# The Internet Protocol

Stefan D. Bruda

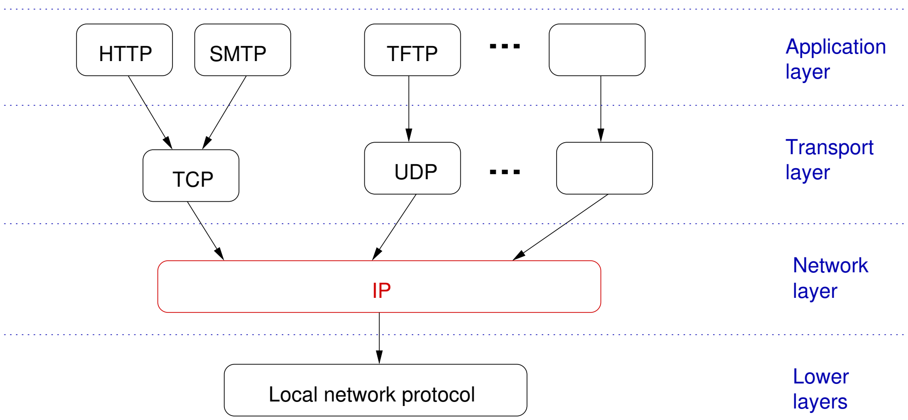
CS 464/564, Fall 2023

## The Internet Protocol

- A (connectionless) network layer protocol
- Designed for use in interconnected systems of packet-switched computer communication networks (store-and-forward paradigm)
  - Each participant maintains queues for incoming and outgoing packets
- Provides for transmitting blocks of data called datagrams from sources to destinations
  - The datagram may possibly go through intermediate hosts
  - Sources and destinations are hosts identified by fixed length addresses
- Also provides for fragmentation and reassembly of long datagrams, if necessary, for transmission through “small packet” networks
- The workhorse of data exchange
- Both TCP and UDP use it to carry packets from one host to another
- Much like UDP (which is thus a thin layer on top of IP) in behaviour



## Relation to other protocols






## Interfaces

- IP is called on by host-to-host protocols in an internet environment
- In turn, IP calls on local network protocols to carry the internet datagram to the next gateway or destination host
- A participating endpoint host needs to know its IP address (192.168.0.1), netmask (255.255.255.0), and its gateway address(es) (192.168.0.254)
  - A host can infer its broadcast address (192.168.0.255) whose use implies the sending of the datagram to all the hosts within the netmask
    - With these coordinates, the host sits in the 192.168.0.0/24 network
  - Anything addressed to an IP address within the netmask is passed directly to the lower network layer (MAC)
  - Other datagrams are sent to the gateway by calling once more the lower layer
- The gateway is a host that connects to two (or more) networks via two (or more) local network interfaces. It is also called a router

# IP addresses

- IP addresses have a fixed length of four bytes (32 bits)
- An address begins with a network number, followed by local address (called the ”rest” field)
  - For instance 192.168.0.15 is formed from the 192.168.0.0 (192.168.0.0/24) network number followed by the local address 15
- Three classes of IP addresses (historical importance only):
  - Class A high order bit is 0, next 7 bits are the network, the last 24 bits are the rest


  - Class B high order bits are 10, next 14 bits are the network, last 16 bits are the rest
  - Class C high order bits are 110, next 21 bits are the network, last 8 bits are the rest
- Nowadays the network and the rest are given exclusively by the netmask



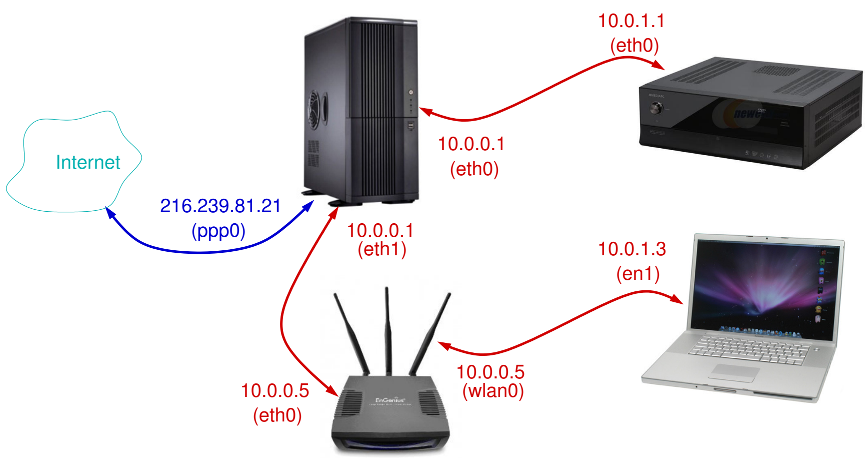
# Private networks

- A private network uses private IP address spaces (RFC 1918, RFC 4193)
- Private addresses are not globally delegated
  - They are not allocated to any specific organization; IP packets addressed by them cannot be transmitted onto the public Internet
  - If a private network needs to connect to the Internet, it must use either a network address translator (NAT), or a proxy server
- Private addresses can in fact coexist with “real” addresses
- Private IP ranges:

| Class | Address range | No. addresses | Mask | Rest |
|---|---|---|---|---|
| One class A | 10.0.0.0–10.255.255.255 | 16,777,216 | 255.0.0.0 | 24 bits |
| 16 class B | 172.16.0.0–172.31.255.255 | 1,048,576 | 255.240.0.0 | 20 bits |
| 256 class C | 192.168.0.0–192.168.255.255 | 65,536 | 255.255.0.0 | 16 bits |

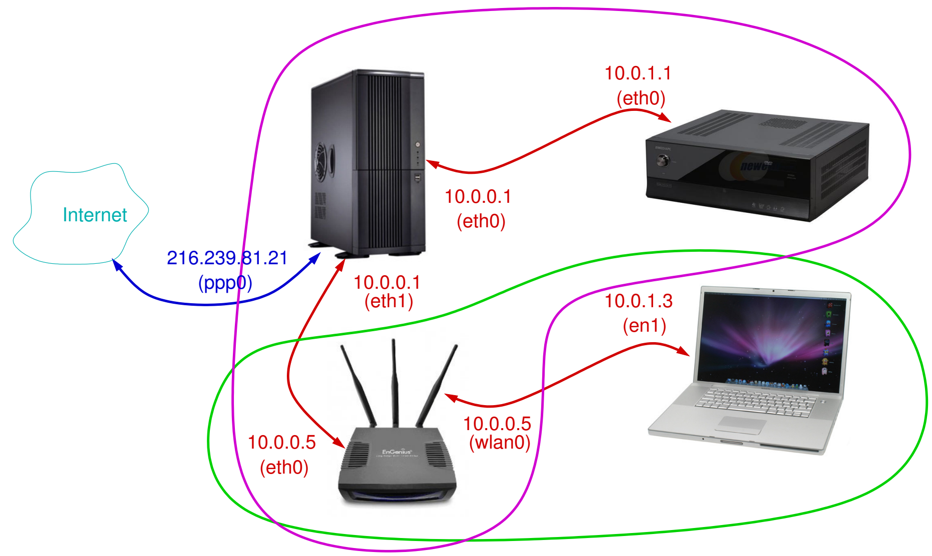


# Interfaces (cont’d)

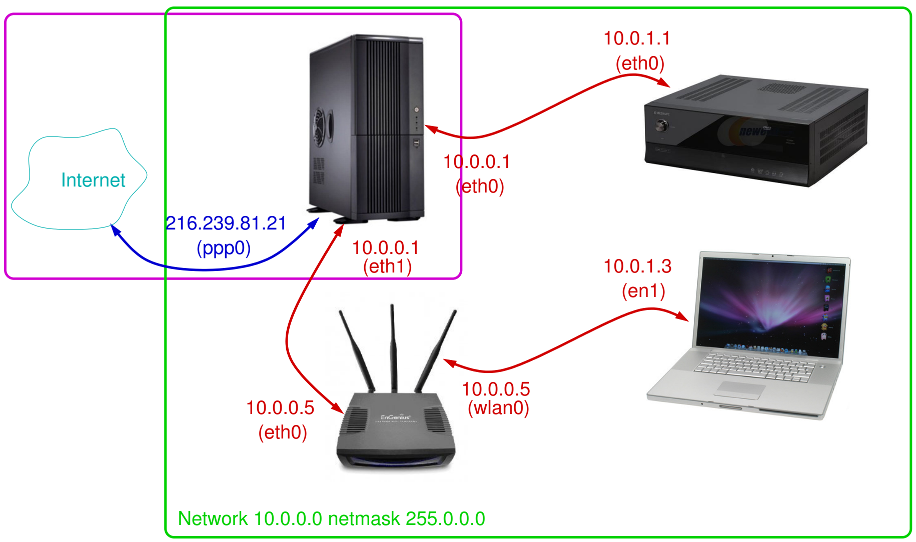





# Interfaces (cont’d)



# Interfaces (cont'd)






# Local network

- The network layer does not like multiple interfaces with the same IP address
  - So this kind of interfaces must be bridged into a single (virtual) interface – a level 2 layer operation
- The gateway (or router) knows how to route packets based on a routing table:

```
< pascal:/etc/conf.d > route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags   Use   Iface
0.0.0.0         216.239.80.253  0.0.0.0         UG        0   ppp0
10.0.0.0        0.0.0.0         255.0.0.0       U         0   br0
127.0.0.0       127.0.0.1       255.0.0.0       UG        0   lo
192.168.1.0     0.0.0.0         255.255.255.0   U         0   eth4
216.239.80.253  0.0.0.0         255.255.255.255 UH        0   ppp0
```

- Every machine in the `10.0.0.0/8` network specifies the router as "default gateway"

```
route add default gw 10.0.0.1
```

  - The gateway must be reachable at the level 2 layer!



# Example

- 10.0.1.3 sends a TCP packet to 216.109.118.67
- TCP calls on the IP to take a TCP packet (including the TCP header and user data) as the data portion of a datagram
  - TCP provides the addresses and other parameters
- IP assembles the datagram, notices that 216.109.118.67 is not a local address, and thus sends the packet to the gateway (10.0.0.1) through eth0
- The gateway receives the packet and repeats the same algorithm
  - the destination address is not in the 10.0.0.0 network, so the gateway sends the packet through its ppp0 interface
  - NAT also takes place here (whenever applicable)



# Operation of IP

Two basic functions: addressing and fragmentation

- IP modules use the addresses carried in the internet header to transmit internet datagrams toward their destinations, hop by hop
- This (distributed) selection of a transmission path is called routing
- In the process the packets may be fragmented
  - the fragmenting and reassembling is the exclusive duty of IP
- The model of operation is that an IP module resides in each host engaged in internet communication and in each gateway that interconnects networks
  - These modules share common rules for interpreting address fields and for fragmenting and assembling internet datagrams
  - In addition, these modules (especially in gateways) have procedures for making routing decisions and other functions (routing algorithms)
  - IP treats each internet datagram as an independent entity unrelated to any other internet datagram
    - There are no connections or logical circuits

## Key mechanisms

- Type of Service indicates the quality of the service desired
  - Abstract or generalized set of parameters which characterize the service choices provided in the networks that make up the internet
  - Used by gateways to select the actual transmission parameters, the network to be used for the next hop, or the next gateway
- Time to Live is an upper bound on the lifetime of an internet datagram
  - Set by the sender and reduced at the points along the route where it is processed
  - If the time to live reaches zero before the datagram reaches its destination, the datagram is destroyed
- Options provide for control functions needed or useful in some situations but unnecessary for the most common communications
  - Include provisions for timestamps, security, and special routing
- Header Checksum provides a verification that the information used in processing internet datagram has been transmitted correctly
  - If the header checksum fails, the datagram is discarded at once by the entity which detects the error



## Routing

- The internet protocol does not provide a reliable communication facility
  - No acknowledgments (either end-to-end or hop-by-hop)
  - No error control for data (only a header checksum)
  - Mo retransmissions
  - No flow control
- IP provides a store-and-forward, packet switching internet
  - Datagrams are stored into queues in various routers and forwarded between routers until they reach their destination
- An IP datagram has the capability to provide a route to be followed
  - A route is a sequence of IP addresses
  - Split into two parts
    - Recorded route or the route travelled so far, and
    - Source route or the route yet to be followed
  - Then the routing algorithm is very simple
  - However, is the source route becomes empty at some point, the routing algorithm forwards the datagram solely according to the destination address
    - The recorded route continues to be filled in
    - We then enter the realm of real routing algorithms



## The optimality principle

- All the routing algorithms are based on the optimality principle:

  *If a router J is on the optimal path from router I to router K then the optimal path from J to K also falls along the same route*
- As a consequence, the set of all the optimal routes from all the sources to a given destination form a tree rooted at the destination (the sink tree)
  - No loops, so each packet will be delivered after a finite number of hops if following the optimal route
  - In practice life is not that easy
    - Links and routers go down and come back up
    - The picture a router has for the internet is not necessarily the same as the picture other routers have



## Routing algorithms

Various algorithms have been used, including:

- Flooding: every incoming datagram is sent to every outgoing line
  - Is there any possibility that the number of duplicate datagrams increase without bounds?
  - Flooding is very inefficient, but has its uses (e.g., in military applications)
- Shortest path routing: when forwarding a packet, a router computes the shortest path to the destination and sends the datagram to the next hop along this path
  - The metrics used for paths are varied, including the number of hops, the geographic distance, and delivery delay (including queuing or not)
  - The shortest path is computed using a greedy algorithm such as Dijkstra's
- Distance vector routing: (ARPANET until 1979) each router maintains a table giving the best known distance to each destination and which network interface to use to get there
  - Tables are constructed by exchanging information between routers

# Link state routing

- Most widely used algorithm nowadays
- Each router performs an algorithm consisting of the following steps:
  1. Discover the neighbours and learn their network addresses
  2. Measure the delay or cost to each neighbour
  3. Construct a packet telling all it just learned
  4. Send this packet to all the other routers
  5. Compute the shortest path to all the other routers
- In effect, the topology of the network and the delays are experimentally discovered/measured
- Dijkstra's algorithm or equivalent can then be used to find the shortest path



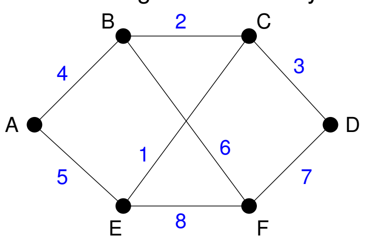
# Learn about neighbours, measure delays

- Once a router is booted, it sends a "HELLO" packet to each interface
  - inter-router links are conceptually viewed as point-to-point
  - the router on the other end is supposed to send back a reply telling who it is
  - the names must be globally unique (e.g., the MAC address)
- The router sends then an "ECHO" packet to its neighbours, which is bounced back immediately
  - reasonable estimate of the delay
  - may include actual network traffic (by including queueing time) or not



# Constructing the link state packets

- A link state packet contains the identity of the sender, a sequence number, age, and a list of neighbours
  - For each neighbour the delay to that neighbour is also given




| A | B | C | D | E | F |
|---|---|---|---|---|---|
| Seq. | Seq. | Seq. | Seq. | Seq. | Seq. |
| Age | Age | Age | Age | Age | Age |
| B / 4 | A / 4 | B / 2 | C / 3 | A / 5 | B / 6 |
| E / 5 | C / 2 | D / 3 | F / 7 | C / 1 | D / 7 |
| | F / 6 | E / 1 | | F / 8 | E / 8 |

  - Issue: when to build link state packets?
    - Periodically, or
    - Whenever a significant event occurs (neighbour goes down, neighbour comes back up, neighbour communication changes properties dramatically)

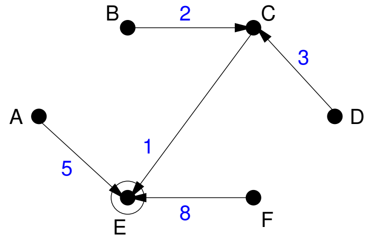


# Distribute link state packets

- We use flooding
- Routers keep track of all the (most recent versions of) source–sequence pairs they see to contain flooding
  - When a new packet comes in, it is checked against the corresponding pair
  - If it is new, it is forwarded on all the lines
  - If the stored sequence number is larger (or is a duplicate), the packet is discarded
  - The age is decremented each second and the packet is discarded when age reaches zero; this guards against corrupted or wrapped sequence numbers

# Compute routes

- Once a router has accumulated all the packets, it can reconstruct the network graph
  - Each edge in the graph is actually represented twice, once for each direction
- Now we run Dijkstra's algorithm at each router to compute the minimum-cost spanning tree from the router to all the other destination




- The result is installed in the router as a routing table and normal operation begins
  - The routing table does not store the whole tree, just pairs destination–next hop

| Destination | Next hop |
|---|---|
| A | A |
| F | F |
| C | C |
| B | C |
| C | C |



# Characteristics

- For an internet with $n$ routers of degree $k$ the memory required to store the routing table is $O(k \times n)$
  - For large internets this can be a problem
  - However routing tables can often be reduced in size dramatically by using a "default" (or "everything else") entry:

| Destination | Next hop |
|---|---|
| A | A |
| F | F |
| default | C |

  - Additionally, the Internet is a huge place, but internets are not very large since they are separated by "border" routers with routing tables that look like this:

```
Destination     Gateway         Genmask         Iface
10.0.0.0        0.0.0.0         255.0.0.0       lan
0.0.0.0         216.239.80.245  0.0.0.0         wan
```

- Link state routing is sensitive to hardware failure (but what algorithm isn't?)
- In practical settings link state routing works well, so (slightly improved) variants are in wide use today