

Computer Networking

Network Layer - IP

Prof. Andrzej Duda
duda@imag.fr

http://duda.imag.fr

1

Network Layer

Chapter goals:

- understand principles behind network layer services:
 - virtual circuits vs. datagrams
 - addressing
 - packet forwarding
- instantiation and implementation in the Internet

Overview:

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

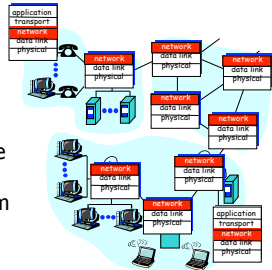
2

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



3

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)

4

Virtual circuits

"source-to-dest path behaves much like telephone circuit"

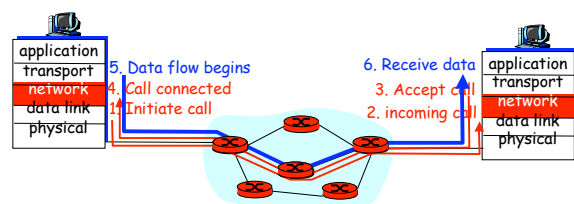
- performance-wise
- network actions along source-to-dest path

- 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

5

Virtual circuits: signaling protocols

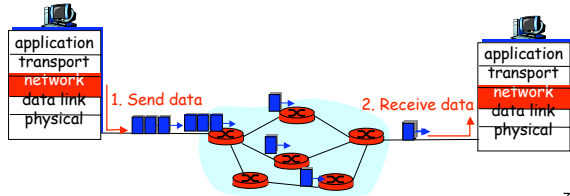
- used to setup, maintain, teardown VC
- used in ATM, Frame-Relay, X.25
- not used in today's Internet
 - but MPLS at 2.5 (between Link and Network Layer)



6

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

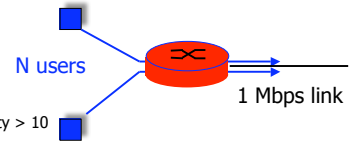


7

Packet switching vs. Circuit switching

Packet switching allows more users to use network!

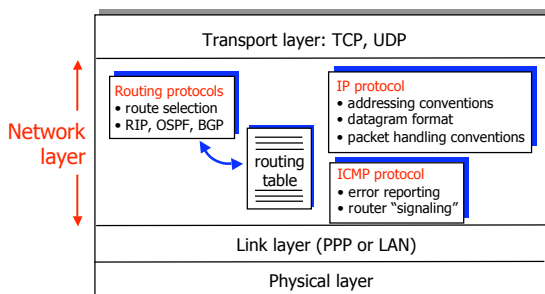
- Eg. 1 Mbit link
- each user:
 - 100 Kb/s when "active"
 - active 10% of time
- circuit-switching:
 - 10 users
- packet switching:
 - with 35 users, probability > 10 active less than .004



8

The Internet Network layer

Host, router network layer functions:



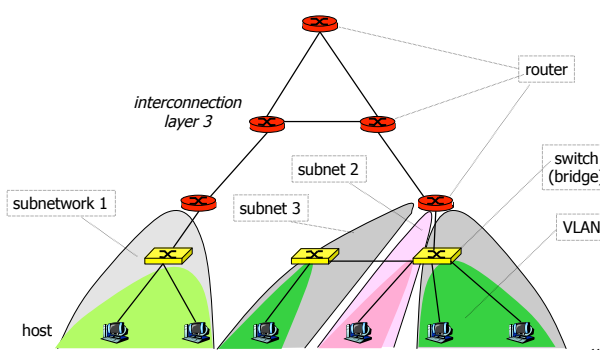
9

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

10

Interconnection structure - layer 3



11

Interconnection at layer 3

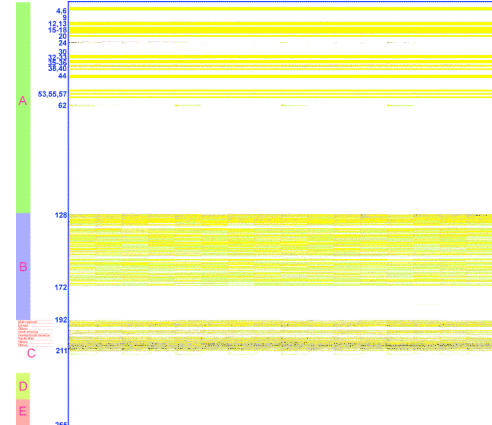
- Routers
 - interconnect subnetworks
 - logically separate groups of hosts
 - managed by one entity
- Forwarding based on IP address
 - structured address space
 - routing tables: aggregation of entries
 - works if no loops - routing protocols
 - scalable inside one administrative domain

12

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 internal use (Intranet)
 - 172.16/12
 - 192.168/16
- 1,2: source IP@ only; 3,4,5: destination IP@ only

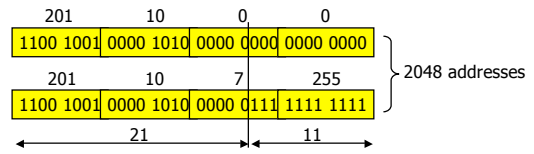
Used addresses in Internet



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.

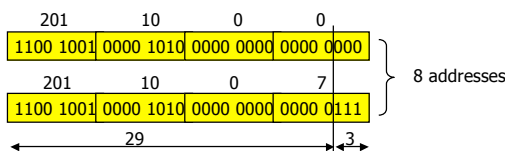
CIDR



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

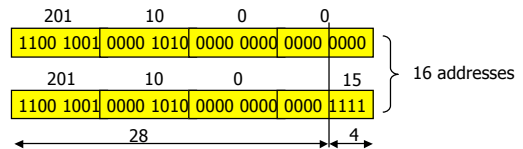
1 C class network: 256 addresses
 256 × 8 = 2048 addresses

Choosing prefix length



- prefix = 201.10.0.0/29
 - 8 addresses
 - 2 broadcast addresses: 201.10.0.0, 201.10.0.7
 - only 6 addresses can be used for hosts

Choosing prefix length



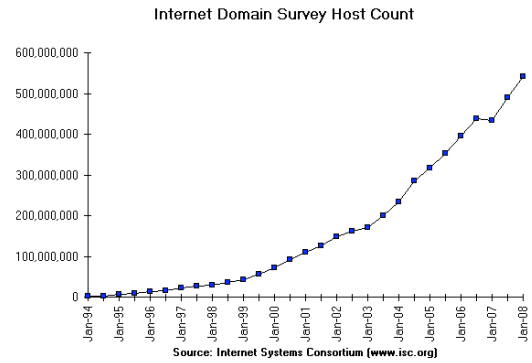
- prefix = 201.10.0.0/28
 - 201.10.0.16/28, 201.10.0.32/28, 201.10.0.48/28...
 - 16 addresses
 - 2 broadcast addresses: 201.10.0.0, 201.10.0.15
 - only 14 addresses can be used for hosts

Address allocation

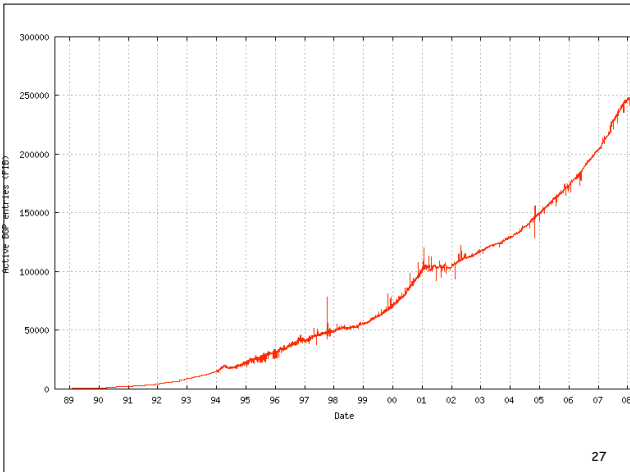
- World coverage
 - Europe and the Middle East (RIPE NCC)
 - Africa (ARIN & RIPE NCC)
 - North America (ARIN)
 - Latin America including the Caribbean (ARIN)
 - Asia-Pacific (APNIC)
- Current allocations of Class C
 - 193-195/8, 212-213/8, 217/8 for RIPE
 - 199-201/8, 204-209/8, 216/8 for ARIN
 - 202-203/8,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

25

Number of hosts



26



27

IP Addresses and subnet mask

- subnet mask at ETHZ = 255.255.0.0
- CIDR 129.132/16
- subnet mask at KTK = 255.255.255.192
- CIDR 129.132.119.64/26
- question: subnet prefix and host parts of spr13.tik.ee.ethz.ch = 129.132.119.77 ?

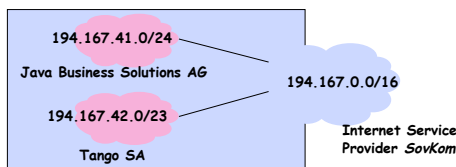
129.132.119.77 : 10000001.10000100.01110111.01001101
 255.255.255.192: 11111111.11111111.11111111.11000000

answer:

subnet prefix = 129.132.119.64 (64=01000000)
 host = 13=001101 (6 bits)

28

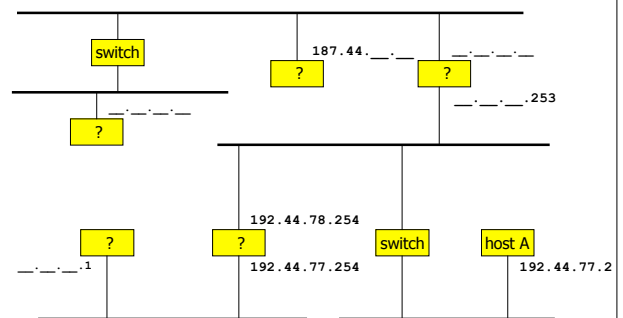
IP Addresses



- 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 194.167.42.0 to 194.167.43.255 total: $2^9 - 2$ addresses

29

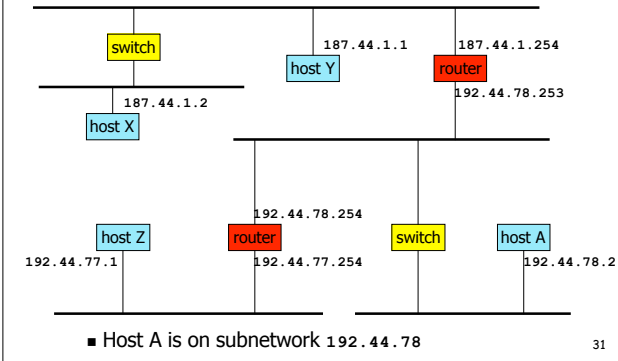
Example



- Can host A have this address?

30

Example



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 lresuns: Next Hop Table

destination@	subnetMask	nextHop
DEFAULT		128.178.156.1

Physical Interface Tables

IP	subnetMask
128.178.156.24	255.255.255.0

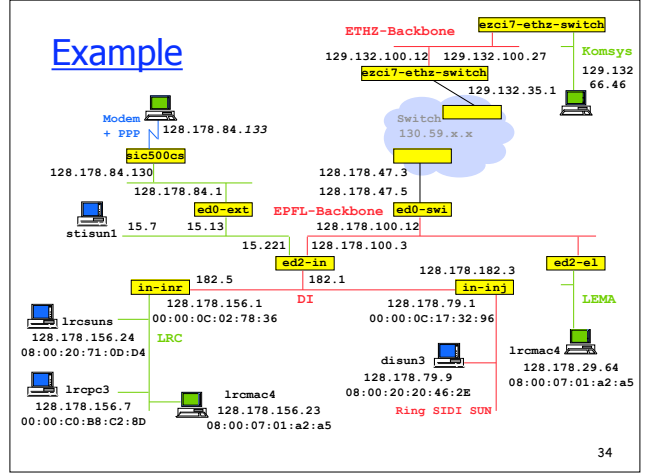
At in-inj: Next Hop Table

destination@	subnetMask	nextHop
128.178.156.0	255.255.255.0	128.178.182.5
DEFAULT		128.178.182.1

Physical Interface Tables

IP	subnetMask
128.178.79.1	255.255.255.0
128.178.182.3	255.255.255.0
	33

Example

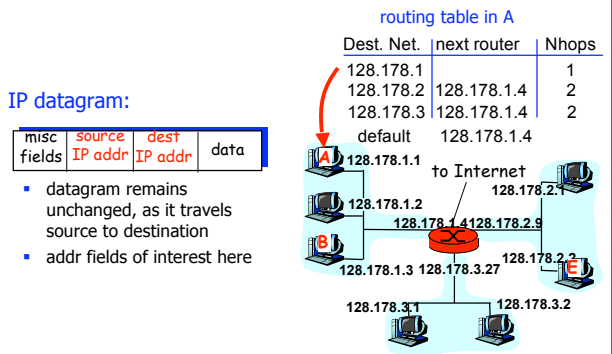


IP packet forwarding algorithm

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 35

Getting a datagram from source to dest.

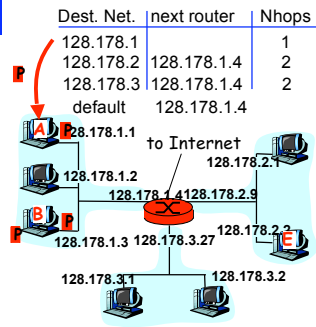


Getting a datagram from source to dest.: same subnetwork

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



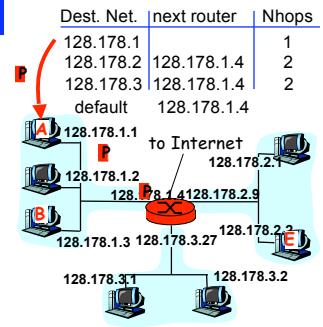
37

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.....



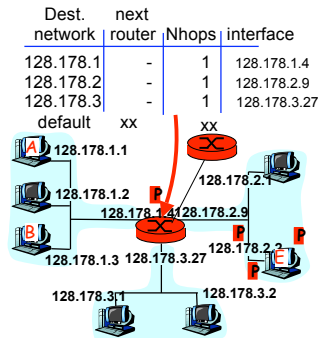
38

Getting a datagram from source to dest.: different subnetworks

misc fields	128.178.1.1	128.178.2.3	data
-------------	-------------	-------------	------

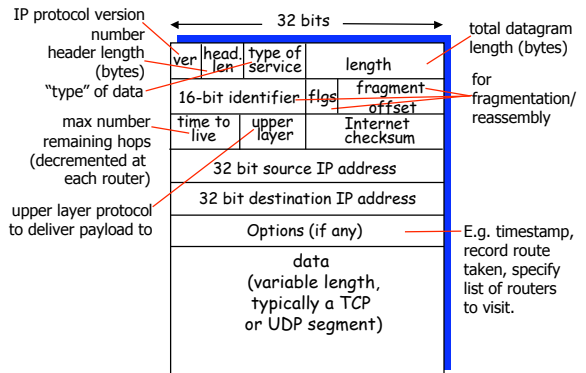
Arriving at 128.178.1.4, destined for 128.178.2.2

- 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!)



39

IP datagram format



40

IP header

- Version
 - IPv4, futur 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)

41

IP header

- Packet size
 - in bytes including header
 - in bytes including header
 - <= 64 Kbytes; limited in practice by link-level MTU (*Maximum Transmission Unit*)
 - every subnet should forward packets of 576 = 512 + 64 bytes
- Id
 - unique identifier for re-assembling
- Flags
 - M : *more* ; set in fragments
 - F : prohibits fragmentation

42

IP header

- Offset
 - position of a fragment in multiples of 8 bytes
- TTL (*Time-to-live*)
 - in secondes
 - 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

43

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

44

Configuration of a Unix host

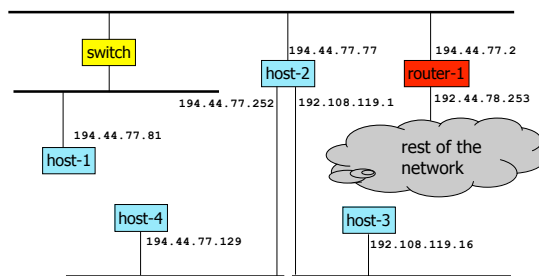
```

/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 host-1 netmask +
Setting netmask of le0 to 255.255.255.128
# + means netmask from /etc/netmasks
host-1# ifconfig -a
le0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING>
    inet 192.44.77.81 netmask ffffffff 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
    
```

45

Example interconnection



46

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
    
```

47

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
194.44.77.128 194.44.77.252 U       26   504768 le2
194.44.77.0    194.44.77.77 U       24   10702069 le0
192.108.119.0 192.108.119.1 U       2    249777 le1
    
```

48

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
    
```

49

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
    
```

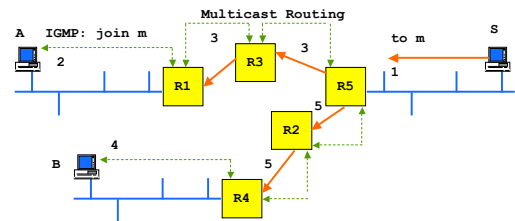
50

IP Broadcasting, Multicasting

- Broadcast = send to all
 - sent to all hosts on one net/subnet; used by NetBIOS for discovery
- Multicast = send to a group
 - IP multicast address = class D = 224.0.0.0 to 239.255.255.255
 - 224.0.0.1 = all multicast capable systems on subnet
 - 224.0.0.2 = all multicast capable routers on subnet
 - used for: routing, conferencing, radio distribution, ...
- IP uses open group paradigm
 - multicast IP addresses are logical (= non topological)
 - for receiving data sent to multicast address *m*, a host must subscribe to *m*
 - for sending to multicast address *m*, a host simply put *m* in the dest addr field

51

IP Multicast Principles



- hosts subscribe via IGMP join messages sent to router
- routers build distribution tree via multicast routing
- sources do not know their destinations
- packet replication is done by routers

52

IP Multicast Forwarding Algorithm

Packet Forwarding (host, router)

```

Read address MA = destination IP@
/* assume it is multicast */
for every physical interface PI
  if MA is enabled on PI then
    send directly to PI
    
```

At lrcsuna: Physical Interface Tables

IP	subnetMask
128.178.156.24	255.255.255.0
224.2.166.207	
224.2.127.255	

Send directly (Ethernet)

```

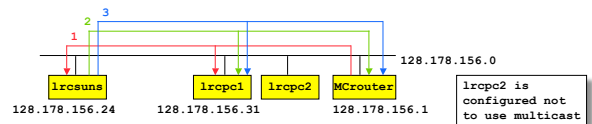
send directly(MA, MAC@):
  map last 23 bits of MA to last 23 bits
  of MAC address
  send MAC frame with      DA = 01-00-5E-xx-xx-xx,
                          SA = own i/f address
    
```

- Systems have to know which group they belong to
 - Hosts: application processes register to IP
 - Routers: learn if members present with IGMP
- Direct send to link layer:
 - algorithmic mapping of 23 last bits : ex : 224.2.166.207 -> 01-00-5E-02-A6-CF

53

IGMP: Internet Group Management Protocol

- Purpose: manage group membership inside one subnet
- routers: know if group is present on an interface
 - know whether to forward locally or not
- hosts: know if a multicast address is already in use locally



- 1: IGMP query, TTL=1, IGMP group @ = 0, dest IP@ = 224.0.0.1; source IP@ = 128.178.156.1
- 2: IGMP report, TTL=1, IGMP group @ = 224.2.166.207, dest IP@ = 224.2.166.207; source IP@ = 128.178.156.24
- 3: IGMP report, TTL=1, IGMP group @ = 224.2.127.255, dest IP@ = 224.2.127.255; source IP@ = 128.178.156.24

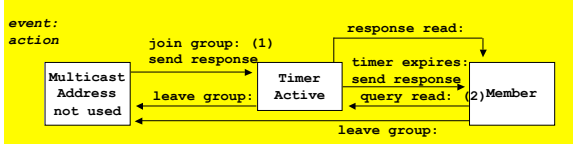
54

IGMP Host Implementation

Host Implementation

- goal: avoid avalanche effects - one router originated query might cause a burst of reports
- solution = synchronization avoidance protocol
 - 1. hosts delay responses randomly
 - 2. hosts listen to responses, only first one answers

Host IGMP Finite State Machine



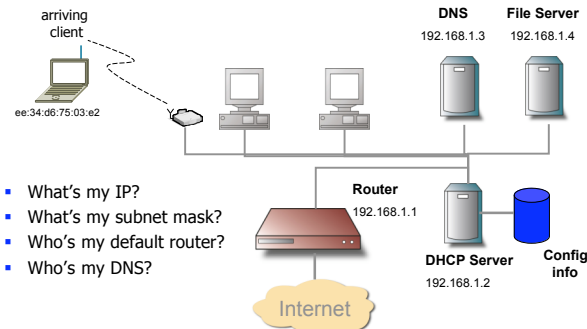
DHCP

- 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)
 - IP source address 0, broadcast 255.255.255.255

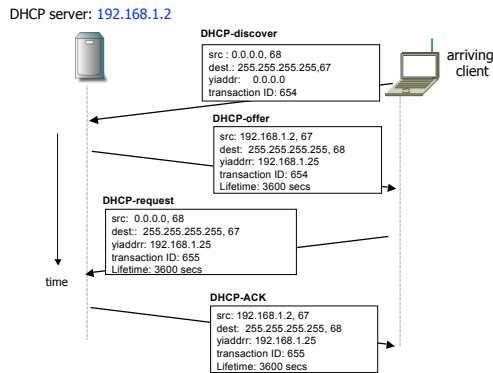
DHCP

- 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

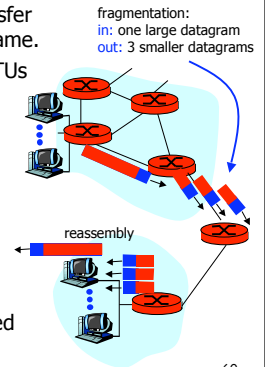


DHCP client-server scenario



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 ?

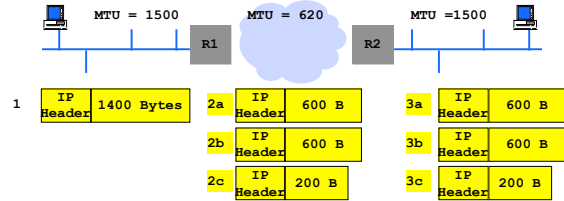
Link technology	MTU
Ethernet	1500
802.3 with LLC/SNAP	1492
FDDI	4352
X.25	576
Frame Relay	1600
ATM with AALS	9180
Hyperchannel	65535
PPP	296 to 1500

```
lrcsuns$ ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
    inet 127.0.0.1 netmask f0000000
le0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 128.178.156.24 netmask ffffffff 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

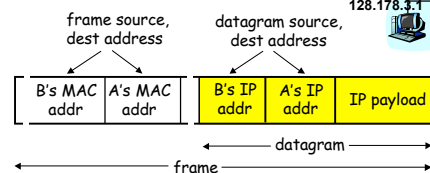
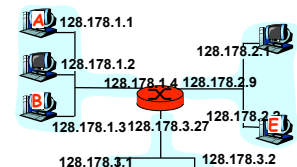
Why different addresses at IP and MAC?

- 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

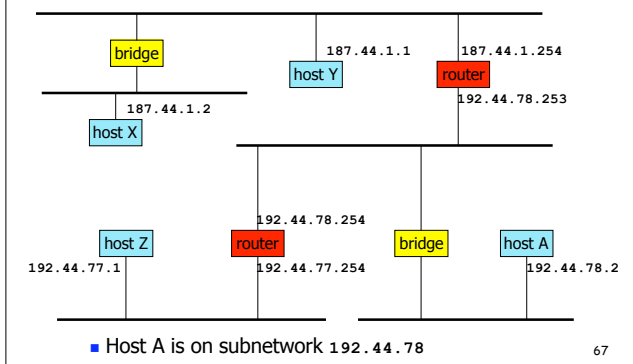
MAC Address resolution

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



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)

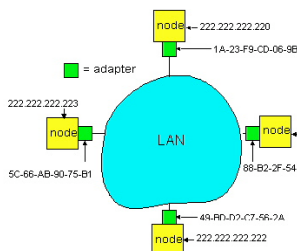
Router needs to know MAC address of A

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

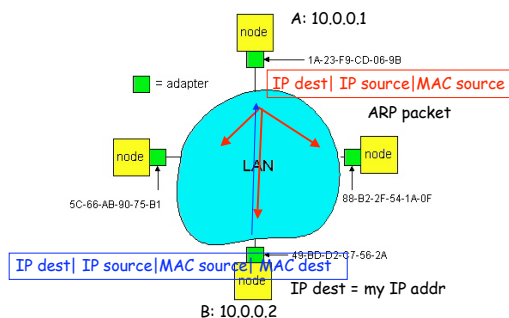


ARP protocol

- 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)

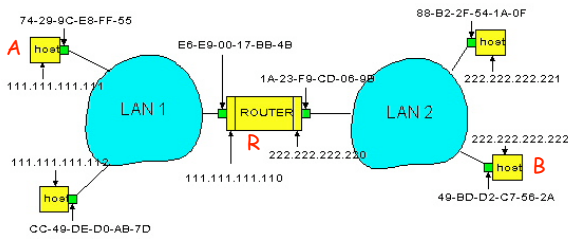
sender Ethernet address
sender IP address
target Ethernet address ???
target IP address

- 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

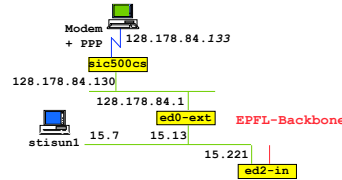


- In routing table at source Host, find router 111.111.111.110
- In ARP table at source, find MAC address E6-E9-00-17-BB-4B, etc

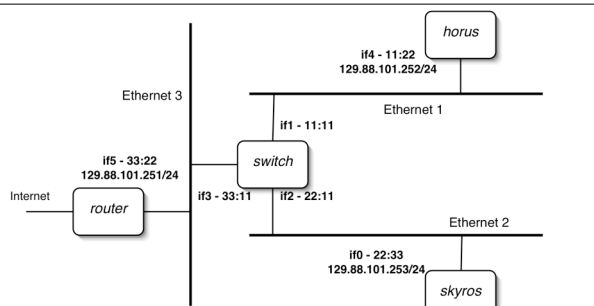
73

Proxy ARP

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



74



- I'm executing the following commands on **skyros**:
 - `ping 129.88.101.252`
 - `ping 195.221.19.1`
- What will you observe on the LANs?

75

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

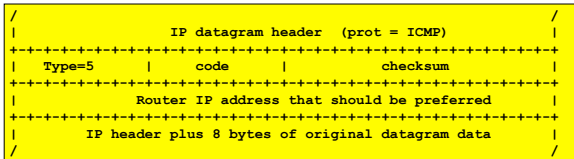
Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	router advertisement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

77

ICMP Redirect

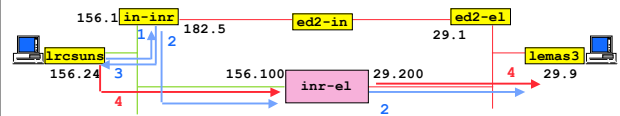
- Sent by router to source host to inform source that destination is directly connected
 - host updates the routing table
 - ICMP redirect can be used to update the router table (eg. in-inj route to LRC?)

ICMP Redirect Format



78

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	icmp	type=redir code=host cksum 128.178.156.100 xxxxxxx (28 bytes of 1)
4:	128.178.29.9	128.178.156.24	udp

79

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  e0
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
```

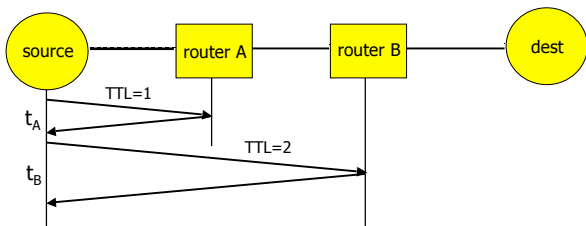
80

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
 - ...

81

Traceroute



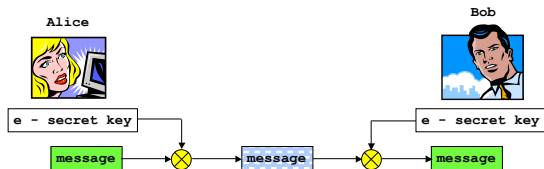
82

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

83

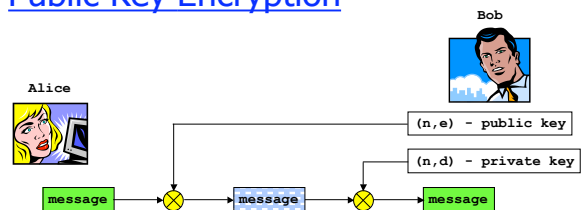
Secret Key Encryption



- Secret key encryption (DES, 3DES,...)
 - encrypted message $c = f(e, m)$
 - decrypted message $m = f^{-1}(e, c)$
- Must exchange the key
- Efficient encryption

84

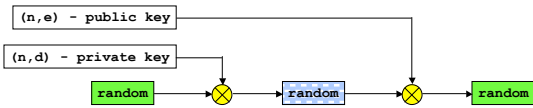
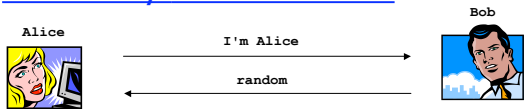
Public Key Encryption



- RSA encryption
 - encrypted message $c = (m^e \text{ mod } n)$
 - decrypted message $m = (c^d \text{ mod } n)$
- Key property
 - $(m^e)^d \text{ mod } n = m$
- Slow

85

Public Key Authentication



- Authentication
 - random challenge (nonce), used only once
- Bob verifies
 - $(r^d)^e \bmod n = r$

86

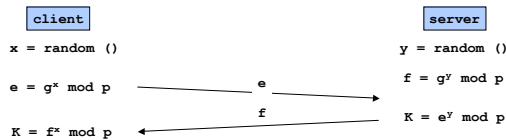
Integrity - digital signature



- Hash, digest, or MAC (Message Authentication Code)
 - 128 or 160 bits (MD5, SHA-1)
- Bob decrypts H(m) using the public key and verifies if
 - $H(m) = H(\text{message})$

87

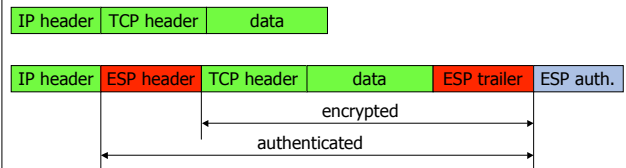
Diffie-Hellman key exchange



- Known parameters
 - g - generator** (e.g. $g = 2$)
 - p - large prime**
 - e.g. $2^{1024} - 2^{960} - 1 + 2^{64} \lfloor \text{floor}(2^{894} \pi + 129093) \rfloor$
 - $1 < x, y < (p - 1) / 2$
 - $K = (g^x \bmod p)^y \bmod p = (g^y \bmod p)^x \bmod p$

88

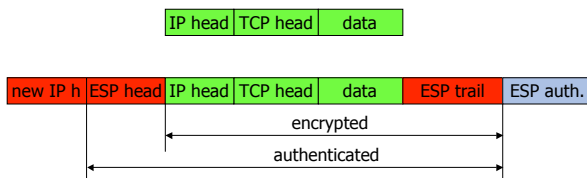
IPsec



- Transport mode
 - only IP payload is encrypted

89

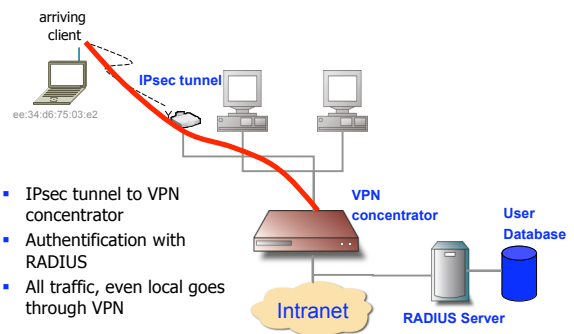
IPsec



- Tunnel mode
 - new IP header

90

VPN - Virtual Private Network



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

91

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

92