - non-real time: latency up to several minutes
  - two level hierarchy
    - internal routing inside an administrative domain (called autonomous system - AS)
    - external routing between AS (administrative domains or ISPs)
  - uses routing protocols such as
    - internal: RIP, OSPF, EIGRP
    - external: BGP

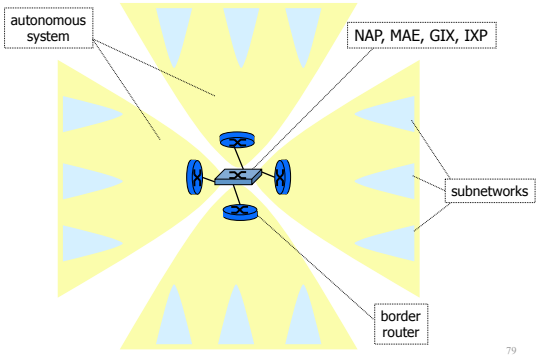
Routing Table maintenance

- at host
  - configuration
  - ICMP redirect
  - ICMP router discovery messages
- at routers
  - configuration
    - static routing table
  - routing protocols between routers:
    - exchange topology and addressing information
    - build routing table

Autonomous systems



Interconnection of AS



Interconnection of AS

- Border routers
  - ✓ interconnect AS
- NAP or GIX, or IXP
  - ✓ exchange of traffic - peering
- Route construction
  - ✓ based on the path through a series of AS
  - ✓ based on administrative policies
  - ✓ routing tables: aggregation of entries
  - ✓ works if no loops and at least one route - external routing protocols, e.g. BGP

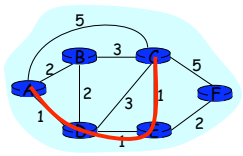
Routing

Routing protocol

Goal: determine "good" path (sequence of routers) thru network from source to dest.

Graph abstraction for routing algorithms:

- graph nodes are routers
- graph edges are physical links
  - ✓ link cost: delay, $ cost, or congestion level

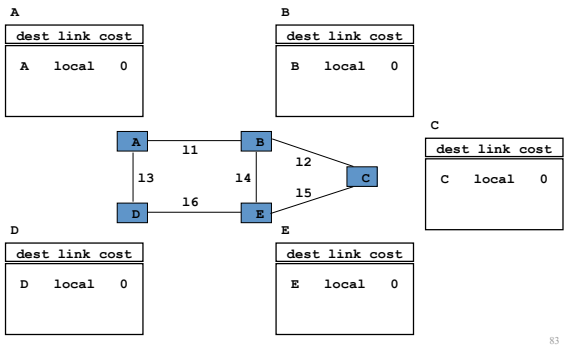


- "good" path:
  - typically means minimum cost path
  - other def's possible

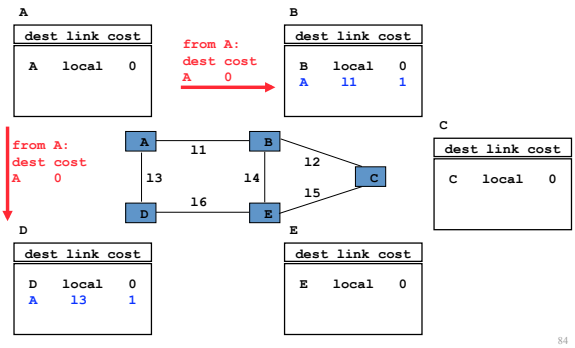
Distance vector example

- Simple network
  - ✓ routers connected by links
  - ✓ destinations = subnetworks connected to routers
  - ✓ symmetric links
  - ✓ cost = number of hops
- RIP (Routing Information Protocol)

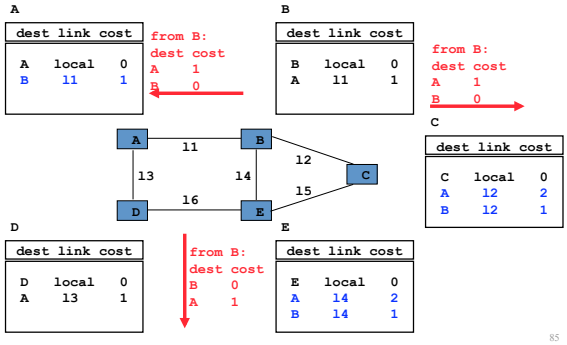
Initialization



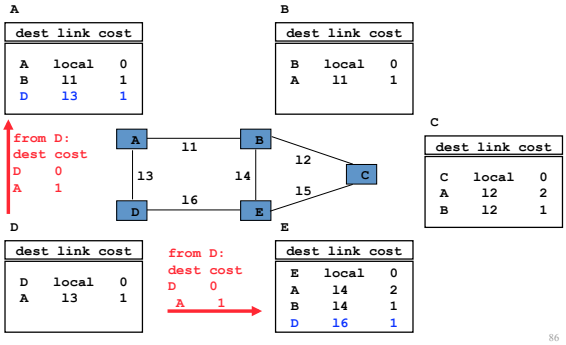
Distance vector announcement



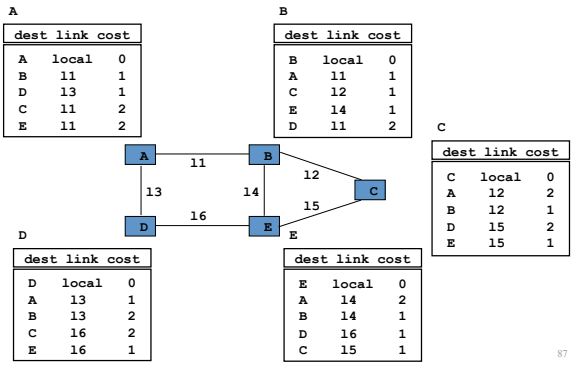
Distance vector announcement



Distance vector announcement



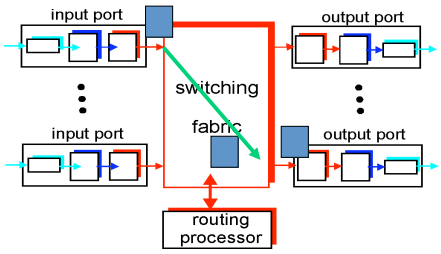
Final



Router Architecture Overview

Two key router functions:

- run *routing* algorithms/protocol (RIP, OSPF, BGP)
- *forward* datagrams from incoming to outgoing link



IPSec

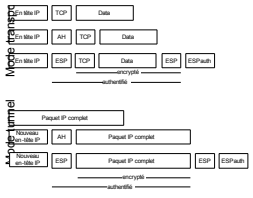
- IPSec
  - ✓ In IP
  - ✓ Mandatory in IPv6, facultative in IPv4
  - ✓ Must be set up in any final clients
    - End to end security
    - Creates a secured virtual link
- Services
  - ✓ Confidentiality
  - ✓ Authentication
  - ✓ Replay

IPSec

- Properties
  - ✓ Based on Diffie Hellman
  - ✓ Symetric and Asymetric Cryptography
- AH (Authentification Header)
  - ✓ Another field in the IP packet
  - ✓ Uses the id field against the replay
- ESP (Encapsulating Security Payload)
  - ✓ The packet is encyphered
    - Encapsulation if confidentiality of the headers
- Completely independent of algorithms
  - ✓ DES, RC5, Blowfish, HMAC-MD5, HMAC-SHA-1

IPSec

- Modes
  - ✓ Transport: transforms the payload
    - Next protocol : AH or ESP
    - Uniquely for terminal clients
  - ✓ Tunnel: encapsulates the whole packet
    - Any network device



IPSec

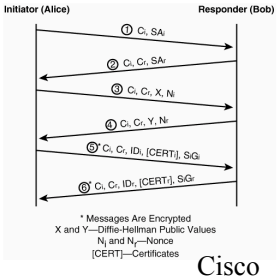
- Security parameters
  - ✓ Manual
    - Configuration of each device, and each association
    - Heavy …
  - ✓ Auto
    - Dynamical protocol
    - Negocation, update, and deletion of all parameters
    - How does it work?
      - Mutual authentication
        - » Shared secret or asymmetric cryptography
        - » Uses the X509 certificates with eventual Authority of Certification
      - Creates a dynamical shared secret with Diffie-Hellman

IPSec

- Security Association (SA)
  - Identified by destination / serial number / protocol (AH, ESP)
  - A data structure which contains all the parameters
    - keys, duration of the keys, algorithms, hosts …
    - Unidirectional: one flow ⇢ 2 SA
  - ISAKMP SA (or IKE SA)
    - Uniquely for control traffic
    - Negotiates, creates, modifies, deletes one SA
    - The duration is longer than for IPSec SA
  - IPSec SA
    - For the protocols ESP and AH
- Key exchange
  - ✓ ISAKMP (Internet Security Association and Key Management Protocol)
    - Stores the SA, constructs the messages
  - ✓ IKE (Internet Key Exchange)
    - An implementation of ISAKMP

IPSec

- IKE (phase 1), ISAKMP SA
  - ✓ 1 : Negotiation
    - Algorithms, parameters, authentication method…
  - ✓ 2 : Diffie-Hellman (X and Y)
    - Creation of a shared secret K
    - Creation of several derived keys
  - ✓ 3 : Authentication
    - Public keys: sign the messages with the certificate
    - Shared secret: generate a common hash
    - ID: IP addresses, email, X500 dn…
- IKE (Phase 2), IPSec SA
  - ✓ Negotiation of what algorithms to use
  - ✓ Keys establishment, necessary for AH and ESP



Cisco Press

IPSec

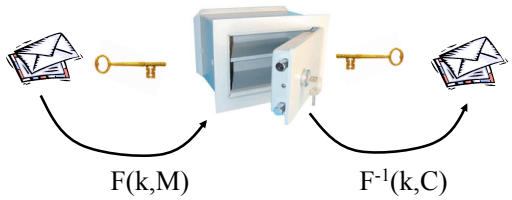
- Ipsec deployment
  - ✓ Integrated in most distributions of Linux
    - openswan to manage IKE
  - ✓ The *support tools* of Windows
- Configuration
  - ✓ No common rule
  - ✓ Interoperability sometimes problematic

IPsec - secure IP communication

- Key exchange
  - ✓ based on Diffie-Hellman (form a shared secret using public keys)
  - ✓ secret symmetrical keys used for authentication and encryption
- Authentication
  - ✓ AH (Authentication Header): encrypted hash (MD5)
- Encryption
  - ✓ ESP (Encapsulating Security Payload): 3DES
- Similar to ssh tunnel, but all upper protocols may benefit from secure communication

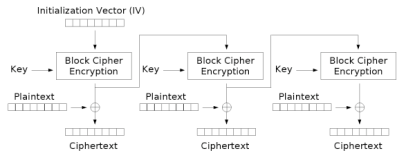
Symmetrical cryptography

- Based on so-called "private keys"



Symmetrical cryptography

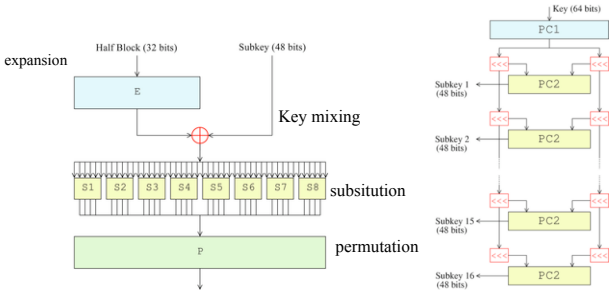
- Output Feedback
  - IV enciphered N times and XOR with block N
  - An error is not propagated
  - Vulnerability
    - Bit inversion



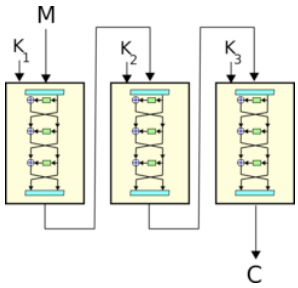
Output Feedback (OFB) mode encryption

wiki

DES

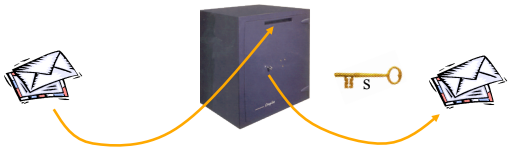


3DES



Asymmetrical cryptography

- Based on so-called "public keys"



F (p , M) G (s , C)

Diffie Hellman

- Create a secret with no prior information
  - ✓ Create symmetrical keys without having to exchange a secret over a secure channel
- Algorithm:
  - ✓ **A** and **B** send each other a number *g*
  - ✓ **A** chooses a large random integer *x*
  - ✓ **A** → $X = g^x (n)$ and sends it to **B**.
  - ✓ **B** chooses a large random integer *y*
  - ✓ **B** → $Y = g^y (n)$ and sends it to **A**.
- Then, each party at their end:
  - ✓ **A** → $k = Y^x (n)$
  - ✓ **B** → $k' = X^y (n)$
- We get:
  - ✓ $k = k' = g^{xy} (n)$
  - ✓ → Creation of a shared secret
  - ✓ In practice, n is of the order of 512 or 1024 bits
- Cryptanalysis
  - ✓ Listening-in to *X, Y, g* then reverse operation: discrete modulo n logarithm → costly
- Weakness?

Asymmetrical cryptography

- RSA
  - Choose 2 large prime numbers *p* and *q*
  - Calculate n=p.q and $\phi(n)=(p-1)(q-1)$
  - Choose random e
    - $1<e<\phi(n)$
    - and such that largest common divider $(e, \phi(n)) = 1$
  - $e.d = 1 \pmod{\phi(n)}$ (Bachet-Meriziac theorem)
  - Publish the public key (e,n), keep the private key (d,n)
- ✓ Enciphering
  - $C = M^e$ modulo(n)
- ✓ Deciphering
  - $M = C^d \pmod n$ because $C^d = (M^e)^d = M^{ed} = M$ modulo(n)
    - $M^{ed} = M$ modulo(n): this formula can be demonstrated Since $x^{\phi(n)} = 1$ (mod n) with euler theorem (if n cannot divide $\phi(n)$)

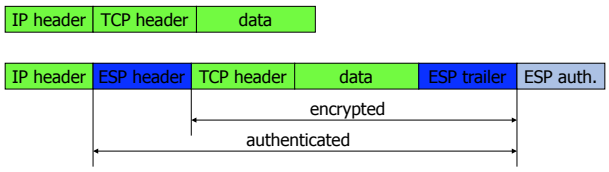
Asymmetrical cryptography

- Strength of RSA
  - You know e and n
  - You would have to find d such that
    e.d = 1 modulo ((p-1)(q-1))
  - …therefore also find p and q
  - And n=p.q
  - … but costly factorization

Asymmetrical cryptography

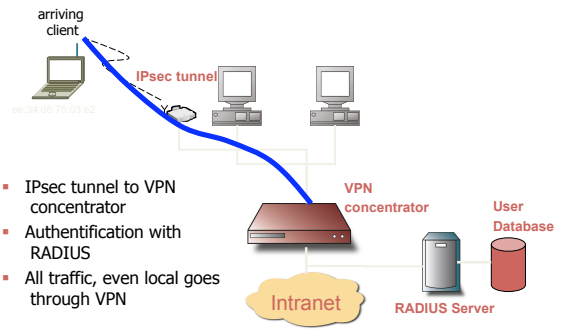
- Small example
  - p=7, q=13
  - n = p.q = 91
  - $\phi(n) = (p-1)(q-1) = 72$
  - e=5 $1<e<72$
  - $d = e^{-1} \bmod 72 = 29$ $29*5 \bmod 72 = 1$
    and Gcd (29,72) =1
  - M = 17
  - $C = M^e \bmod n = 17^5 \bmod 91 = 75$
  - $M = C^d \bmod n = 75^{29} \bmod 91 = 17$

IPsec



- Transport mode
  - ✓ only IP payload is encrypted

VPN - Virtual Private Network



- IPsec tunnel to VPN concentrator
- Authentification with RADIUS
- All traffic, even local goes through VPN

Summary

- The network layer transports packets from a sending host to the receiver host.
- Main components:
  - ✓ addressing
  - ✓ packet forwarding
  - ✓ routing protocols and routers (or how a router works)
- Routing protocols will be seen later in the advanced course
- Internet network layer
  - ✓ connectionless
  - ✓ best-effort

Question

- Ensimag has given the range of IP addresses 128.178.197/24 to the student association. The student network manager has to assign addresses to other students: they need 12 subnetworks with 12 hosts each.
  - ✓ What is the subnet mask she has to define?
  - ✓ How many hosts there can be at most on each subnetwork?
  - ✓ What is the first and the last address of subnetwork 12?
  - ✓ What is the broadcast address on subnetwork 12?

Answers

- What is the subnet mask she has to define?
  - ✓ 11111111.11111111.11111111.11110000
  - ✓ 255.255.255.240
- How many hosts there can be at most on each subnetwork?
  - ✓ 14 = 16 - 2
- What is the first and the last address of subnetwork 12?
  - ✓ 10000000.10110010.11000101.11100001
  - ✓ 192+1=193, 128.178.197.193
  - ✓ 10000000.10110010.11000101.11101110
  - ✓ 192+14=206, 128.178.197.206
- What is the broadcast address on subnetwork 12?
  - ✓ 10000000.10110010.11000101.11101111
  - ✓ 128.178.197.207

Answers

- What is the subnet mask she has to define?
  - ✓ 11111111.11111111.11111111.11110000
  - ✓ 255.255.255.240
- How many hosts there can be at most on each subnetwork?
  - ✓ 14 = 16 - 2
- What is the first and the last address of subnetwork 12?
  - ✓ 10000000.10110010.11000101.10110001
  - ✓ 176+1=177, 128.178.197.177
  - ✓ 10000000.10110010.11000101.10111110
  - ✓ 176+14=190, 128.178.197.190
- What is the broadcast address on subnetwork 12?
  - ✓ 10000000.10110010.11000101.10111111
  - ✓ 128.178.197.191

Routing

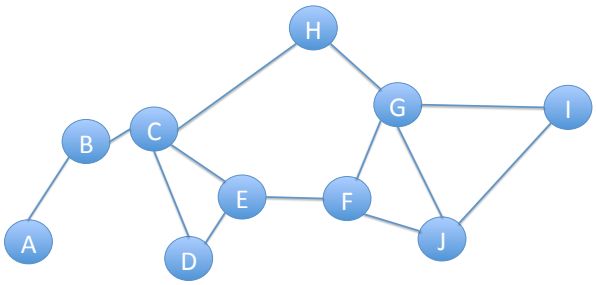
- Topology known
  - ✓ A graph with one edge per link
  - ✓ Router = vertex
  - ✓ Algorithm to find the shortest route
    - Distance (nb of hops), delay, bandwidth…
  - ✓ Low complexity
    - Large topology
    - Only a small time to fing it
      - Real time routing (a few μs)

Bellman-Ford

Dijkstra

Question

- Let execute Dijkstra in A for this topology



Answer

A	0	A	0	A	0	A	0	A	0
B	inf	B	1/A	B	inf	B	inf	B	inf
C	inf	C	inf	C	2/B	C	inf	C	inf
D	inf	D	inf	D	inf	D	3/C	D	inf
E	inf	E	inf	E	inf	E	3/C	E	inf
F	inf	F	inf	F	inf	F	inf	F	inf
G	inf	G	inf	G	inf	G	inf	G	inf
H	inf	H	inf	H	inf	H	3/C	H	inf
I	inf	I	inf	I	inf	I	inf	I	inf
J	inf	J	inf	J	inf	J	inf	J	inf