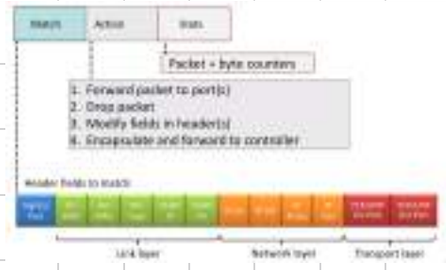
▸ Network Layer : Control Plane

- **Generalized forwarding**
  - → match + action → packet arrived, take action
  - → each router contains a forwarding table

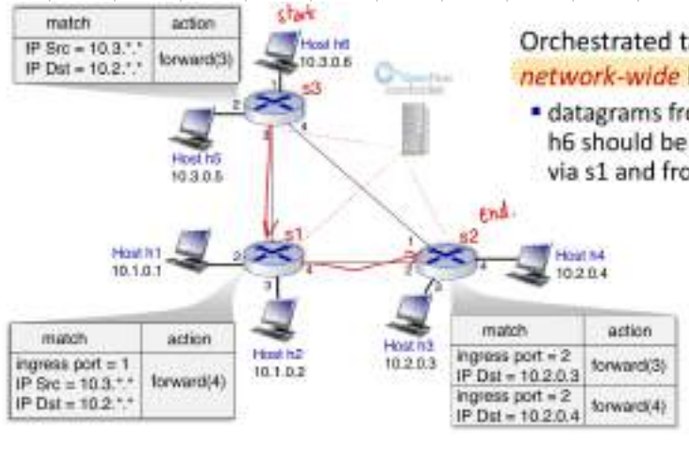
- **Flow table abstraction**
  - → flow : defined by header fields
  - → rules of flow to handle a packet
  - **OpenFlow** : flow table entries



"Program" Network-wide behaviors.

↳ Router : Match: IP Prefix, Action: forward to output link
↳ Firewall : Match: IP & TCP/UDP Port, Action: permit / deny
↳ Switch : Match: Dest. MAC Address, Action: forward / flood
↳ NAT : Match: IP & Port, Action: rewrite address & port

→ Example :



Orchestrated tables can create *network-wide* behavior, e.g.,:
- datagrams from hosts h5 and h6 should be sent to h3 or h4, via s1 and from there to s2

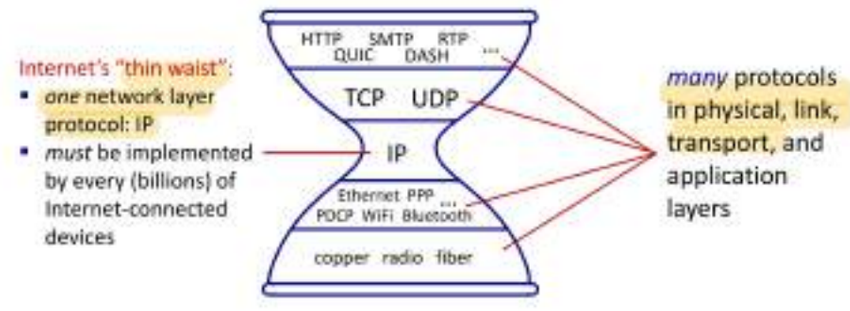
- **Middleboxes** "any intermidary box performing function apart from normal, standard fn. of an IP router on data path b/w source & dest. host."

ex. NAT, Firewalls, Load Balancers, Caches, Application-Specific.

→ move towards "white box" hardware
↳ programmable local actions.

- SDN : logically centralized control / config management.
- NFV : programable services over white box networking

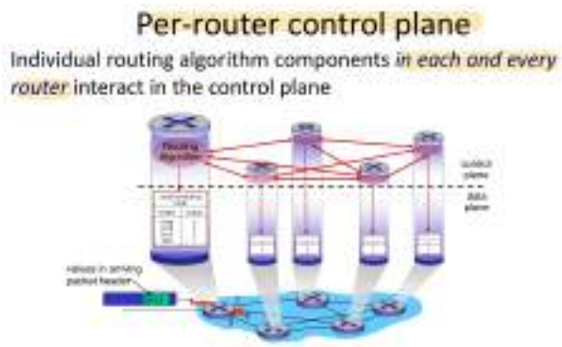
- **IP Hourglass**



→ How are forwarding tables (Dest.-based forwarding) or flow tables (generalized forwarding) computed?

⇒ **Control Plane**

→ 2 ways to structuring network control plane :
- per-router control : Traditional.
- logically centralized control : SDN



- Routing Protocols
  - → "determine good paths" b/w source to dest.
  - → "Good": least cost, fastest, least congested.
  - → Abstraction: GRAPH
    - ↳ Router: nodes ↳ link: edges ↳ cost

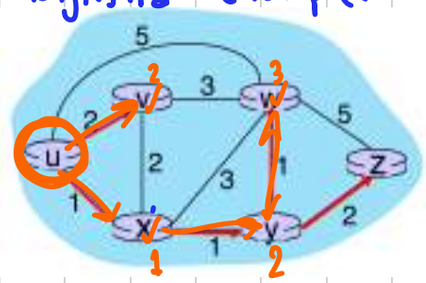
- Routing Algorithm Classification

→ How fast routes change?
- Static: slow
- Dynamic: quickly periodically updated.

→ Information?
- Global: Link State (LS) complete topology
- Decentralized: Distance Vector (DV) Iterative, exchange w/ neighbors router

- Dijkstra Algorithm: Link-state
  - → Centralized; link costs known to all nodes
  - → Iterative; after k rounds, known least cost path to k destinations

Dijkstra Example.



from u

# round	visited nodes	cost to destination nodes				
#	N'	v	w	x	y	z
0	u	2,u	5,u	1,u	∞	∞
1	u, x	2,u	4,x	1,u	2,x	∞
2	u,x,v	2,u	4,x	1,u	2,x	∞
3	u,x,v,y	2,u	3,y	1,u	2,x	4,y
4	u,x,v,y,w	2,u	3,y	1,u	2,x	4,y
5	u,x,v,y,w,z	2,u	3,y	1,u	2,x	4,y

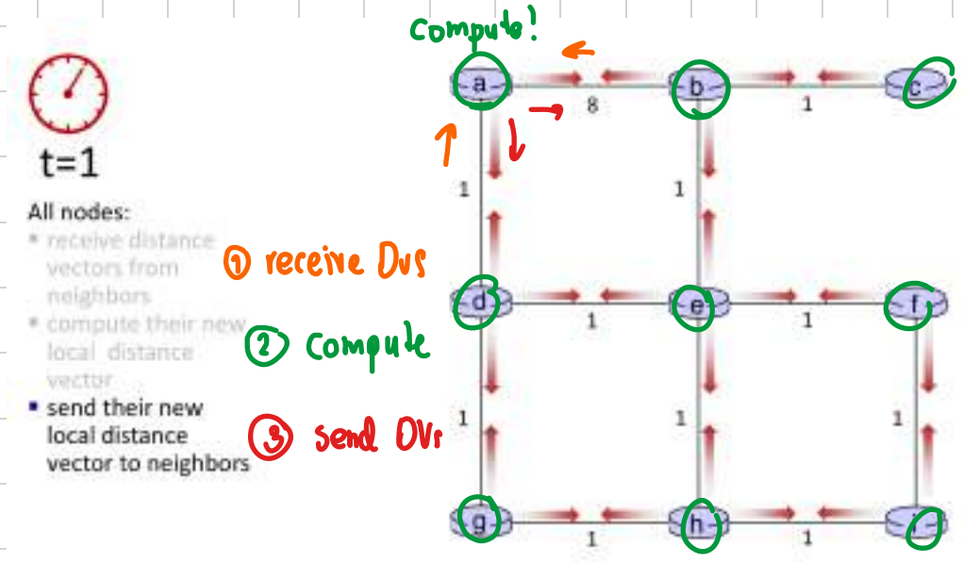
Complexity: $O(n^2)$

more eff: $O(n \lg n)$

- Bellman-Ford Algorithm: Distance Vector (DV)

$$D_x(y) \leftarrow \min_v\{c_{x,v} + D_v(y)\}$$

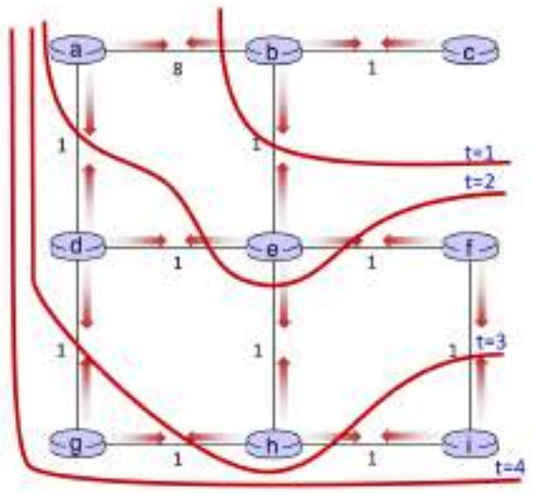
→ each nodes sends its own DV estimate to neighbors



Distance vector: state information diffusion

Iterative communication, computation steps diffuses information through network:

- t=0 c's state at t=0 is at c only
- t=1 c's state at t=0 has propagated to b, and may influence distance vector computations up to **1** hop away, i.e., at b
- t=2 c's state at t=0 may now influence distance vector computations up to **2** hops away, i.e., at b and now at a, e as well
- t=3 c's state at t=0 may influence distance vector computations up to **3** hops away, i.e., at b,a,e and now at c,f,h as well
- t=4 c's state at t=0 may influence distance vector computations up to **4** hops away, i.e., at b,a,e, c, f, h and now at g,i as well



link cost changes:

- node detects local link cost change
- updates routing info, recalculates local DV
- if DV changes, notify neighbors

"good news travel fast, bad news travel slow"

Comparison of LS and DV algorithms

message complexity

LS: n routers, $O(n^2)$ messages sent

DV: exchange between neighbors; convergence time varies

speed of convergence

LS: $O(n^2)$ algorithm, $O(n^2)$ messages
- may have oscillations

DV: convergence time varies
- may have routing loops
- count-to-infinity problem

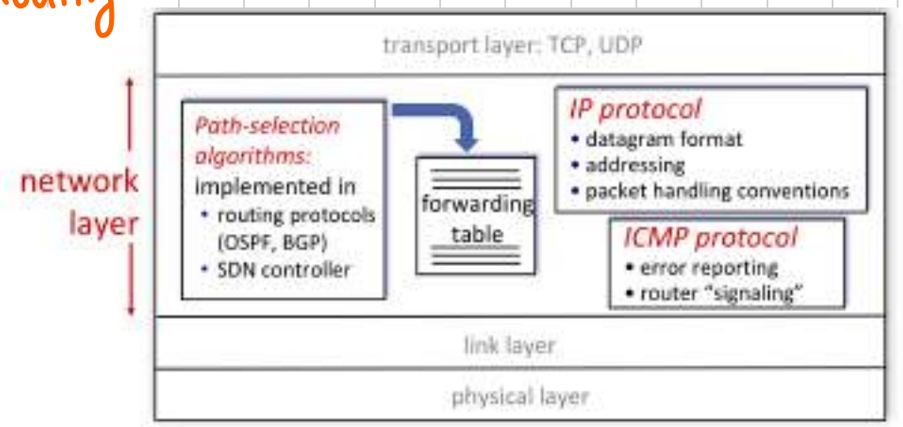
robustness: what happens if router malfunctions, or is compromised?

LS:
- router can advertise incorrect *link* cost
- each router computes only its *own* table

DV:
- DV router can advertise incorrect *path* cost ("I have a *really* low cost path to everywhere"): black-holing
- each router's table used by others: error propagate thru network

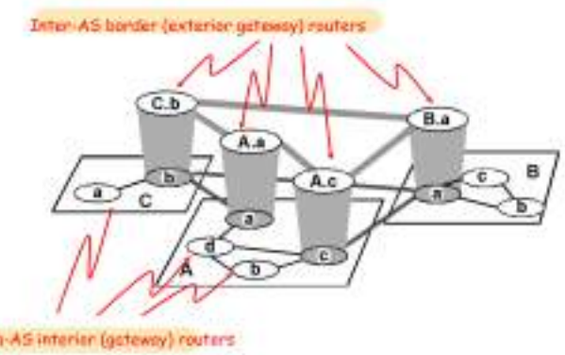
• ISP Routing

internet



intra-AS (aka "intra-domain"):
routing among *within same AS ("network")*
- all routers in AS must run same intra-domain protocol
- routers in different AS can run different intra-domain routing protocols
- gateway router: at "edge" of its own AS, has link(s) to router(s) in other AS'es

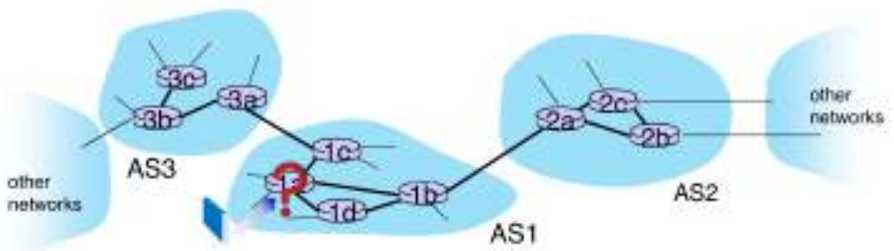
inter-AS (aka "inter-domain"):
routing *among* AS'es
- gateways perform inter-domain routing (as well as intra-domain routing)



Inter-AS routing: a role in intradomain forwarding

- suppose router in AS1 receives datagram destined outside of AS1:
  - router should forward packet to gateway router in AS1, but which one?

AS1 inter-domain routing must:
1. learn which destinations reachable through AS2, which through AS3
2. propagate this reachability info to all routers in AS1



- Intra-AS Routing
  - RIP : routing information protocol / no longer used
  - EIGRP : Enhanced Interior Gateway Routing Protocol
    - ↳ DV
  - OSPF : Open Shortest Path First
    - ↳ LS ↳ IS-IS Protocol

RIP: Routing Information Protocol

- Distance vector (DV) algorithm
- Included in BSD-UNIX Distribution in 1982
- Distance metric: # of hops (max = 15 hops)
- Distance vectors: exchanged every 30 sec via Response Message (also called **advertisement**)
- Each advertisement: route to up to 25 destination nets
- RIPv2 (Authentication, Multicast, VLSM/CIDR)

EIGRP: Enhanced Interior Gateway Routing Protocol

- An advanced distance vector routing protocol developed by Cisco Systems.
- EIGRP is an enhancement of another Cisco routing protocol IGRP
- EIGRP provide extremely **quick convergence** times with minimal network traffic

By default, EIGRP uses the following values in its composite metric to calculate the preferred path to a network:

Bandwidth - The slowest bandwidth among all of the outgoing interfaces, along the path from source to destination.

Delay - The cumulative (sum) of all interface delay along the path (in tens of microseconds).
The following values can be used, but are not recommended, because they typically result in frequent recalculation of the topology table:

Reliability - Represents the worst reliability between the source and destination, which is based on keepalives.

Load - Represents the worst load on a link between the source and destination, which is computed based on the packet rate and the configured bandwidth of the interface.

OSPF (Open Shortest Path First) routing

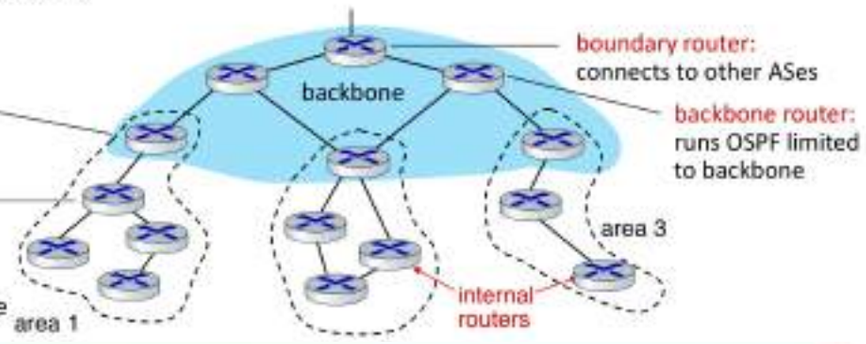
- "open": publicly available
- classic link-state
  - each router floods OSPF link-state advertisements (directly over IP rather than using TCP/UDP) to all other routers in entire AS
  - multiple link costs metrics possible: bandwidth, delay
  - each router has full topology, uses Dijkstra's algorithm to compute forwarding table
- *security:* all OSPF messages authenticated (to prevent malicious intrusion)

Hierarchical OSPF

- two-level hierarchy: local area, backbone.
  - link-state advertisements flooded only in area, or backbone
  - each node has detailed area topology; only knows direction to reach other destinations

area border routers:
"summarize" distances to destinations in own area, advertise in backbone

local routers:
- flood LS in area only
- compute routing within area
- forward packets to outside via area border router



	Interior Gateway Protocols				Exterior Gateway Protocols
	Distance Vector		Link-State		Path Vector
IPv4	RIPv2	EIGRP	OSPFv2	IS-IS	BGP-4
IPv6	RIPng	EIGRP for IPv6	OSPFv3	IS-IS for IPv6	BGP-MP

Protocol	RIP	OSPF	EIGRP
Convergence time	slow	fast	fast
VLSM	no (v2: yes)	yes	yes
Bandwidth usage	high	low	low
Resources usage	low	high	low
Multiple path support	no	yes	yes
Scalability	no	yes	yes
Patented	no	no	yes
Non-IP Protocol	no	no	yes

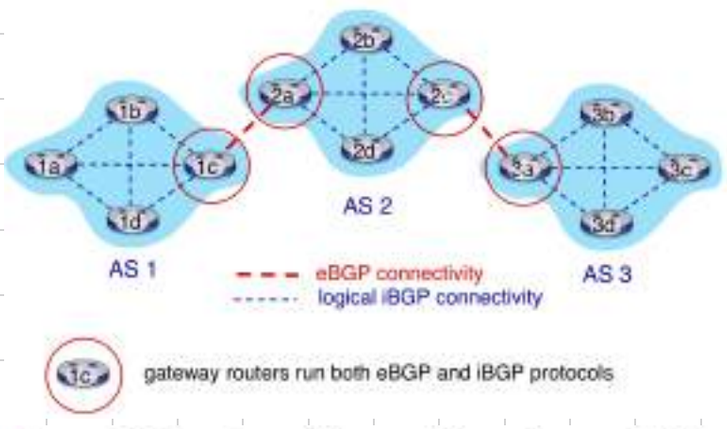
• BGP

BGP used TCP Connection

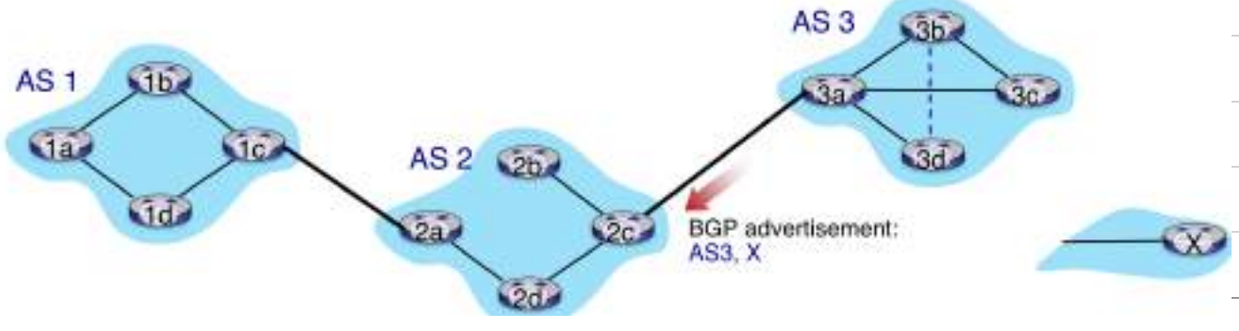
Internet inter-AS routing: BGP

- BGP (Border Gateway Protocol): *the* de facto inter-domain routing protocol
  - "glue that holds the Internet together"
- allows subnet to advertise its existence, and the destinations it can reach, to rest of Internet: *"I am here, here is who I can reach, and how"*
- BGP provides each AS a means to:
  - eBGP: obtain subnet reachability information from neighboring ASes
  - iBGP: propagate reachability information to all AS-internal routers.
  - determine "good" routes to other networks based on reachability information and *policy*

eBGP, iBGP connections



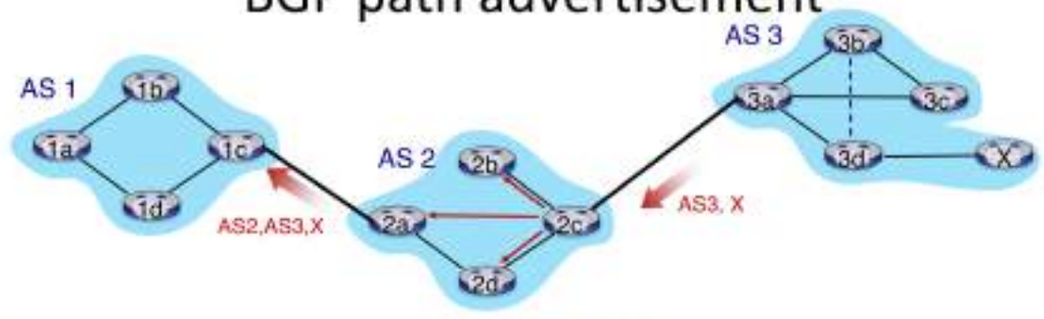
- BGP session: two BGP routers ("peers") exchange BGP messages over semi-permanent TCP connection:
  - advertising *paths* to different destination network prefixes (BGP is a "path vector" protocol)
- when AS3 gateway 3a advertises path AS3,X to AS2 gateway 2c:
  - AS3 *promises* to AS2 it will forward datagrams towards X



Path attributes and BGP routes

- BGP advertised route: prefix + attributes
  - prefix: destination being advertised
  - two important attributes:
    - AS-PATH: list of ASes through which prefix advertisement has passed
    - NEXT-HOP: indicates specific internal-AS router to next-hop AS
- policy-based routing:
  - gateway receiving route advertisement uses *import policy* to accept/decline path (e.g., never route through AS Y).
  - AS policy also determines whether to *advertise* path to other neighboring ASes

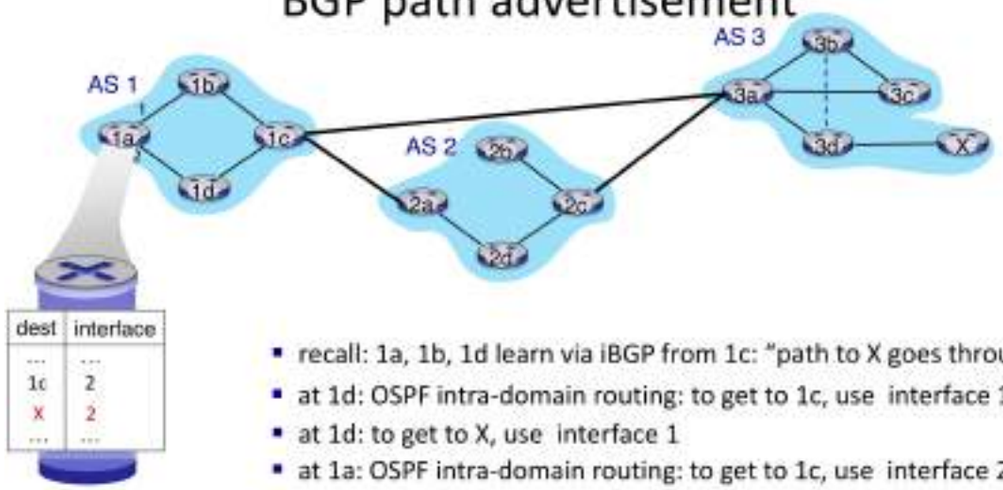
BGP path advertisement



- AS2 router 2c receives path advertisement AS3,X (via eBGP) from AS3 router 3a
- based on AS2 policy, AS2 router 2c accepts path AS3,X, propagates (via iBGP) to all AS2 routers
- based on AS2 policy, AS2 router 2a advertises (via eBGP) path AS2, AS3, X to AS1 router 1c

- BGP messages exchanged between peers over TCP connection
- BGP messages:
  - OPEN: opens TCP connection to remote BGP peer and authenticates sending BGP peer
  - UPDATE: advertises new path (or withdraws old)
  - KEEPALIVE: keeps connection alive in absence of UPDATES; also ACKs OPEN request
  - NOTIFICATION: reports errors in previous msg; also used to close connection

BGP path advertisement



- recall: 1a, 1b, 1d learn via iBGP from 1c: "path to X goes through 1c"
- at 1d: OSPF intra-domain routing: to get to 1c, use interface 1
- at 1d: to get to X, use interface 1
- at 1a: OSPF intra-domain routing: to get to 1c, use interface 2
- at 1a: to get to X, use interface 2

Why different Intra-, Inter-AS routing ?

policy:

- inter-AS: admin wants control over how its traffic routed, who routes through its network
- intra-AS: single admin, so policy less of an issue

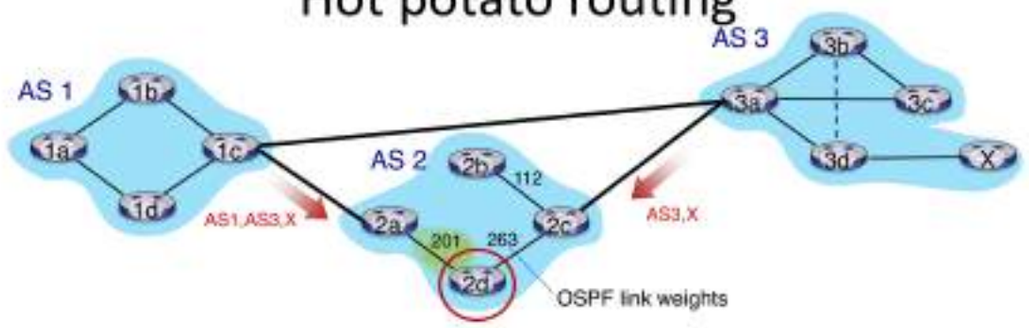
scale:

- hierarchical routing saves table size, reduced update traffic

performance:

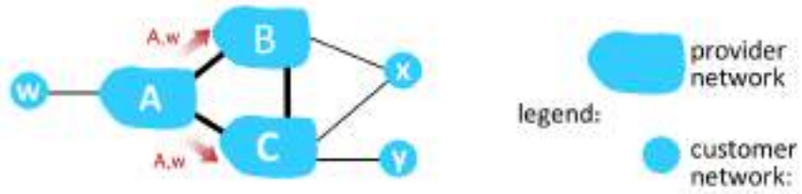
- intra-AS: can focus on performance
- inter-AS: policy dominates over performance

Hot potato routing



- 2d learns (via iBGP) it can route to X via 2a or 2c
- hot potato routing: choose local gateway that has least *intra-domain* cost (e.g., 2d chooses 2a, even though more AS hops to *X*): don't worry about inter-domain cost!

BGP: achieving policy via advertisements



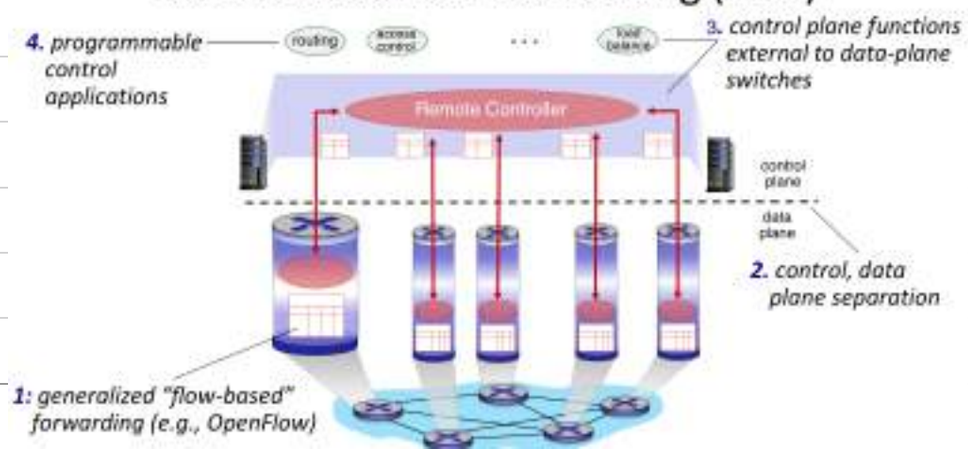
ISP only wants to route traffic to/from its customer networks (does not want to carry transit traffic between other ISPs – a typical "real world" policy)

- A advertises path Aw to B and to C
- B *chooses not to advertise* BAw to C!
  - B gets no "revenue" for routing CBAw, since none of C, A, w are B's customers
  - C does *not* learn about CBAw path
- C will route CAw (not using B) to get to w
- A,B,C are provider networks
- x,w,y are customer (of provider networks)
- x is dual-homed: attached to two networks
- *policy to enforce:* x does not want to route from B to C via x
  - .. so x will not advertise to B a route to C

- router may learn about more than one route to destination AS, selects route based on:
  1. local preference value attribute: policy decision
  2. shortest AS-PATH
  3. closest NEXT-HOP router: hot potato routing
  4. additional criteria

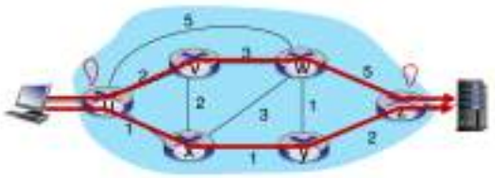
• SDN : Software-defined Networking

Software defined networking (SDN)



Why a *logically centralized* control plane?

- easier network management: avoid router misconfigurations, greater flexibility of traffic flows
- table-based forwarding (recall OpenFlow API) allows "programming" routers
  - centralized "programming" easier: compute tables centrally and distribute
  - distributed "programming" more difficult: compute tables as result of distributed algorithm (protocol) implemented in each-and-every router
- open (non-proprietary) implementation of control plane



← Difficulties of Traditional Routing

Q: what if w wants to route blue and red traffic differently from w to z?

A: can't do it (with destination-based forwarding, and LS, DV routing)

We learned that generalized forwarding and SDN can be used to achieve *any* routing desired

Q: what if network operator wants u-to-z traffic to flow along *uvwz*, rather than *uxyz*?

A: need to re-define link weights so traffic routing algorithm computes routes accordingly (or need a new routing algorithm)!

link weights are difficult to control

Q: what if network operator wants to split u-to-z traffic along uvwz *and* uxyz (load balancing)?

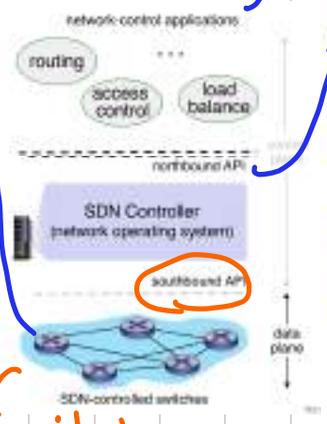
A: can't do it (or need a new routing algorithm)

network-control apps:

- "brains" of control: implement control functions using lower-level services, API provided by SDN controller
- *unbundled:* can be provided by 3rd party: distinct from routing vendor, or SDN controller

Data-plane switches:

- fast, simple, commodity switches implementing generalized data-plane forwarding in hardware
- flow (forwarding) table computed, installed under controller supervision
- API for table-based switch control (e.g., OpenFlow)
  - defines what is controllable, what is not
- protocol for communicating with controller (e.g., OpenFlow)



switch

SDN controller (network OS):

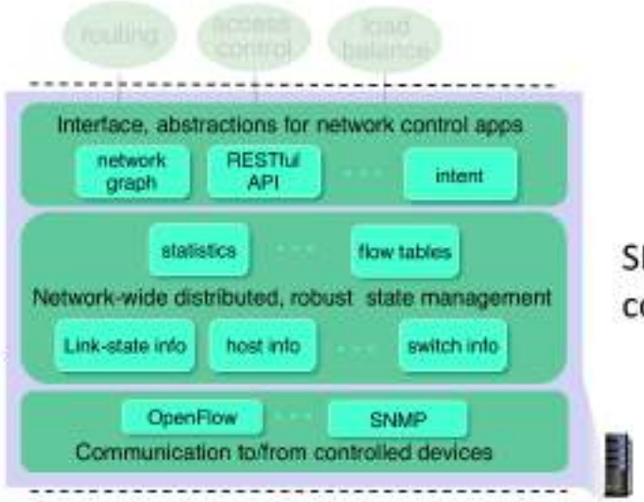
- maintain network state information
- interacts with network control applications "above" via northbound API
- interacts with network switches "below" via southbound API
- implemented as distributed system for performance, scalability, fault-tolerance, robustness

Components of SDN controller

interface layer to network control apps: abstractions API

network-wide state management : state of networks links, switches, services: a *distributed database*

communication: communicate between SDN controller and controlled switches

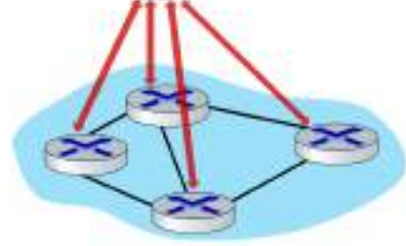


SDN controller

OpenFlow protocol

- operates between controller, switch
- TCP used to exchange messages
  - optional encryption
- three classes of OpenFlow messages:
  - controller-to-switch
  - asynchronous (switch to controller)
  - symmetric (misc.)
- distinct from OpenFlow API
  - API used to specify generalized forwarding actions

OpenFlow Controller



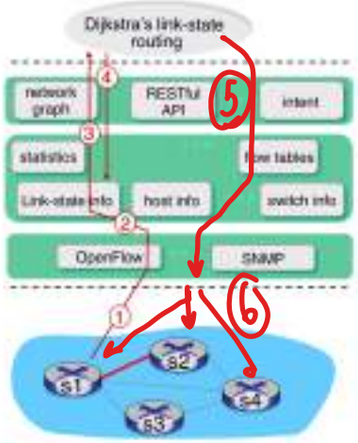
Key controller-to-switch messages

- *features:* controller queries switch features, switch replies
- *configure:* controller queries/sets switch configuration parameters
- *modify-state:* add, delete, modify flow entries in the OpenFlow tables
- *packet-out:* controller can send this packet out of specific switch port

Key switch-to-controller messages

- *packet-in:* transfer packet (and its control) to controller. See packet-out message from controller
- *flow-removed:* flow table entry deleted at switch
- *port status:* inform controller of a change on a port.

SDN: control/data plane interaction example



① S1, experiencing link failure uses OpenFlow port status message to notify controller

② SDN controller receives OpenFlow message, updates link status info

③ Dijkstra's routing algorithm application has previously registered to be called when ever link status changes. It is called.

④ Dijkstra's routing algorithm access network graph info, link state info in controller, computes new routes

⑤ link state routing app interacts with flow-table-computation component in SDN controller, which computes new flow tables needed

⑥ controller uses OpenFlow to install new tables in switches that need updating

→ another controller: ODL : interconnect internal & external apps
ONOS : control apps seperate from controler etc.

SDN: selected challenges

- hardening the control plane: dependable, reliable, performance-scalable, secure distributed system
  - robustness to failures: leverage strong theory of reliable distributed system for control plane
  - dependability, security?
- networks, protocols meeting mission-specific requirements
  - e.g., real-time, ultra-reliable, ultra-secure
- Internet-scaling: beyond a single AS

• ICMP : Internet control message Protocol

- used by hosts and routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP messages 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	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

TTL: time to live

- source sends sets of UDP segments to destination
  - 1st set has TTL =1, 2nd set has TTL=2, etc.
- datagram in *n*th set arrives to nth router:
  - router discards datagram and sends source ICMP message (type 11, code 0 = TTL Expired)
  - ICMP message possibly includes name of router & IP address
- when ICMP message arrives at source: record RTTs

stopping criteria:
- UDP segment eventually arrives at destination host
- destination returns ICMP "port unreachable" message (type 3, code 3)
- source stops

• Network Management.

"Network management includes the deployment, integration and coordination of the hardware, software, and human elements to monitor, test, poll, configure, analyze, evaluate, and control the network and element resources to meet the real-time, operational performance, and Quality of Service requirements at a reasonable cost."

Managing server: application, typically with network managers (humans) in the loop

Network management protocol: used by managing server to query, configure, manage device; used by devices to inform managing server of data, events.

Managed device: equipment with manageable, configurable hardware, software components

Data: device "state" configuration data, operational data, device statistics

Network operator approaches to management

MIB: Management info base

CLI (Command Line Interface)

- operator issues (types, scripts) direct to individual devices (e.g., vis ssh)

SNMP/MIB

- operator queries/sets devices data (MIB) using Simple Network Management Protocol (SNMP)

NETCONF/YANG

- more abstract, network-wide, holistic
- emphasis on multi-device configuration management.
- YANG: data modeling language
- NETCONF: communicate YANG-compatible actions/data to/from/among remote devices

Selected NETCONF Operations

NETCONF	Operation Description
<get-config>	Retrieve all or part of a given configuration. A device may have multiple configurations.
<get>	Retrieve all or part of both configuration state and operational state data.
<edit-config>	Change specified (possibly running) configuration at managed device. Managed device <rpc-reply> contains <ok> or <rpcerror> with rollback.
<lock>, <unlock>	Lock (unlock) configuration datastore at managed device (to lock out NETCONF, SNMP, or CLIs commands from other sources).
<create-subscription>, <notification>	Enable event notification subscription from managed device

SNMP protocol

Two ways to convey MIB info, commands:

Message type	Function
GetRequest GetNextRequest GetBulkRequest	manager-to-agent: "get me data" (data instance, next data in list, block of data).
SetRequest	manager-to-agent: set MIB value
Response	Agent-to-manager: value, response to Request
Trap	Agent-to-manager: inform manager of exceptional event

Sample NETCONF RPC message

```
01 <?xml version="1.0" encoding="UTF-8"?>
02 <rpc message-id="101"  note message id
03   xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
04   <edit-config>    change a configuration
05     <target>
06       <running/> change the running configuration
07     </target>
08     <config>
09       <top xmlns="http://example.com/schema/
         1.2/config">
10         <interface>
11           <name>Ethernet0/0</name>  change MTU of Ethernet 0/0 interface to 1500
12           <mtu>1500</mtu>
13         </interface>
14       </top>
15     </config>
16   </edit-config>
17 </rpc>
```

SNMP: Management Information Base (MIB)

- managed device's operational (and some configuration) data

- gathered into device MIB module
  - 400 MIB modules defined in RFC's; many more vendor-specific MIBs
  - Structure of Management Information (SMI): data definition language

YANG

- data modeling language used to specify structure, syntax, semantics of NETCONF network management data
  - built-in data types, like SMI
- XML document describing device, capabilities can be generated from YANG description
- can express constraints among data that must be satisfied by a valid NETCONF configuration
  - ensure NETCONF configurations satisfy correctness, consistency constraints

NETCONF overview

- goal: actively manage/configure devices network-wide
- operates between managing server and managed network devices
  - actions: retrieve, set, modify, activate configurations
  - atomic-commit actions over multiple devices
  - query operational data and statistics
  - subscribe to notifications from devices
- remote procedure call (RPC) paradigm
  - NETCONF protocol messages encoded in XML
  - exchanged over secure, reliable transport (e.g., TLS) protocol

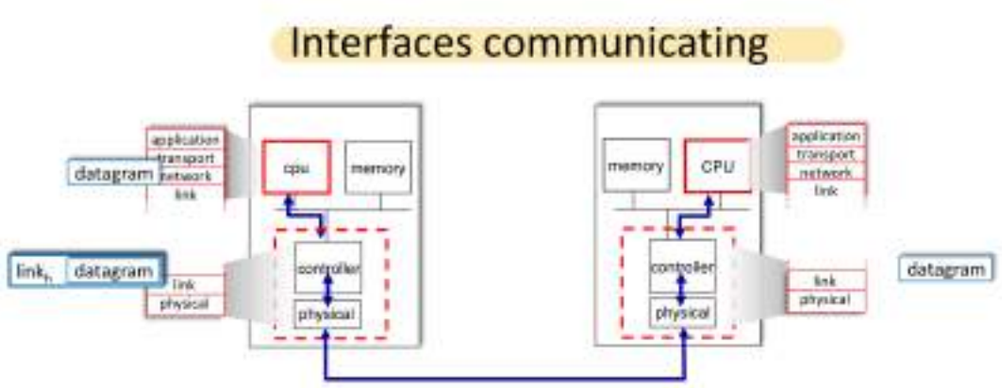
* Data Link Layer : data transfer b/w neighboring network elements

- has responsibility of transferring datagram from one node to physically adjacent node over a link

- transferred by different protocols → provides different services // may/may not provide reliable data transfer

  - Services → framing : encap. datagram into frame
    MAC Address : identify src/dest.
    → Reliable delivery b/w adjacent nodes
    used on low bit-error links
    → flow control
    → error detection : detect
    → error correction : identify & correct @ dest.
    → half-duplex / full-duplex

- link layer is implemented in every host in NIC : Network Interface Card.



- Error Detection
  - error detection & correction bits : "Redundancy" bit.
  - not 100% reliable b/c some may miss the error but rarely

- Parity checking

  parity bit.

  - even parity: 1 1 0 0 1 1
  - → single bit parity: 1 0 1 0 0 0

  → 2D-Parity / Block parity
  → row parity & column parity

				row parity
1	0	1	1	1
1	1	0	1	1
1	0	0	0	1
0	1	1	0	0
1	0	0	0	1

column parity

- Checksum → (1's complement)

sender:
- treat contents of UDP segment (including UDP header fields and IP addresses) as sequence of 16-bit integers
- checksum: addition (one's complement sum) of segment content
- checksum value put into UDP checksum field

receiver:
- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - not equal - error detected
  - equal - no error detected. *But maybe errors nonetheless?* More later ....

P2P: point-to-point

• CRC: Cyclic Redundancy Check.

given data bits m bits
G bit pattern of r+1 bits *required.
⇒ redundancy of r bits
⇒ transmitted data has m+r bits

x	y	x⊕y
0	0	0
0	1	1
1	0	1
1	1	0

• use long division + XOR operation
M = "10011011" G = "1001"

find R

```
          1 0 0 0 1 0 1 0
      ---------------------
1001 ) 1 0 0 1 1 0 1 1 0 0 0
       1 0 0 1
       -------
         1 0 1 1
         1 0 0 1
         -------
           1 0 0 0
           1 0 0 1
           -------
             R = 0 1 0
```

Check error

```
          1 0 0 0 1 0 1 0
      ---------------------
1001 ) 1 0 0 1 1 0 1 1 0 1 0
       1 0 0 1
       -------
         1 0 1 1
         1 0 0 1
         -------
           1 0 0 1
           1 0 0 1
           -------
             0 0 0
```

no error! ⇐ ⇒ ✓

• Multiple access links, protocols

2 types of links: P2P / Broadcast. ⟶
4 btw switch & host shared wire/med.

- single shared broadcast channel
- interference when 2+ simultaneous transmissions
- collision when 2+ nodes receives simultaneous

multiple access protocol

- distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!
  - no out-of-band channel for coordination

An ideal multiple access protocol

given: Multiple Access Channel (MAC) of rate *R* bps

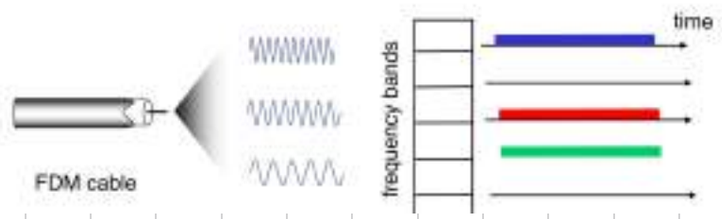
1. when one node wants to transmit, it can send at rate *R*.
2. when M nodes want to transmit, each can send at average rate *R/M*
3. fully decentralized:
   - no special node to coordinate transmissions
   - no synchronization of clocks, slots
4. simple

multiple access channel

- MAC Classes
  - Partitioning : divide in to smaller pieces
    → "Time" / "frequency" / "code"
  - Random access : recover from collision / no divided channel
  - Taking Turns : nodes take turns / more to send, more time.

- Channel Partitioning
  - TDMA: Time - Division

  - FDMA : frequency - Division
    - channel spectrum divided into frequency bands
    - each station assigned fixed frequency band
    - unused transmission time in frequency bands go idle
    - example: 6-station LAN, 1,3,4 have packet to send, frequency bands 2,5,6 idle



- Random access
  - transmit at full channel, no coordination b/w nodes
  - Specified how to
    → detect collisions & recover from collision

    → Slotted ALOHA
      - obtain frame → transmit frame
        → if no collision : transmit next frame
        → if collision: retransmit in subsequent slot w/ probability p.

Slotted ALOHA

Pros:
- single active node can continuously transmit at full rate of channel
- highly decentralized: only slots in nodes need to be in sync
- simple

Cons:
- collisions, wasting slots
- idle slots
- clock synchronization

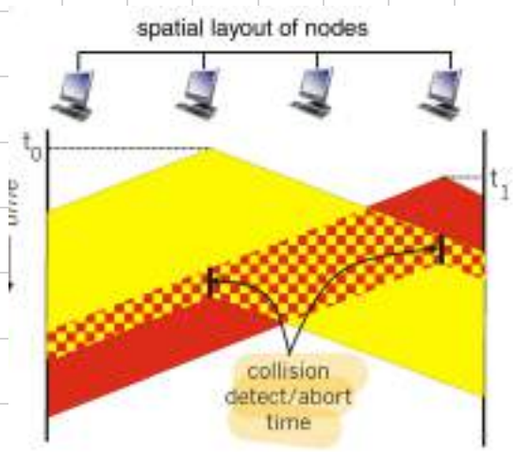
→ Pure ALOHA : frame arrive: transmit imm.
collision ? if no synchronization

→ CSMA : Carrier Sense Multiple Access

Simple CSMA: listen before transmit:
- if channel sensed idle: transmit entire frame
- if channel sensed busy: defer transmission
- human analogy: don't interrupt others!

CSMA/CD: CSMA with *Collision Detection*
- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage
- collision detection easy in wired, difficult with wireless
- human analogy: the polite conversationalist

- collision: entire packet transmission time wasted
  - distance & propagation delay play role in in determining collision probability



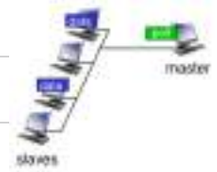
$$efficiency = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

↳ go to 1 when
$t_{prop} \to 0$, $t_{trans} \to \infty$

Ethernet CSMA/CD algorithm

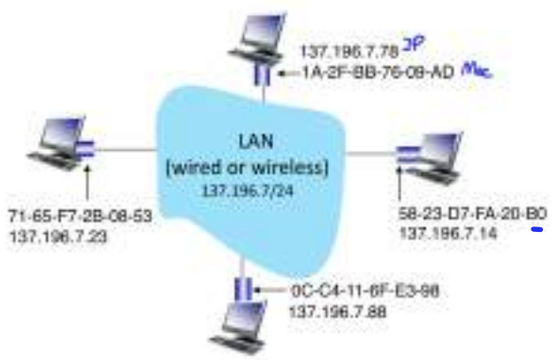
1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel:
   if idle: start frame transmission.
   if busy: wait until channel idle, then transmit
3. If NIC transmits entire frame without collision, NIC is done with frame !
4. If NIC detects another transmission while sending: abort, send jam signal
5. After aborting, NIC enters *binary (exponential) backoff:*
   - after *m*th collision, NIC chooses *K* at random from $\{0,1,2,\ldots,2^m-1\}$. NIC waits $K \cdot 512$ bit times, returns to Step 2
   - more collisions: longer backoff interval

→ Taking Turn



- Taking turns
  - polling from central site, token passing
  - Bluetooth, FDDI, token ring

- MAC Address (hexadecimal) : 12-34-56-78-9A-BC
  ↳ 48-bit MAC Address, fixed, unique
  ↳ locally jet frame from one interface to another physically-connected interface



→ MAC addr has "portability" → move from one another w/o changing addr.

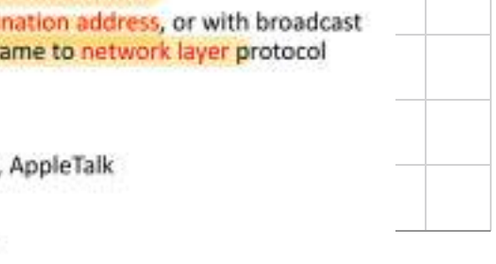
- ARP : Address Resolution protocol.
  ↳ determine interface IP addr, knowing IP addr
  ↳ window: show ARP table using "arp -a"

- ARP Table: mappings b/w IP & MAC address
  ↳ TTL (time-to-live): how long until it forgot.

① Broadcast ARP query ② Replies using ARP reply

- Ethernet
  → wired LAN
  → unreliable (no ACKs/NAKs)
  → & Connectionless (no handshaking)
  → using unslotted CSMA/CD

"Topology"
- Bus
- Ring
- Star/Switched
- Mesh



- frame structure

preamble:
- used to synchronize receiver, sender clock rates

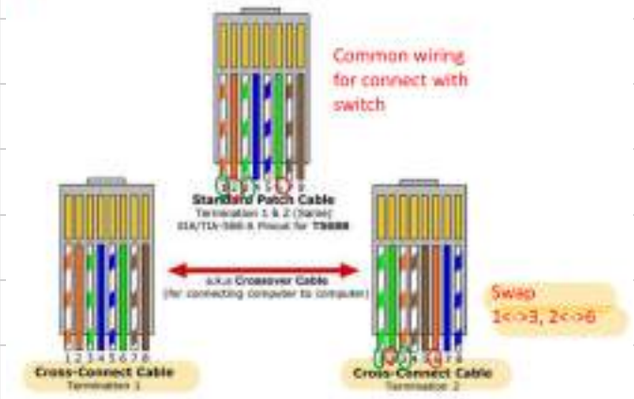
- addresses: 6 byte source, destination MAC addresses
  - if adapter receives frame with matching destination address, or with broadcast address (e.g., ARP packet), it passes data in frame to network layer protocol
  - otherwise, adapter discards frame
- type: indicates higher layer protocol
  - mostly IP but others possible, e.g., Novell IPX, AppleTalk
  - used to demultiplex up at receiver
- CRC: cyclic redundancy check at receiver
  - error detected: frame is dropped

802.3 Ethernet Standard

802.3 Ethernet standards: link & physical layers

Original IEEE Standard	Shorthand Name	Informal Names	Speed	Typical Cabling
802.3i	10BASE-T	Ethernet	10 Mbps	UTP (Unshielded Twisted Pair)
802.3u	100BASE-T	Fast Ethernet	100 Mbps	UTP
802.3z	1000BASE-X	Gigabit Ethernet, GigE	1000 Mbps (1 Gbps)	Fiber
802.3ab	1000BASE-T	Gigabit Ethernet, GigE	1000 Mbps (1 Gbps)	UTP
802.3ae	10GBASE-X	10 GigE	10 Gbps	Fiber (Fiber Optic Cable)
802.3an	10GBASE-T	10 GigE	10 Gbps	UTP
802.3ba	40GBASE-X	40 GigE	40 Gbps	Fiber
802.3ba	100GBASE-X	100 GigE	100 Gbps	Fiber

UTP: Unshielded Twisted Pair



UTP Categories - Copper Cable

UTP Category	Data Rate	Max. Length	Cable Type	Application
CAT1	Up to 1Mbps	-	Twisted Pair	Old Telephone Cable
CAT2	Up to 4Mbps	-	Twisted Pair	Token Ring Networks
CAT3	Up to 10Mbps	100m	Twisted Pair	Token Rink & 10BASE-T Ethernet
CAT4	Up to 16Mbps	100m	Twisted Pair	Token Ring Networks
CAT5	Up to 100Mbps	100m	Twisted Pair	Ethernet, FastEthernet, Token Ring
CAT5e	Up to 1 Gbps	100m	Twisted Pair	Ethernet, FastEthernet, Gigabit Ethernet
CAT6	Up to 10Gbps	100m	Twisted Pair	GigabitEthernet, 10G Ethernet (55 meters)
CAT6a	Up to 10Gbps	100m	Twisted Pair	GigabitEthernet, 10G Ethernet (55 meters)
CAT7	Up to 10Gbps	100m	Twisted Pair	GigabitEthernet, 10G Ethernet (100 meters)

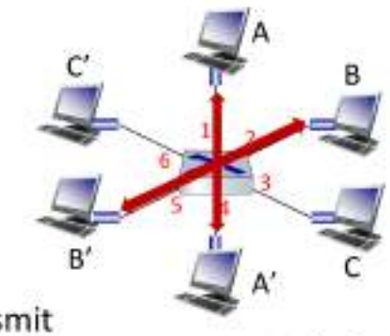
Commonly used.

• Switch

- link-layer device
  - → store & forward ethernet frame
  - → selectively forward frame using CSMA/CD
- transparent : unaware of presence of switches
- plug & play : self-learning, no need to config switch.

Switch: multiple simultaneous transmissions

- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, so:
  - no collisions; full duplex
  - each link is its own collision domain
- switching: A-to-A' and B-to-B' can transmit simultaneously, without collisions
- but A-to-A' and C to A' can *not* happen simultaneously



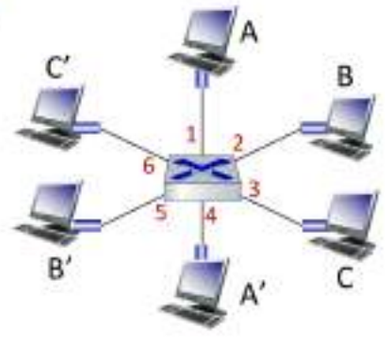
switch with six interfaces (1,2,3,4,5,6)

Switch forwarding table

Q: how does switch know A' reachable via interface 4, B' reachable via interface 5?

A: each switch has a switch table, each entry:

- (MAC address of host, interface to reach host, time stamp)
- looks like a routing table!

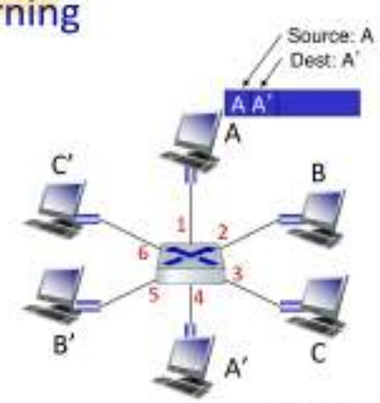


Q: how are entries created, maintained in switch table?

- something like a routing protocol?

Switch: self-learning

- switch *learns* which hosts can be reached through which interfaces
  - when frame received, switch "learns" location of sender: incoming LAN segment
  - records sender/location pair in switch table



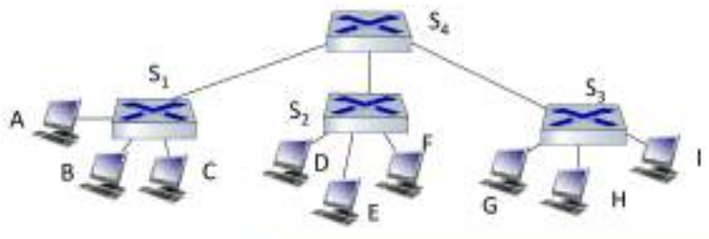
MAC addr	Interface	TTL
A	1	60

Switch table (initially empty)

Switch: frame filtering/forwarding

when frame received at switch:

1. record incoming link, MAC address of sending host
2. index switch table using MAC destination address
3. if entry found for destination
   then {
   if destination on segment from which frame arrived
   then drop frame
   else forward frame on interface indicated by entry
   }
   else flood /* forward on all interfaces except arriving interface */



Q: sending from A to G - how does S_1 know to forward frame destined to G via S_4 and S_3?

- *A:* self learning! (works exactly the same as in single-switch case!)

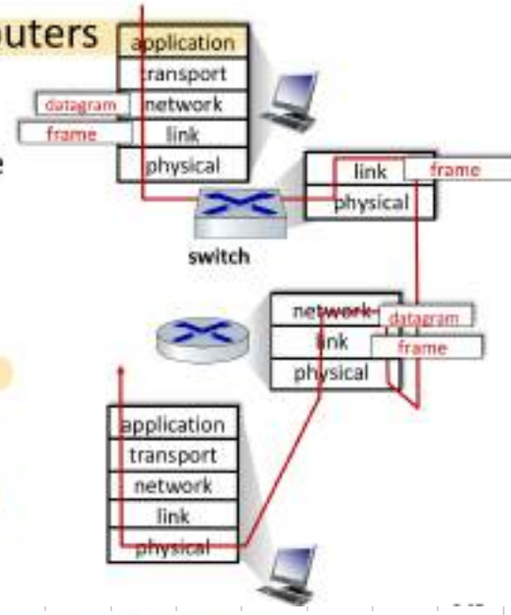
Switches vs. routers

both are store-and-forward:

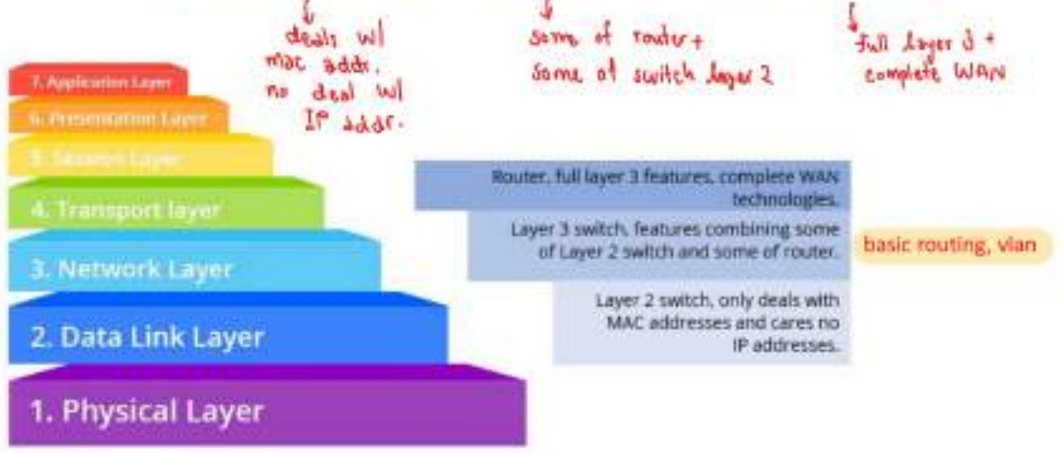
- *routers*: network-layer devices (examine network-layer headers)
- *switches:* link-layer devices (examine link-layer headers)

both have forwarding tables:

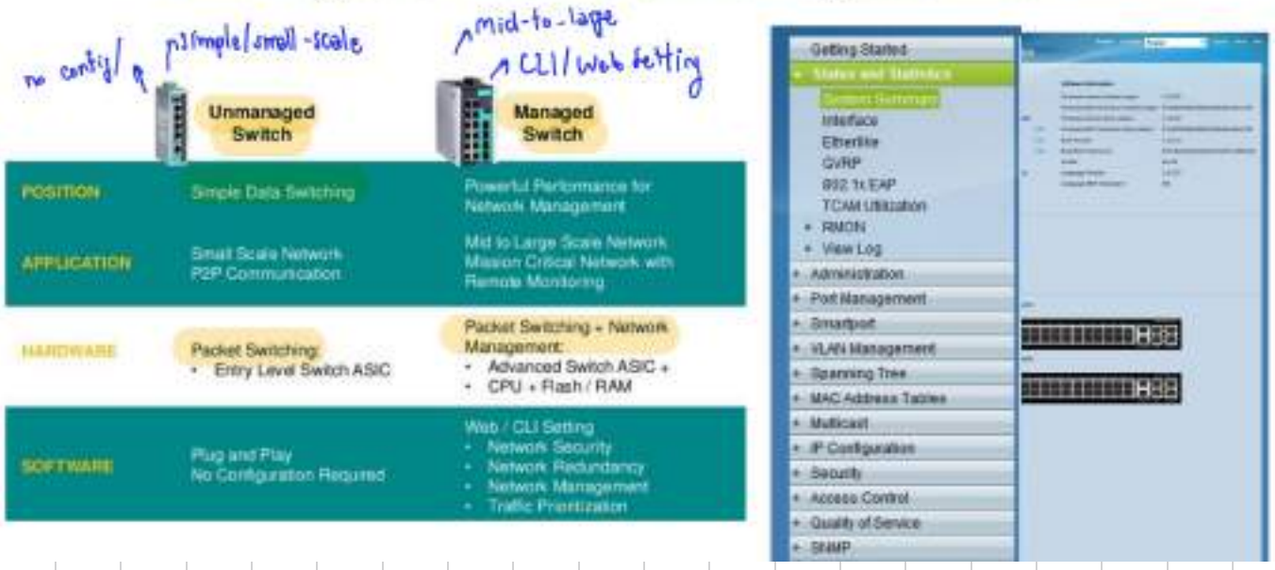
- *routers:* compute tables using routing algorithms, IP addresses
- *switches:* learn forwarding table using flooding, learning, MAC addresses



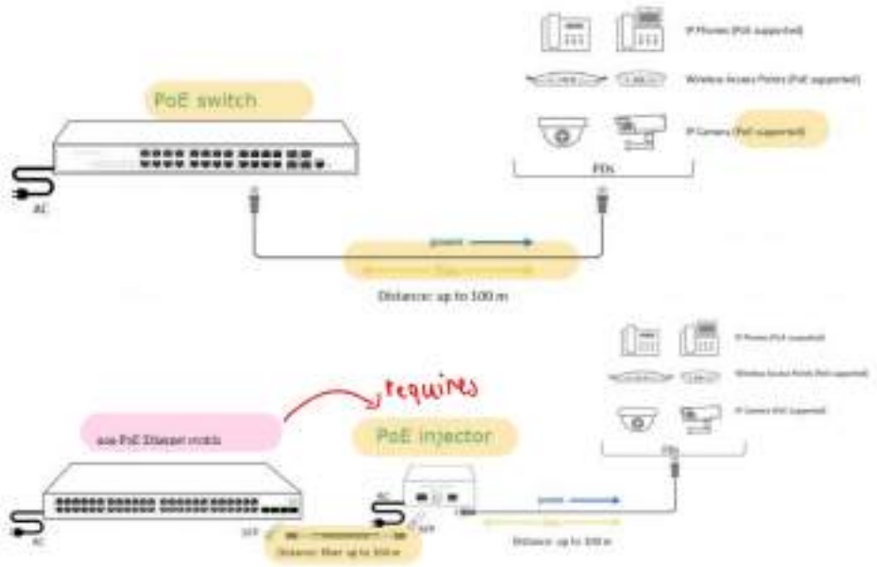
Switch Layer 2 vs Switch Layer 3 vs Router



Managed Switch vs Unmanaged Switch



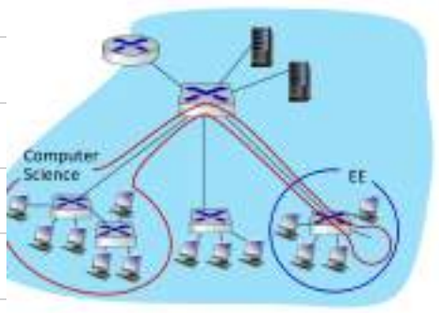
Switch with POE (Power over Ethernet)



- VLANs

Virtual LANs (VLANs): motivation

Q: what happens as LAN sizes scale, users change point of attachment?



single broadcast domain:

- *scaling:* all layer-2 broadcast traffic (ARP, DHCP, unknown MAC) must cross entire LAN
- efficiency, security, privacy, efficiency issues

administrative issues:

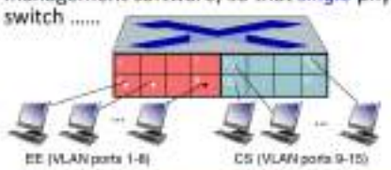
- CS user moves office to EE - *physically* attached to EE switch, but wants to remain *logically* attached to CS switch

Port-based VLANs

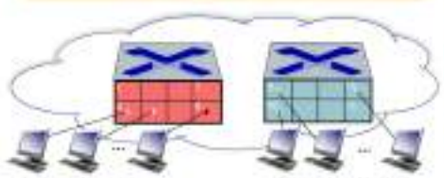
Virtual Local Area Network (VLAN)

switch(es) supporting VLAN capabilities can be configured to define multiple *virtual* LANS over single physical LAN infrastructure.

port-based VLAN: switch ports grouped (by switch management software) so that *single* physical switch .....

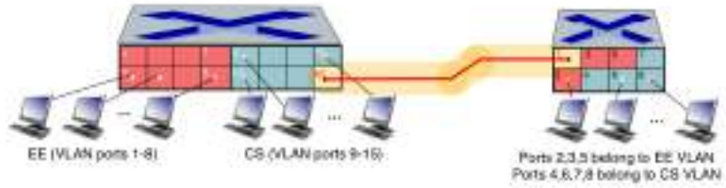


... operates as multiple virtual switches



- traffic isolation: frames to/from ports 1-8 can only reach ports 1-8
  - can also define VLAN based on MAC addresses of endpoints, rather than switch port
- dynamic membership: ports can be dynamically assigned among VLANs
- forwarding between VLANS: done via routing (just as with separate switches)
  - in practice vendors sell combined switches plus routers (Switch L3)

VLANS spanning multiple switches



trunk port: carries frames between VLANS defined over multiple physical switches

- frames forwarded within VLAN between switches can't be vanilla 802.1 frames (must carry VLAN ID info)
- 802.1q protocol adds/removed additional header fields for frames forwarded between trunk ports

802.1Q VLAN

- Max VLAN limit depends on switch model. (max 4094, common 256)
- VLAN1 is also called management VLAN.
- A layer 3 device is required for Inter VLAN
- DHCP with multiple pools support may required for multi-VLAN
- Need managed switch

- MPLS: Multiprotocol label switching

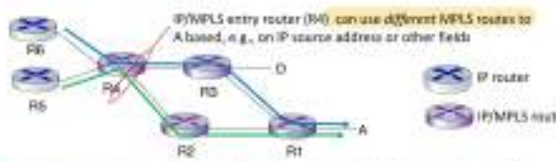
- Why MPLS
  - Use switch instead routing
  - Virtual Private Network
  - QOS
  - Scalable (million endpoints)
  - Fast Speed

- goal: high-speed IP forwarding among network of MPLS-capable routers, using fixed length label (instead of shortest prefix matching)
  - faster lookup using fixed length identifier
  - borrowing ideas from Virtual Circuit (VC) approach
  - but IP datagram still keeps IP address!

MPLS capable routers

- a.k.a. label-switched router
- forward packets to outgoing interface based only on label value (*don't inspect IP address*)
  - MPLS forwarding table distinct from IP forwarding tables
- *flexibility:* MPLS forwarding decisions can *differ* from those of IP
  - use destination *and* source addresses to route flows to same destination differently (traffic engineering)
  - re-route flows quickly if link fails: pre-computed backup paths

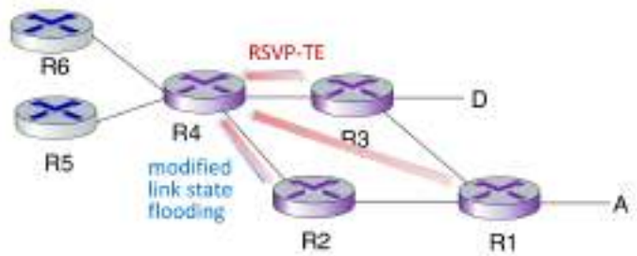
MPLS versus IP paths



- IP routing: path to destination determined by destination address alone
- MPLS routing: path to destination can be based on source *and* destination address
  - flavor of generalized forwarding (MPLS 10 years earlier)
  - *fast reroute:* precompute backup routes in case of link failure

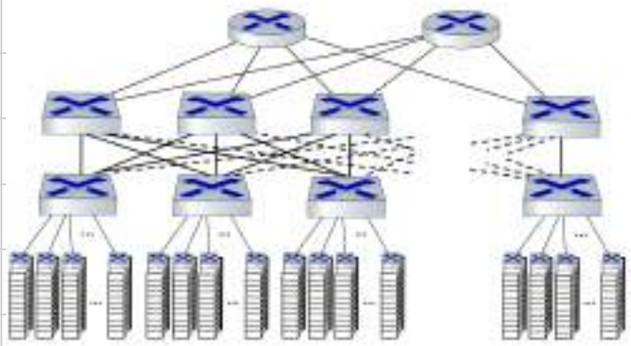
MPLS signaling

- modify OSPF, IS-IS link-state flooding protocols to carry info used by MPLS routing:
  - e.g., link bandwidth, amount of "reserved" link bandwidth
- entry MPLS router uses RSVP-TE signaling protocol to set up MPLS forwarding at downstream routers



- Data center Networks
  - challenge:- multiple app w/ massive number of client
    - reliability
    - managing & balancing load.

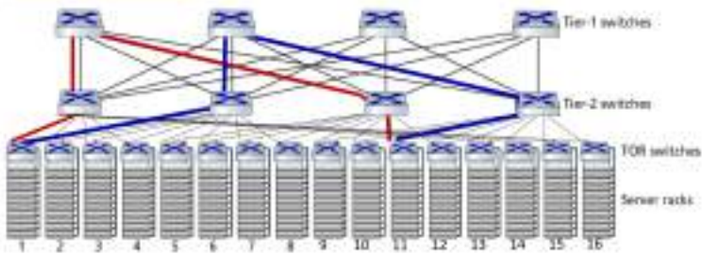
Datacenter networks: network elements



- **Border routers**
  - connections outside datacenter
- **Tier-1 switches**
  - connecting to ~16 T-2s below
- **Tier-2 switches**
  - connecting to ~16 TORs below
- **Top of Rack (TOR) switch**
  - one per rack
  - 40-100Gbps Ethernet to blades
- **Server racks**
  - 20- 40 server blades: hosts

Datacenter networks: multipath

- rich interconnection among switches, racks:
  - increased throughput between racks (multiple routing paths possible)
  - increased reliability via redundancy



two disjoint paths highlighted between racks 1 and 11

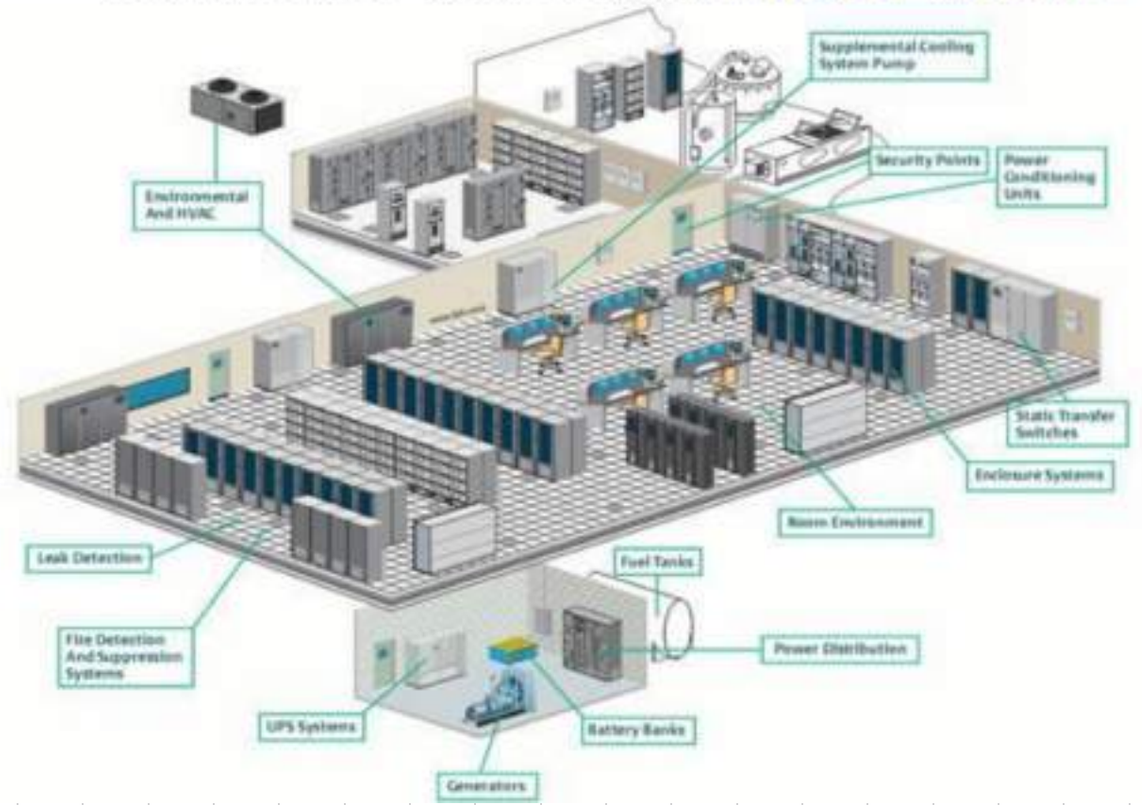
load balancer: application-layer routing

- receives external client requests
- directs workload within data center
- returns results to external client (hiding data center internals from client)

Datacenter networks: protocol innovations

- **link layer:**
  - RoCE: remote DMA (RDMA) over Converged Ethernet
- **transport layer:**
  - ECN (explicit congestion notification) used in transport-layer congestion control (DCTCP, DCQCN)
  - experimentation with hop-by-hop (backpressure) congestion control
- **routing, management:**
  - SDN widely used within/among organizations' datacenters
  - place related services, data as close as possible (e.g., in same rack or nearby rack) to minimize tier-2, tier-1 communication

Datacenter networks: Infrastructure



A day in life of web request.

Scenario: visit reg.siit.tu.ac.th using laptop in campus network

① Connect to the Internet

↳ obtain IP address when connected to network.

* DHCP : laptop need its IP address, addr of first-hop router, addr of DNS server.

× DHCP Request (transport: UDP → network: IP → Datalink: 802.3 Ethernet)

encap. in UDP → IP → 802.3 Ethernet

broadcast on LAN, received at DHCP server

DHCP server create DHCP Acknowledge

contain IP of client, first-hop router, DNS server

② Where's mac addr. of interface router

* ARP : need mac addr of router

ARP broadcast → received by router → ARP reply w/ mac addr → client know

③ Where's web server addr.

* DNS : DNS query "where's is reg.siit.tu.ac.th"

encap in UDP, IP, 802.3 Eth.

forward to 1st hop router → to outside → DNS server

DNS reply w/ IP addr of regsiit.tu.ac.th

④ Open Webpage Now!

4.1) Establish connection

* TCP : open TCP socket to web server

↳ 3-way handshake ♥

↳ TCP SYN → SYNACK → ESTAB

↳ connection opened!

4.2) Send HTTP Request

* HTTP : send request to TCP Socket

* HTTP Request

↳ encap. in TCP / IP / 802.3 Eth

↳ routed to reg.siit.tu.ac.th

↳ web server reply w/ HTTP reply

↳ encap back in TCP/IP/802.3 Eth

then routed back to client.

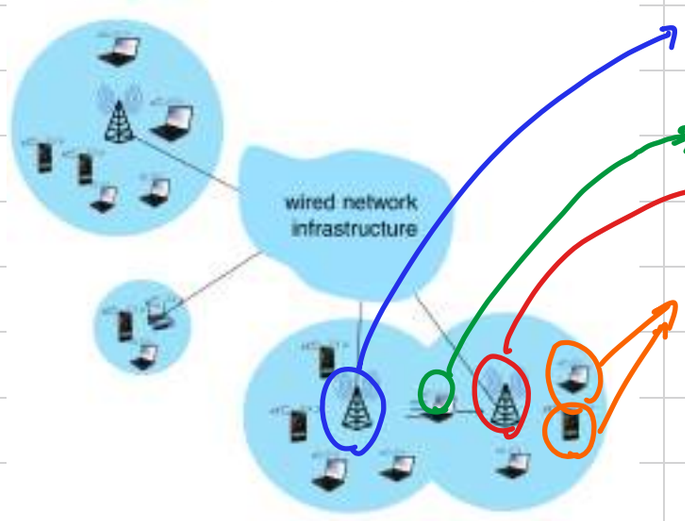
reply w/ webpage

×

Wireless & Mobile Network.

→ challenges ① Wireless : communicate over wireless link
② Mobility : changes point of attachment.

• Elements in wireless network

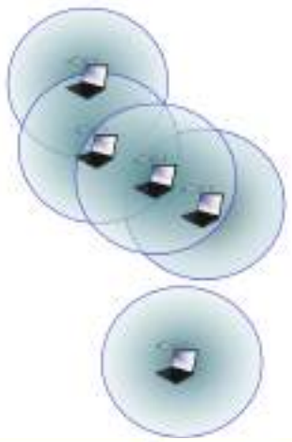


→ infrastructure node : handoff

→ wireless link : connect to base station

→ base station : connect to wired network

→ hosts
(maybe stationary / mobile)



ad hoc mode
- no base stations
- nodes can only transmit to other nodes within link coverage
- nodes organize themselves into a network: route among themselves

	single hop	multiple hops
infrastructure (e.g., APs)	host connects to base station (WiFi, cellular) which connects to larger Internet	host may have to relay through several wireless nodes to connect to larger Internet: *mesh net*
no infrastructure	no base station, no connection to larger Internet (Bluetooth, ad hoc nets)	no base station, no connection to larger Internet. May have to relay to reach other a given wireless node MANET, VANET

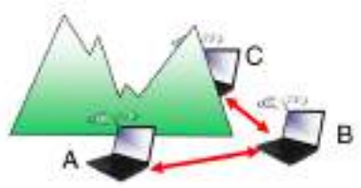
• Characteristics

• differences from wired link.
→ decreased signal strength (path loss)
→ interference
→ multipath propagation

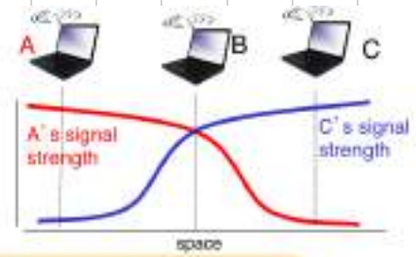
• SNR: Signal-to-Noise Ratio
→ larger SNR, the better ability to extract signal from noise

→ TRADEOFF : SNR vs BER (bit error rate)
→ physical layer
→ power ↑, SNR ↑, BER ↓
→ SNR
→ choose physical layer that meets BER requirement, giving highest throughput.

• Multiple wireless sender



Hidden terminal problem
- B, A hear each other
- B, C hear each other
- A, C can not hear each other means A, C unaware of their interference at B



Signal attenuation:
- B, A hear each other
- B, C hear each other
- A, C can not hear each other interfering at B

Wireless LAN: standard IEEE 802.11

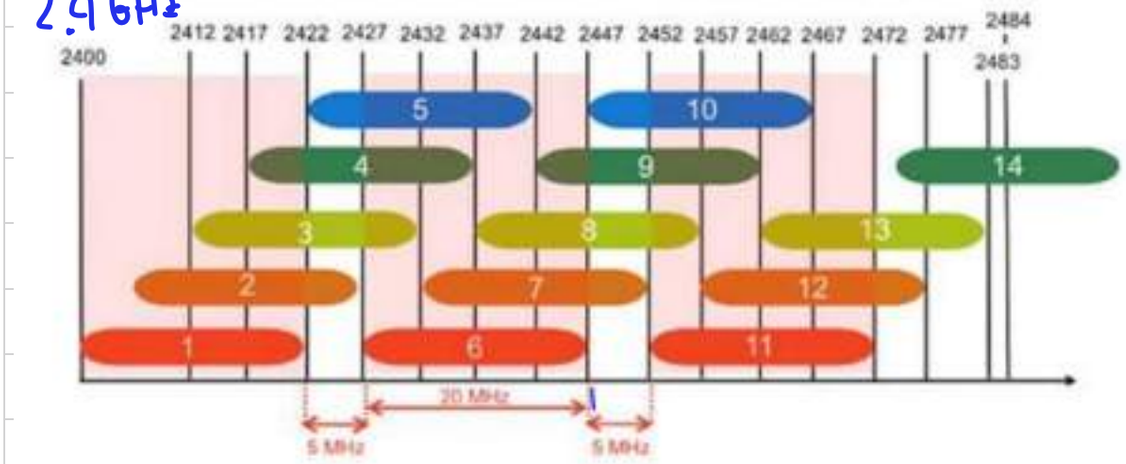
WLAN Frequency Bands	
2.4 GHz	5 GHz
Slow Data Transmission	Fast Data Transmission
Covers Long Distance	Covers Short Distance
14 Channels	23 Channels
Overlapping Channels	No Overlapping
IEEE 802.11b IEEE 802.11g IEEE 802.11ax	IEEE 802.11a IEEE 802.11n IEEE 802.11ac IEEE 802.11ax

IEEE 802.11b | 802.11g | 802.11n | 802.11ac | 802.11ax | 802.11be

Wi Fi: Wi-Fi | Wi-Fi 4 | Wi-Fi 5 | Wi-Fi 6/6E | Wi-Fi 7 (?)

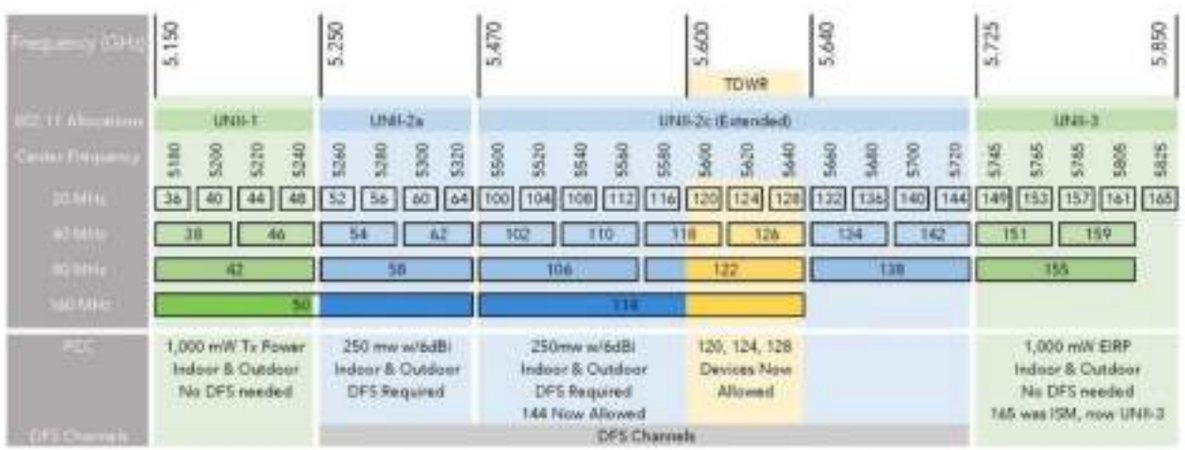
Rel. Year	1999	2007	2009	2013	2020	2023(?)
Freq. Band	2.4 GHz	2.4 GHz	2.4 + 5 GHz	5 GHz	2.4 + 5 + 6 GHz (6E)	2.4 + 5 + 6 GHz
Bandwidth	20 MHz	20 MHz	40 MHz	80 MHz, 160 MHz	80 MHz, 160 MHz	240 MHz, 320 MHz

2.4GHz



5GHz.

5 GHz Channel Allocations

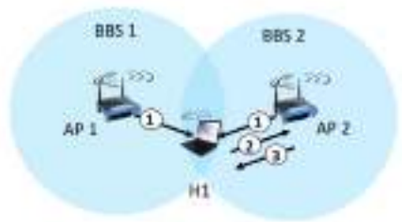


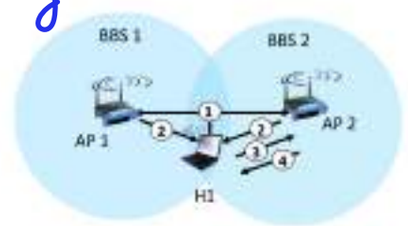
802.11

- LAN Architecture (access point)
  - → wireless host communicates w/ base station → (AP)

- Basic Service Set (BSS) (aka "cell") in infrastructure mode contains:
  - wireless hosts
  - access point (AP): base station
  - ad hoc mode: hosts only
- spectrum divided into channels at different frequencies
  - AP admin chooses frequency for AP
  - interference possible: channel can be same as that chosen by neighboring AP!
- arriving host: must associate with an AP
  - scans channels, listening for *beacon frames* containing AP's name (SSID) and MAC address
  - selects AP to associate with
  - then may perform authentication
  - then typically run DHCP to get IP address in AP's subnet

Passive / Active Scanning





passive scanning:
(1) beacon frames sent from APs
(2) association Request frame sent: H1 to selected AP
(3) association Response frame sent from selected AP to H1

active scanning: (hidden SSID)
(1) Probe Request frame broadcast from H1
(2) Probe Response frames sent from APs
(3) Association Request frame sent: H1 to selected AP
(4) Association Response frame sent from selected AP to H1

IEEE 802.11: multiple access

- avoid collisions: 2+ nodes transmitting at same time
- 802.11: CSMA - sense before transmitting
  - don't collide with detected ongoing transmission by another node
- 802.11: *no* collision detection!
  - difficult to sense collisions: high transmitting signal, weak received signal due to fading
  - can't sense all collisions in any case: hidden terminal, fading
  - goal: *avoid collisions:* CSMA/CollisionAvoidance → CSMA/CA

IEEE 802.11 MAC Protocol: CSMA/CA

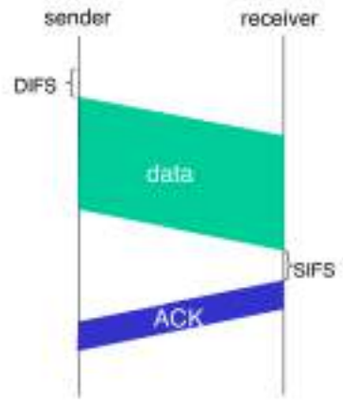
802.11 sender

1 if sense channel idle for DIFS then
transmit entire frame (no CD)

2 if sense channel busy then
start random backoff time
timer counts down while channel idle
transmit when timer expires
if no ACK, increase random backoff interval, repeat 2

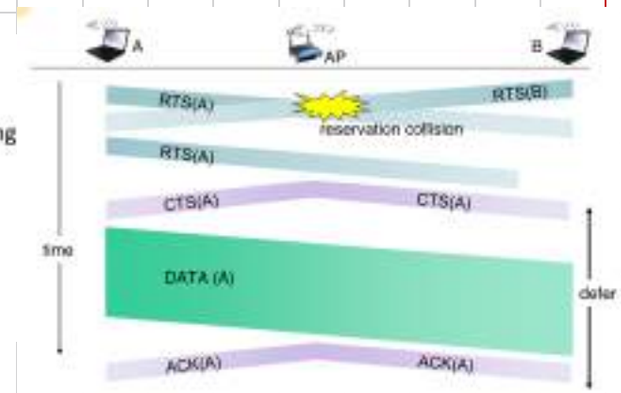
802.11 receiver

if frame received OK
return ACK after SIFS (ACK needed due to hidden terminal problem)

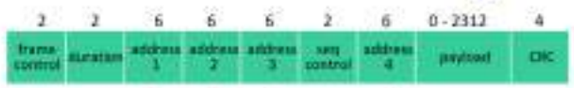


idea: sender "reserves" channel use for data frames using small reservation packets

- sender first transmits *small* request-to-send (RTS) packet to BS using CSMA
  - RTSs may still collide with each other (but they're short)
- BS broadcasts clear-to-send CTS in response to RTS
- CTS heard by all nodes
  - sender transmits data frame
  - other stations defer transmissions



802.11 frame: addressing

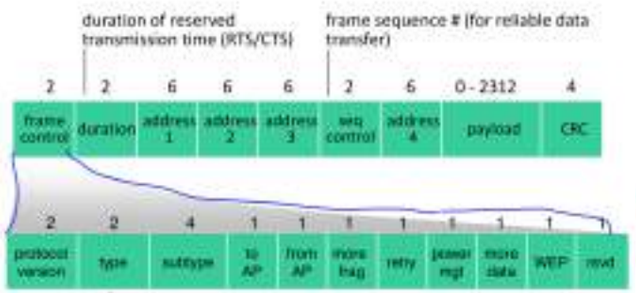


Address 1: MAC address of wireless host or AP to receive this frame

Address 2: MAC address of wireless host or AP transmitting this frame

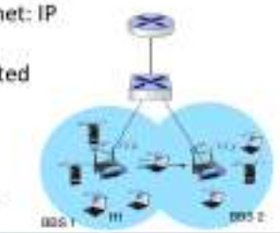
Address 3: MAC address of router interface to which AP is attached

Address 4: used only in ad hoc mode



802.11: mobility within same subnet

- H1 remains in same IP subnet: IP address can remain same
- switch: which AP is associated with H1?
  - self-learning : switch will see frame from H1 and "remember" which switch port can be used to reach H1



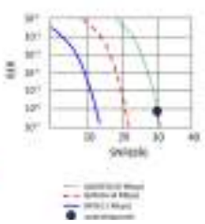
802.11: advanced capabilities

Rate adaptation

- base station, mobile dynamically change transmission rate (physical layer modulation technique) as mobile moves, SNR varies

1. SNR decreases, BER increase as node moves away from base station

2. When BER becomes too high, switch to lower transmission rate but with lower BER



power management

- node-to-AP: "I am going to sleep until next beacon frame"
  - AP knows not to transmit frames to this node
  - node wakes up before next beacon frame
- beacon frame: contains list of mobiles with AP-to-mobile frames waiting to be sent
  - node will stay awake if AP-to-mobile frames to be sent; otherwise sleep again until next beacon frame

• Cellular Network.

Radio Interfaces Frequency Bands

Item	3G (UMTS)					3G (CDMA)			
Band Class	B1	B2	B4	B5	B8	BC0		BC1	
Frequency (Mhz)	2100	1900 PCS	1700/2100 AWS 1	850	900 GSM	800		1900 PCS	
Item	4G (LTE)								
Band Class	B1	B2	B3	B4	B5	B7	B8	B12	B13
Frequency (Mhz)	2100	1900 PCS	1800+	1700/2100 AWS 1	850	2600	900	700 ac	700 c
Band Class	B17	B20	B25	B26	B28	B30	B38	B40	B41
Frequency (Mhz)	700 bc	800 DD	1900+	850+	700 APT	2300 WCS	TD 2600	TD 2300	TD 2500

4G/5G cellular networks

- *the* solution for wide-area mobile Internet
- widespread deployment/use:
  - more mobile-broadband-connected devices than fixed-broadband-connected devices devices (5-1 in 2019)!
  - 4G availability: 97% of time in Korea (90% in US)
- transmission rates up to 100's Mbps
- technical standards: 3rd Generation Partnership Project (3GPP)
  - wwww.3gpp.org
  - 4G: Long-Term Evolution (LTE) standard

4G/5G cellular networks

similarities to wired Internet

- edge/core distinction, but both below to same carrier
- global cellular network: a network of networks
- widespread use of protocols we've studied: HTTP, DNS, TCP, UDP, IP, NAT, separation of data/control planes, SDN, Ethernet, tunneling
- interconnected to wired Internet

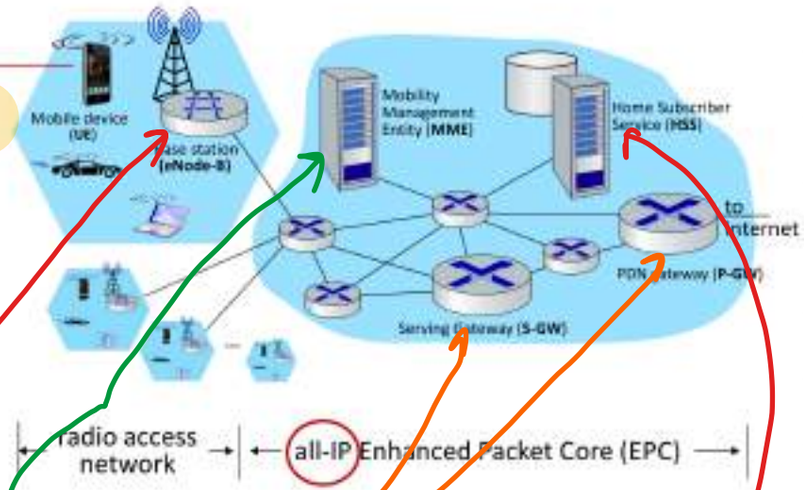
differences from wired Internet

- different wireless link layer
- user "identity" (via SIM card)
- business model: users subscribe to a cellular provider
  - strong notion of "home network" versus roaming on visited nets
  - global access, with authentication infrastructure, and inter-carrier settlements

Elements of 4G LTE architecture

Mobile device:

- smartphone, tablet, laptop, IoT, … with 4G LTE radio
- 64-bit International Mobile Subscriber Identity (IMSI), stored on SIM (Subscriber Identity Module) card
- User Equipment (UE)



Base station:

- at "edge" of carrier's network
- manages wireless radio resources, mobile devices in its coverage area ("cell")
- coordinates device authentication with other elements
- similar to WiFi AP but:
  - active role in user mobility
  - coordinates with nearly base stations to optimize radio use

Serving Gateway (S-GW), PDN Gateway (P-GW)

- lie on data path from mobile to/from Internet
- P-GW
  - gateway to mobile cellular network
  - Looks like other internet gateway router
  - provides NAT services

Home Subscriber Service

- stores info about mobile devices for which the HSS's network is their "home network"
- works with MME in device authentication

Mobility Management Entity

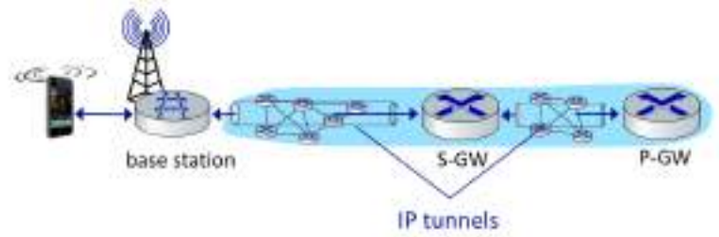
- device authentication (device-to-network, network-to-device) coordinated with mobile home network HSS
- mobile device management:
  - device handover between cells
  - tracking/paging device location
- path (tunneling) setup from mobile device to P-GW

Network Layer:

LTE: data plane control plane separation

control plane

- new protocols for mobility management , security, authentication (later)



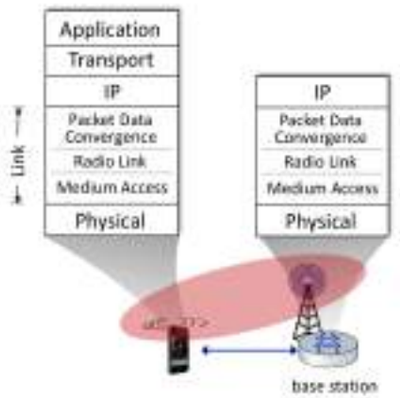
data plane

- new protocols at link, physical layers
- extensive use of tunneling to facilitate mobility

LTE data plane protocol stack: first hop

LTE link layer protocols:

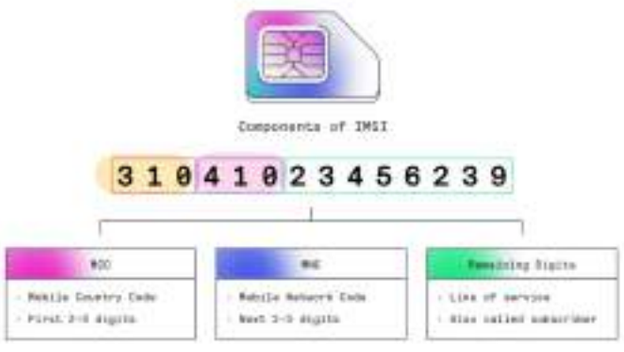
- Packet Data Convergence: header compression, encryption
- Radio Link Control (RLC) Protocol: fragmentation/reassembly, reliable data transfer
- Medium Access: requesting, use of radio transmission slots



LTE radio access network:

- Downstream channel: FDM, TDM within frequency channel (OFDM - orthogonal frequency division multiplexing)
  - "orthogonal": minimal interference between channels
  - upstream: FDM, TDM similar to OFDM
- each active mobile device allocated two or more 0.5 ms time slots over 12 frequencies
  - scheduling algorithm not standardized – up to operator

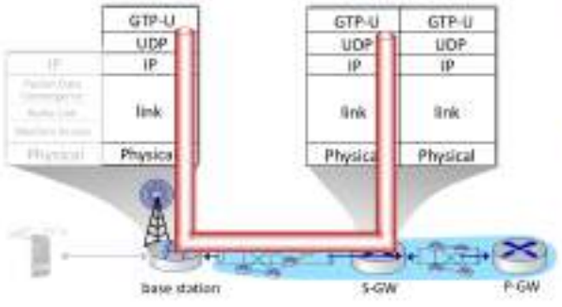
International Mobile Subscriber Identity (IMSI)



Thailand

MCC	MNC	Network	Operator or brand name	Status
520	0	CAT Telecom	My by CAT	Operational
520	1	Advanced Info Service	AIS	Operational
520	2	CAT Telecom	CAT	Operational
520	3	AWN	AIS 3G	Operational
520	4	Real Future	True Move	Operational
520	5	DTN	dTac	Operational
520	10		WCS	Inactive
520	15	Telephone Organization of Thailand	TOT 3G	Operational
520	18	Total Access Communications (DTAC)	dTac	Operational
520	23	Digital Phone (AIS)	AIS	Operational
520	25	True Corporation	WE-PCT	Operational
520	88	True Corporation	True Move H	Inactive
520	99	True Corporation	True Move	Operational

LTE data plane protocol stack: packet core



tunneling:

- mobile datagram encapsulated using GPRS Tunneling Protocol (GTP), sent inside UDP datagram to S-GW
- S-GW re-tunnels datagrams to P-GW
- supporting mobility: only tunneling endpoints change when mobile user moves

LTE data plane: associating with a BS

① BS broadcasts primary synch signal every 5 ms on all frequencies
- BSs from multiple carriers may be broadcasting synch signals

② mobile finds a primary synch signal, then locates 2nd synch signal on this freq.
- mobile then finds info broadcast by BS: channel bandwidth, configurations; BS's cellular carrier info
- mobile may get info from multiple base stations, multiple cellular networks

③ mobile selects which BS to associate with (*e.g.*, preference for home carrier)

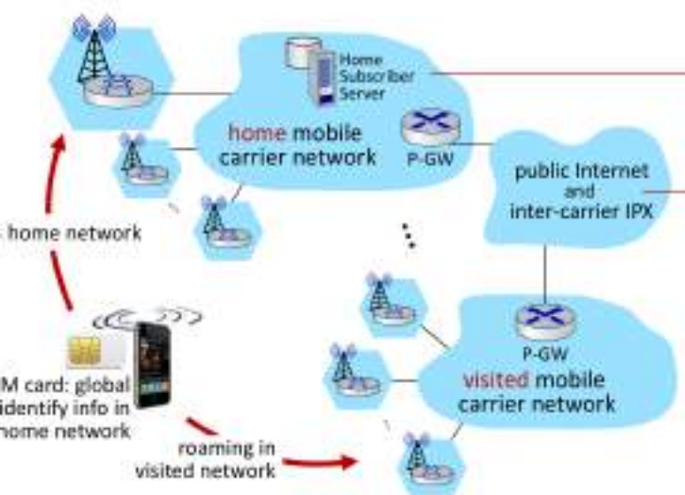
④ more steps still needed to authenticate, establish state, set up data plane

LTE mobiles: sleep modes

as in WiFi, Bluetooth: LTE mobile may put radio to "sleep" to conserve battery:

- light sleep: after 100's msec of inactivity
  - wake up periodically (100's msec) to check for downstream transmissions
- deep sleep: after 5-10 secs of inactivity
  - mobile may change cells while deep sleeping – need to re-establish association

Global cellular network: a network of IP networks



home network HSS:
- identify & services info, while in home network and roaming

all IP:
- carriers interconnect with each other, and public internet at exchange points
- legacy 2G, 3G: not all IP, handled otherwise

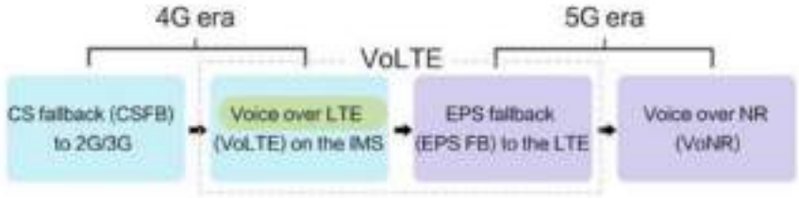
On to 5G!

- Goal: 10x increase in peak bitrate, 10x decrease in latency, 100x increase in traffic capacity over 4G
- 5G NR (new radio):
  - two frequency bands: FR1 (450 MHz–6 GHz) and FR2 (24 GHz–52 GHz): millimeter wave frequencies
  - not backwards-compatible with 4G
  - MIMO: multiple directional antenna
- Millimeter wave frequencies: much higher data rates (Gigabit level), but over shorter distances
  - pico-cells: cells diameters: 10-100 m
  - massive, dense deployment of new base stations required

Cellular networks 3G, 4G, VoLTE

Voice over LTE

4G (LTE) is only for Data Communication

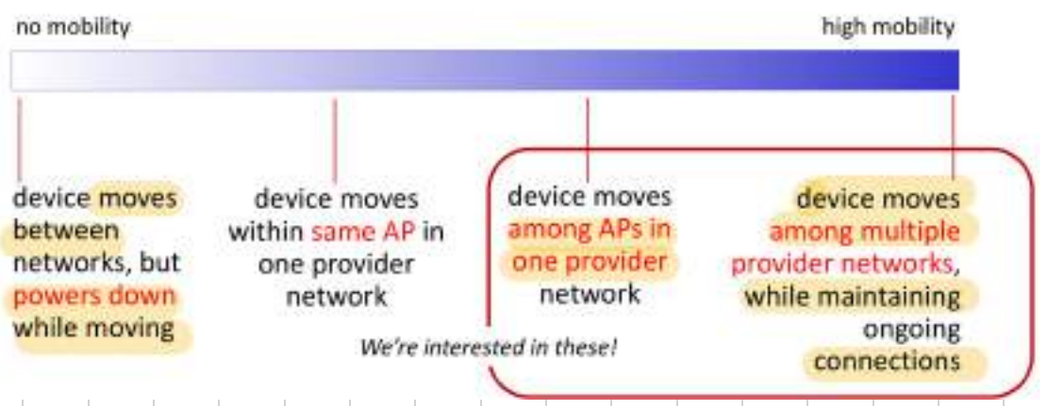


In the absence of the above CS mechanism for a voice call on a 4G network, LTE networks use a mechanism called Circuit switched Fallback (CSFB). This makes use of the existing 3G and 2G networks and tries to channelize voice communication for 4G

- Mobility

What is mobility?

- spectrum of mobility, from the network perspective:



Mobility approaches

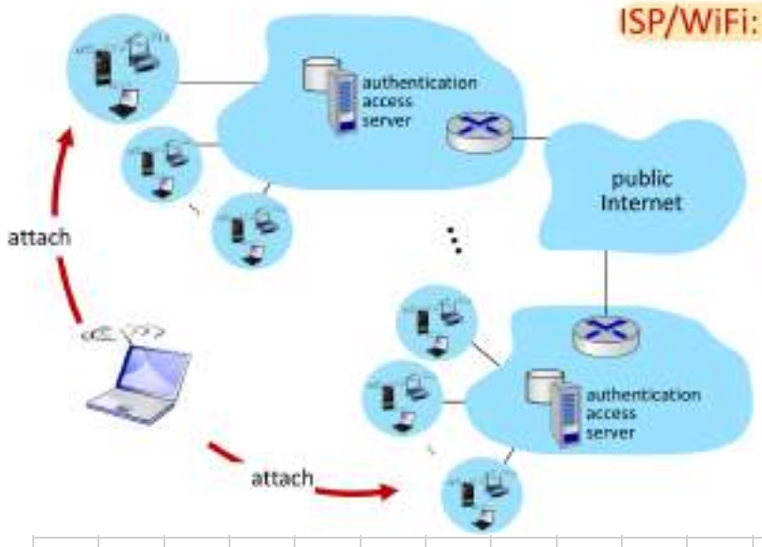
- let end-systems handle it: functionality at the "edge"
  - *indirect routing:* communication from correspondent to mobile goes through home network, then forwarded to remote mobile
  - *direct routing:* correspondent gets foreign address of mobile, send directly to mobile

Home network, visited network: 4G/5G

home network:
- (paid) service plan with cellular provider, e.g., Verizon, Orange
- home network HSS stores identify & services info

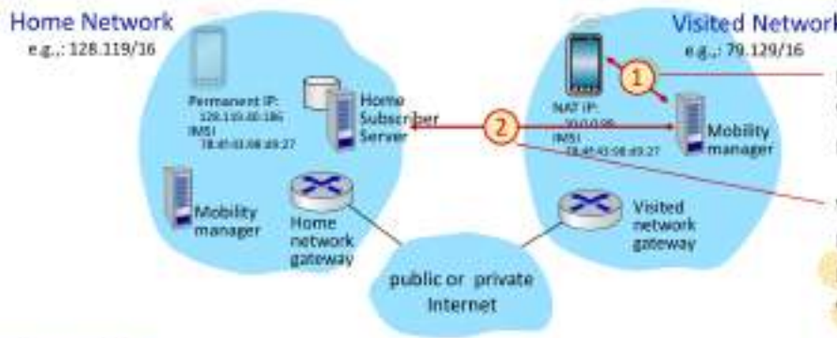
visited network:
- any network other than your home network
- service agreement with other networks: to provide access to visiting mobile

Home network, visited network: ISP/WiFi



ISP/WiFi: no notion of global "home"
- credentials from ISP (e.g., username, password) stored on device or with user
- ISPs may have national, international presence
- different networks: different credentials
  - some exceptions (e.g., eduroam)
  - bc.xx@tu.ac.th

Registration: home needs to know where you are!

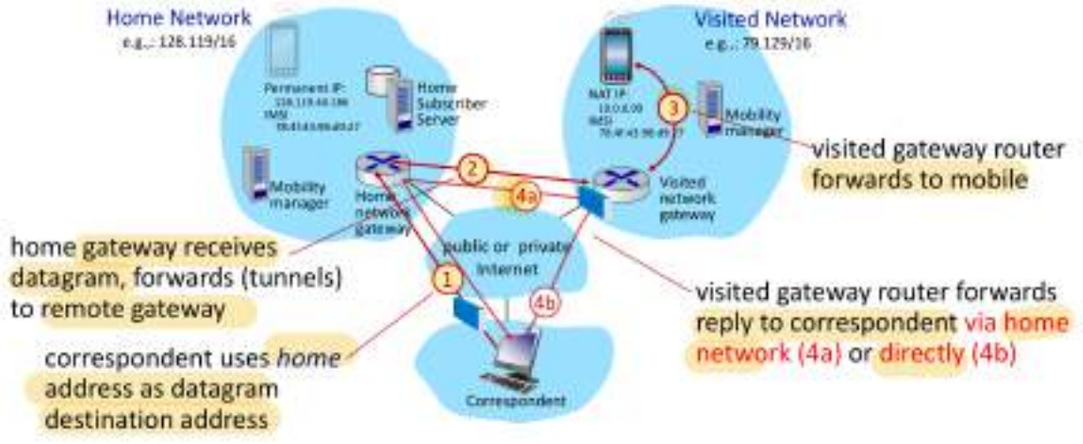


mobile *associates* with visited mobility manager

visited mobility manager *registers* mobile's location with home HSS

end result:
- visited mobility manager knows about mobile
- home HSS knows location of mobile

Mobility with indirect routing



visited gateway router forwards to mobile

home gateway receives datagram, forwards (tunnels) to remote gateway

visited gateway router forwards reply to correspondent via home network (4a) or directly (4b)

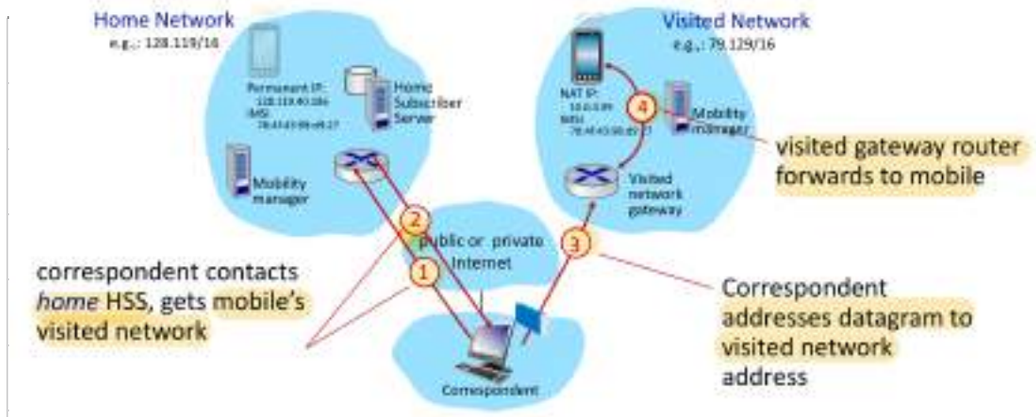
correspondent uses *home* address as datagram destination address

Mobility with indirect routing: comments

- triangle routing:
  - inefficient when correspondent and mobile are in same network

- mobile moves among visited networks: transparent to correspondent!
  - registers in new visited network
  - new visited network registers with home HSS
  - datagrams continue to be forwarded from home network to mobile in new network
  - *on-going (e.g., TCP) connections between correspondent and mobile can be maintained!*

Mobility with direct routing



Wireless, mobility: impact on higher layer protocols

- logically, impact *should* be minimal …
  - best effort service model remains unchanged
  - TCP and UDP can (and do) run over wireless, mobile
- … but performance-wise:
  - packet loss/delay due to bit-errors (discarded packets, delays for link-layer retransmissions), and handover loss
  - TCP interprets loss as congestion, will decrease congestion window un-necessarily
  - delay impairments for real-time traffic
  - bandwidth a scare resource for wireless links

• Network Security

What is network security?

Confidentiality: only sender, intended receiver should "understand" message contents
- sender encrypts message
- receiver decrypts message

Authentication: sender, receiver want to confirm identity of each other

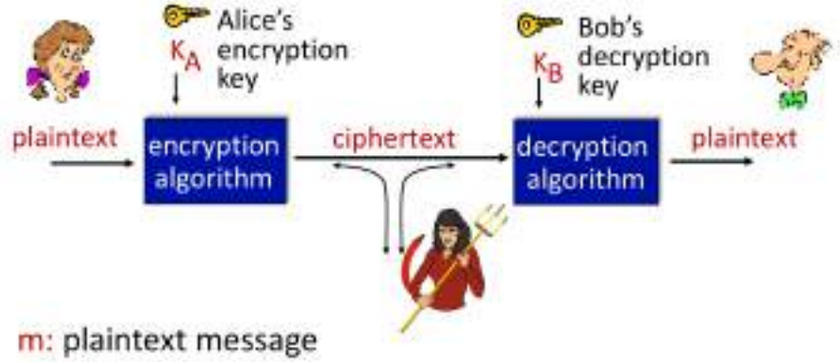
Message integrity: sender, receiver want to ensure message not altered (in transit, or afterwards) without detection

Access and availability: services must be accessible and available to users

Q: What can a "bad guy" do?

A: A lot! (recall section 1.6)
- eavesdrop: intercept messages
- actively insert messages into connection
- impersonation: can fake (spoof) source address in packet (or any field in packet)
- hijacking: "take over" ongoing connection by removing sender or receiver, inserting himself in place
- denial of service: prevent service from being used by others (e.g., by overloading resources)

• Cryptography



m: plaintext message

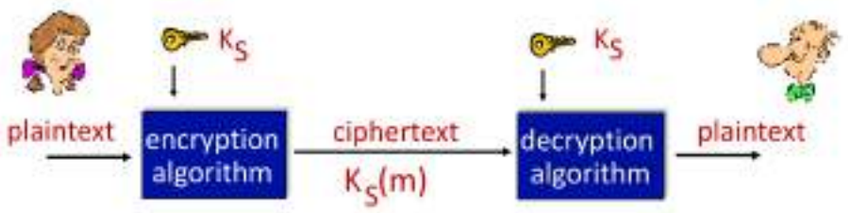
$K_A(m)$: ciphertext, encrypted with key K_A

$m = K_B(K_A(m))$

Breaking an encryption scheme

- cipher-text only attack: Trudy has ciphertext she can analyze
- two approaches:
  - brute force: search through all keys
  - statistical analysis
- known-plaintext attack: Trudy has plaintext corresponding to ciphertext
  - *e.g.,* in monoalphabetic cipher, Trudy determines pairings for a,l,i,c,e,b,o,
- chosen-plaintext attack: Trudy can get ciphertext for chosen plaintext

Symmetric key cryptography



symmetric key crypto: Bob and Alice share same (symmetric) key: K

- *e.g.,* key is knowing substitution pattern in mono alphabetic substitution cipher

Q: how do Bob and Alice agree on key value?

Simple encryption scheme

substitution cipher: substituting one thing for another

- monoalphabetic cipher: substitute one letter for another

```
plaintext:  abcdefghijklmnopqrstuvwxyz
ciphertext: mnbvcxzasdfghjklpoiuytrewq
```

e.g.:
```
Plaintext: bob. i love you. alice
ciphertext: nkn. s gktc wky. mgsbc
```

Encryption key: mapping from set of 26 letters to set of 26 letters

A more sophisticated encryption approach

- n substitution ciphers, $M_1,M_2,\ldots,M_n$
- cycling pattern:
  - e.g., n=4: M_1,M_3,M_4,M_3,M_2; M_1,M_3,M_4,M_3,M_2; ..
- for each new plaintext symbol, use subsequent substitution pattern in cyclic pattern
  - dog: d from M_1, o from M_3, g from M_4

Encryption key: n substitution ciphers, and cyclic pattern

- key need not be just n-bit pattern

Symmetric key crypto: DES

DES: Data Encryption Standard

- US encryption standard [NIST 1993]
- 56-bit symmetric key, 64-bit plaintext input
- block cipher with cipher block chaining
- how secure is DES?
  - DES Challenge: 56-bit-key-encrypted phrase decrypted (brute force) in less than a day
  - no known good analytic attack
- making DES more secure:
  - 3DES (triple-DES): encrypt 3 times with 3 different keys

AES: Advanced Encryption Standard

- Symmetric-key NIST standard, replaced DES (Nov 2001)
- processes data in 128 bit blocks
- 128, 192, or 256 bit keys
- brute force decryption (try each key) taking 1 sec on DES, takes 149 trillion years for AES

Public Key Cryptography

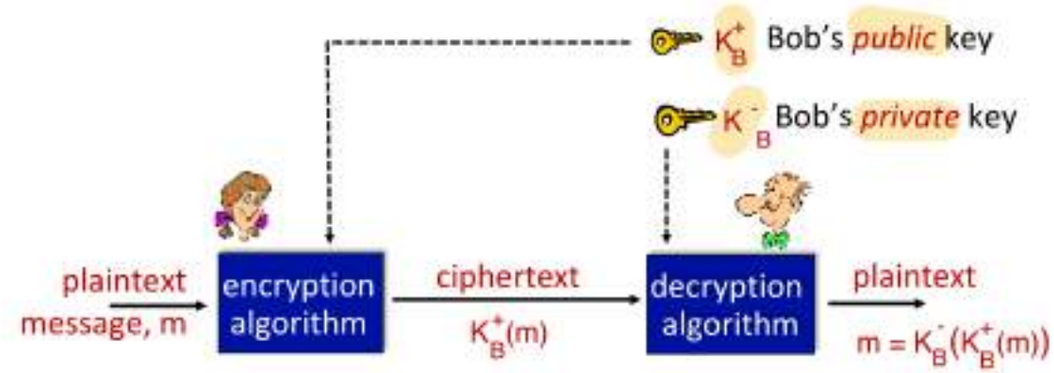
symmetric key crypto:

- requires sender, receiver know shared secret key
- Q: how to agree on key in first place (particularly if never "met")?

public key crypto

- *radically* different approach [Diffie-Hellman76, RSA78]
- sender, receiver do *not* share secret key
- *public* encryption key known to *all*
- *private* decryption key known only to receiver

• secret key
• public key



encrypt by public key, decrypt by private key

RSA (Rivest–Shamir–Adleman)

- message: just a bit pattern
- bit pattern can be uniquely represented by an integer number
- thus, encrypting a message is equivalent to encrypting a number

example:

- m= 10010001. This message is uniquely represented by the decimal number 145.
- to encrypt m, we encrypt the corresponding number, which gives a new number (the ciphertext).

RSA in practice: session keys

- Exponentiation in RSA is computationally intensive
- DES is at least 100 times faster than RSA
- use public key crypto to establish secure connection, then establish second key – symmetric session key – for encrypting data

session key, K_S

- Bob and Alice use RSA to exchange a symmetric session key K_S
- once both have K_S, they use symmetric key cryptography
- EX: SSH (Secure-Shell)

• Authentication
prove others identity of yourself.

Protocol ap1.0: Alice says "I am Alice"

fail

Protocol ap1.0: Alice says "I am Alice"

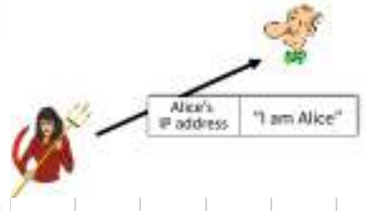
in a network, Bob can not "see" Alice, so Trudy simply declares herself to be Alice

Protocol ap2.0: Alice says "I am Alice" in an IP packet containing her source IP address

fail

Protocol ap2.0: Alice says "I am Alice" in an IP packet containing her source IP address

Protocol ap3.0: Alice says "I am Alice" and sends her secret password to "prove" it.



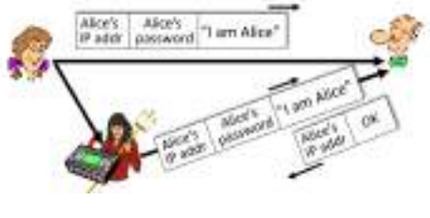
Trudy can create a packet "spoofing" Alice's address

failure scenario??

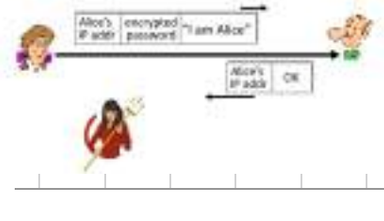
fail

modified 3.0.

Protocol ap3.0: Alice says "I am Alice" and sends her encrypted secret password to "prove" it.



playback attack: Trudy records Alice's packet and later plays it back to Bo



failure scenario??

Protocol ap3.0: Alice says "I am Alice" and sends her encrypted secret password to "prove" it.

playback attack still works: Trudy records Alice's packet and later plays it back to Bob

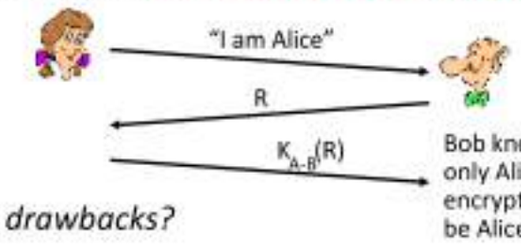
Authentication: a fourth try

Goal: avoid playback attack

nonce: number (R) used only once-in-a-lifetime

protocol ap4.0: to prove Alice "live", Bob sends Alice nonce, R

- Alice must return R, encrypted with shared secret key



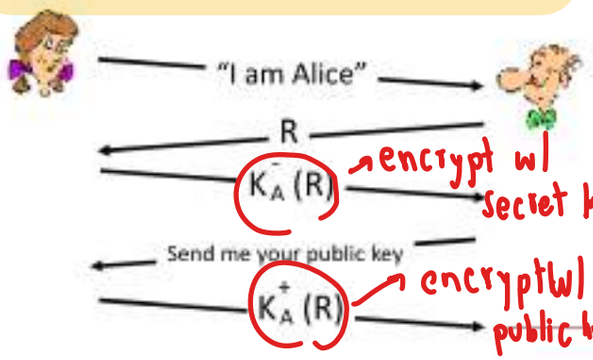
Bob know Alice is live, and only Alice knows key to encrypt nonce, so it must be Alice!

Failures, drawbacks?

Authentication: ap5.0

ap4.0 requires shared symmetric key - can we authenticate using public key techniques?

ap5.0: use nonce, public key cryptography



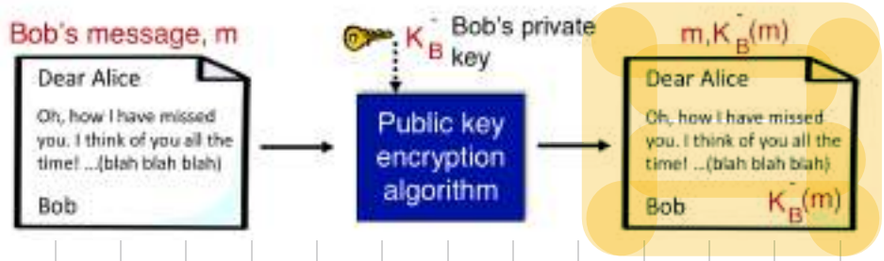
Bob computes $K_A^+(K_A^-(R)) = R$

and knows only Alice could have the private key, that encrypted R such that $K_A^+(K_A^-(R)) = R$

Digital signatures

cryptographic technique analogous to hand-written signatures:

- sender (Bob) digitally signs document: he is document owner/creator.
- *verifiable:* recipient (Alice) can prove to someone that Bob, and no one else (including Alice), must have signed document
- simple digital signature for message m:
  - Bob signs m by encrypting with his private key K_B^-, creating "signed" message, $K_B^-(m)$



- suppose Alice receives msg m, with signature: m, $K_B^-(m)$
- Alice verifies m signed by Bob by applying Bob's public key K_B^+ to $K_B^-(m)$ then checks $K_B^+(K_B^-(m)) = m$.
- If $K_B^+(K_B^-(m)) = m$, whoever signed m must have used Bob's private key

Alice thus verifies that:

- Bob signed m
- no one else signed m
- Bob signed m and not m'

Message digests

Computationally expensive to public-key-encrypt long messages

goal: fixed-length, easy- to-compute digital "fingerprint"

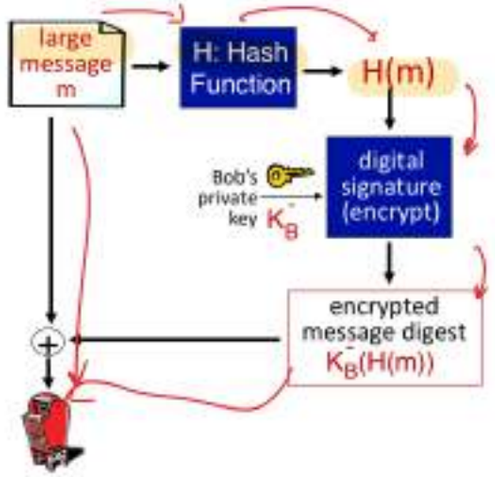
- apply hash function H to *m*, get fixed size message digest, *H(m)*

Hash function properties:

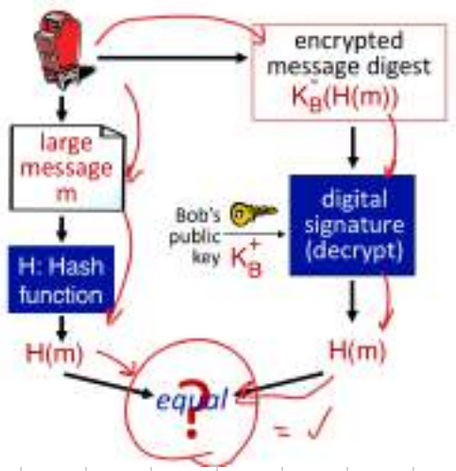
- many-to-1
- produces fixed-size msg digest (fingerprint)
- given message digest *x*, computationally infeasible to find *m* such that *x = H(m)*

Digital signature = signed message digest

Bob sends digitally signed message:



Alice verifies signature, integrity of digitally signed message:

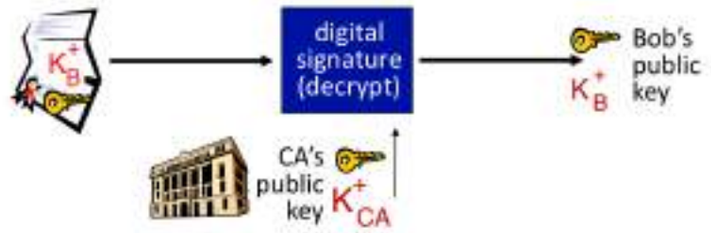


- MD5 hash function widely used (RFC 1321) → obsolete
  - computes 128-bit message digest in 4-step process.
  - arbitrary 128-bit string x, appears difficult to construct msg m whose MD5 hash is equal to x
- SHA-1 is also used
  - US standard [NIST, FIPS PUB 180-1]
  - 160-bit message digest

Public key Certification Authorities (CA)

- Certification Authority (CA): binds public key to particular entity, E
- entity (person, website, router) registers its public key with CE provides "proof of identity" to CA
  - CA creates certificate binding identity E to E's public key
  - certificate containing E's public key digitally signed by CA: CA says "this is E's public key"

- when Alice wants Bob's public key:
  - gets Bob's certificate (Bob or elsewhere)
  - apply CA's public key to Bob's certificate, get Bob's public key



- Operational Security

Firewalls

firewall

isolates organization's internal network from larger Internet, allowing some packets to pass, blocking others

Firewalls: why

prevent denial of service attacks:

- SYN flooding: attacker establishes many bogus TCP connections, no resources left for "real" connections

prevent illegal modification/access of internal data

- e.g., attacker replaces CIA's homepage with something else

allow only authorized access to inside network

- set of authenticated users/hosts

three types of firewalls:

- stateless packet filters
- stateful packet filters
- application gateways

Stateless packet filtering

Policy	Firewall Setting
no outside Web access	drop all outgoing packets to any IP address, port 80
no incoming TCP connections, except those for institution's public Web server only.	drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80
prevent Web-radios from eating up the available bandwidth.	drop all incoming UDP packets - except DNS and router broadcasts.
prevent your network from being used for a smurf DoS attack.	drop all ICMP packets going to a "broadcast" address (e.g. 130.207.255.255)
prevent your network from being tracerouted	drop all outgoing ICMP TTL expired traffic

- internal network connected to Internet via router firewall
- filters packet-by-packet, decision to forward/drop packet based on:
  - source IP address, destination IP address
  - TCP/UDP source, destination port numbers
  - ICMP message type
  - TCP SYN, ACK bits
- example 1: block incoming and outgoing datagrams with IP protocol field = 17 (UDP) and with either source or dest port = 23
  - result: all incoming, outgoing UDP flows and telnet connections are blocked
- example 2: block inbound TCP segments with ACK=0
  - result: prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside

Access Control Lists

ACL: table of rules, applied top to bottom to incoming packets: (action, condition) pairs: looks like OpenFlow forwarding (Ch. 4)!

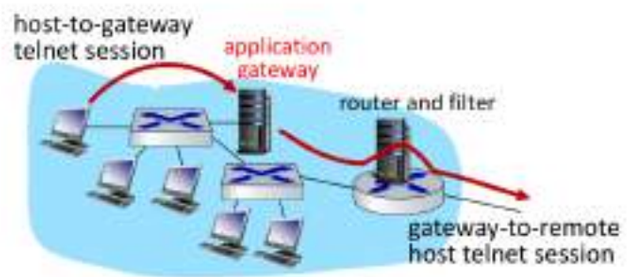
action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	---
deny	all	all	all	all	all	all

action	source address	dest address	protocol	source port	dest port	flag bit
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK

- *stateful packet filter:* track status of every TCP connection
  - track connection setup (SYN), teardown (FIN): determine whether incoming, outgoing packets "makes sense"
  - timeout inactive connections at firewall: no longer admit packets

Application gateways

- filter packets on application data as well as on IP/TCP/UDP fields.
- *example:* allow select internal users to telnet outside



1. require all telnet users to telnet through gateway.
2. for authorized users, gateway sets up telnet connection to dest host
   - gateway relays data between 2 connections
3. router filter blocks all telnet connections not originating from gateway

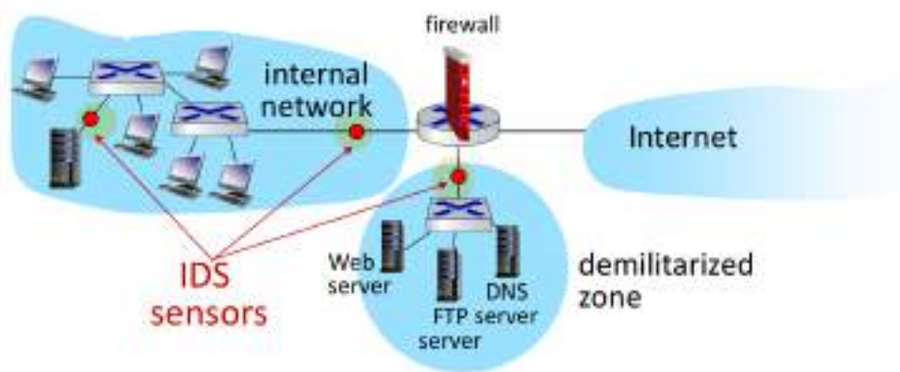
Limitations of firewalls, gateways

- IP spoofing: router can't know if data "really" comes from claimed source
- if multiple apps need special treatment, each has own app. gateway
- client software must know how to contact gateway
  - e.g., must set IP address of proxy in Web browser
- filters often use all or nothing policy for UDP
- *tradeoff:* degree of communication with outside world, level of security
- many highly protected sites still suffer from attacks

Intrusion detection systems

- packet filtering:
  - operates on TCP/IP headers only
  - no correlation check among sessions
- IDS: intrusion detection system
  - deep packet inspection: look at packet contents (e.g., check character strings in packet against database of known virus, attack strings)
  - examine correlation among multiple packets
    - port scanning
    - network mapping
    - DoS attack

multiple IDSs: different types of checking at different locations



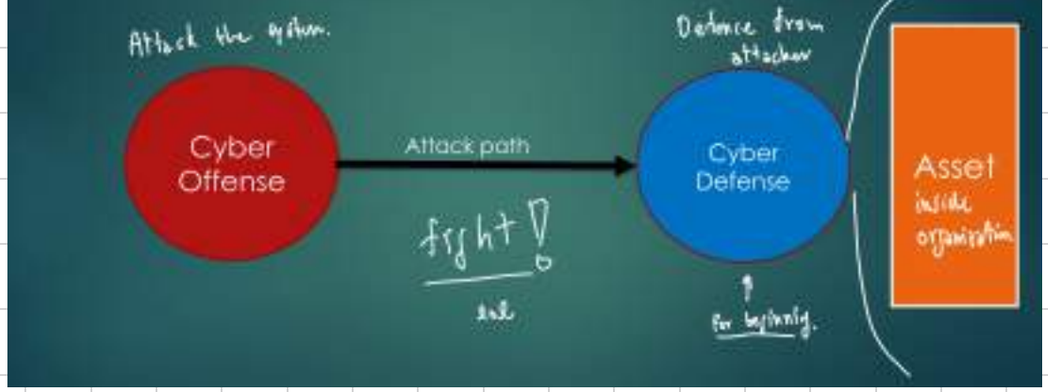
• Cyber Security

What is Cyber Security?

Cyber security is all about reducing the risk of attacks to computers, networks, or software.

From CIA to APT: An Introduction to Cyber Security, Amoroso

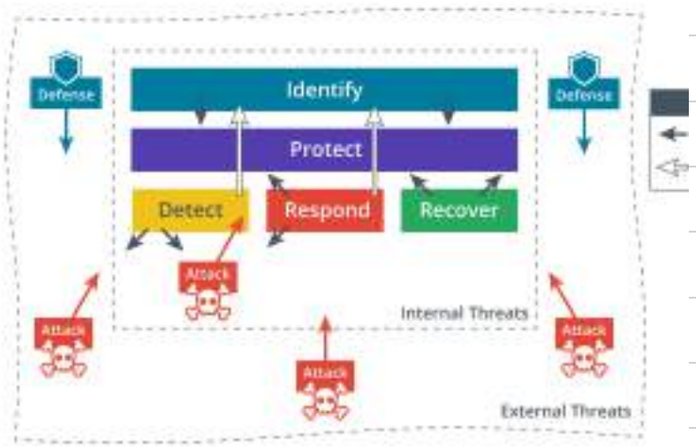
Cyber Offense vs. Cyber Defense



NIST Cybersecurity Framework

(National Institute of Standards and Technology (NIST))

- Identify
- Protect
- Detect
- Respond
- Recover



Cyber Security Competencies

- Risk assessments and testing
- Specifying, sourcing, installing, and configuring secure devices and software
- Access control and user privileges
- Auditing logs and events
- Incident reporting and response
- Business continuity and disaster recovery
- Security training and education programs

Roles and Responsibilities

- Overall responsibility
  - Chief Security Officer (CSO)
  - Chief Information Security Officer (CISO)
- Managerial
- Technical
  - Information Systems Security Officer (ISSO)
- Non-technical
- Due care/liability

Information Security Business Units

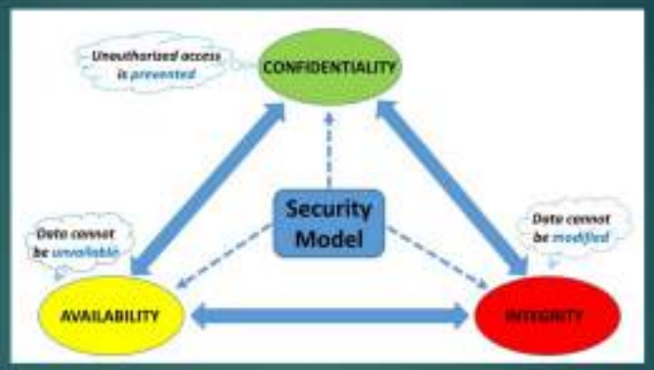
required by that law, 2018

- Security Operations Center (SOC)
- Incident response
  - Cyber incident response team (CIRT)
  - Computer security incident response team (CSIRT)
  - Computer emergency response team (CERT)

• Concept

CIA Model - Core Security Goals

- Confidentiality
- Integrity
- Availability



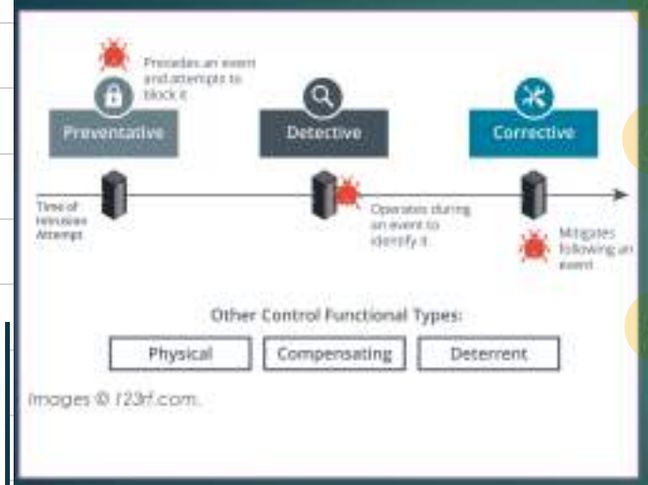
AAA (authentication, authorization, and accounting)

Security Control Categories

- Technical
  - Controls implemented in operating systems, software, and security appliances
- Operational
  - Controls that depend on a person for implementation
- Managerial
  - Controls that give oversight of the system

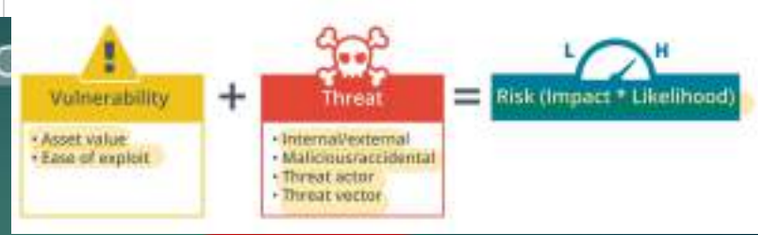
- Confidentiality
  - Encryption
  - Access controls
  - Steganography
- Integrity
  - Hashing
  - Digital signatures
  - Certificates
  - Non-repudiation
- Availability
  - Redundancy
  - Fault tolerance
  - Patching

Security Control Functional Types



- Preventive
  - Physically or logically restricts unauthorized access
  - Operates before an attack
- Detective
  - May not prevent or deter access, but it will identify and record any attempted or successful intrusion
  - Operates during an attack
- Corrective
  - Responds to and fixes an incident and may also prevent its reoccurrence
  - Operates after an attack

Basic Risk Concepts



- Threats
- Vulnerabilities
  - Any weakness
- Risk is
  - The likelihood that a threat will exploit a vulnerability
- Risk mitigation
  - Reduces the chances that a threat will exploit a vulnerability by implementing controls

• Threat Actors

Attributes of Threat Actors

- Known threats versus adversary behaviors
- Internal/external
- Intent/motivation
  - Maliciously targeted versus opportunistic
  - Accidental/unintentional
- Level of sophistication
  - Resources/funding
  - Adversary capability levels

Threat Actors (Black hat)

- Script kiddie
  - Little expertise
- Hacktivist
  - Part of an activist movement
- Insider
  - Employee (can become a malicious insider)
- Organized crime
  - Typically motivated by money
- Competitor
- Nation state
  - advanced persistent threat (APT)

- Criminal syndicates
  - Operate across legal jurisdictions
  - Motivated by criminal profit
  - Can be very well resourced and funded
- Competitors
  - Cyber espionage
  - Combine with insider threat

Impacts from Vulnerabilities

- Data breaches and data exfiltration impacts
  - Data breach is where confidential data is read or transferred without authorization
  - Data exfiltration is the methods and tools by which an attacker transfers data without authorization
- Identity theft
  - Abuse of data from privacy breaches
- Data loss and availability loss impacts
  - Availability is also a critical security property
- Financial and reputation impacts

Insider Threat Actors

- Malicious insider threat
  - Has or has had authorized access
  - Employees, contractors, partners
  - Sabotage, financial gain, business advantage
- Unintentional insider threat
  - Weak policies and procedures
  - Weak adherence to policies and procedures
  - Lack of training/security awareness
  - Shadow IT

State Actors and Advanced Persistent Threats

- State-backed groups
  - Attached to military/secret services
  - Highly sophisticated
- Advanced Persistent Threat (APT)
- Espionage and strategic advantage
- Deniability

Software Vulnerabilities and Patch Management

- Exploits for faults in software code
- Applications
  - Different impacts and exploit scenarios
  - Client versus server apps
- Operating system (OS)
  - Obtain high level privileges
- Firmware
  - PC firmware
  - Network appliances and Internet of Things devices
- Improper or weak patch management
  - Undocumented assets
  - Failed updates and removed patches

Attack Surface and Vectors

- Attack surface
  - Points where an attacker can discover/exploit vulnerabilities in a network or application
- Vectors
  - Direct access
  - Removable media
  - Email
  - Remote and wireless
  - Supply chain
  - Web and social media
  - Cloud

Zero-day and Legacy Platform Vulnerabilities

- Zero-day
  - Vulnerability is unknown to the vendor
  - Threat actor develops an exploit for which there is no patch
  - Likely to be used against high value targets
- Legacy platform
  - Vendor no longer releases security patches

Weak Host Configurations

- Default settings
  - Vendor may not release product in a default-secure configuration
- Unsecured root accounts
  - Threat actor will gain complete control
  - Limit ability to login as superuser
- Open permissions
  - Configuration errors allowing unauthenticated access
  - Allowing write access when only read access is appropriate
- Open ports and services
  - Restrict using an access control list
  - Disable unnecessary services or block ports
  - Block at network perimeter
- Unsecure protocols
  - Cleartext data transmissions are vulnerable to snooping and eavesdropping
- Weak encryption
  - Storage and transport encryption
  - Key is generated from a weak password
  - Cipher has weaknesses
  - Key distribution is not secure
- Errors
  - Error messages that reveal too much information

Third-Party Risks

- Supply chains
- Vendor management
  - Process for selecting suppliers and evaluating risks
  - System integration
  - Lack of vendor support
- Outsourced code development
- Data storage
- Cloud-based versus on-premises risks

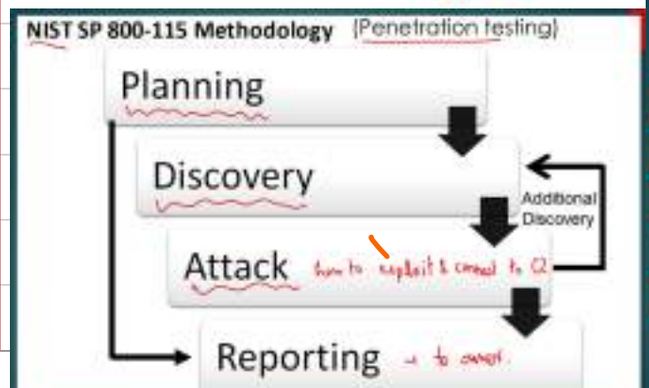
• Security Assessment

Penetration Testing

- **Vulnerability assessment**: The practice of evaluating a computer, network, or application to identify potential weaknesses.
- **Penetration testing (pen testing)**: The practice of evaluating a computer, network, or application to identify potential vulnerabilities, and then exploiting them to gain unauthorized access to key systems and data, and culminating in the production of evidence and a report.

simulate a hacker, hack to the system

Cyber Kill Chain



Penetration Testing

- Pen test or ethical hacking
- Verify threat
  - Identify vulnerability and the vector by which it could be exploited
- Bypass security controls
  - Identify lack of controls or ways to circumvent existing controls
- Actively test security controls
  - Examine weaknesses that render controls ineffective
- Active and highly intrusive techniques, compared to vulnerability assessment

Rules of Engagement

- Agreement for objectives and scope
- Authorization to proceed from system owner and affected third-parties
- Attack profile
  - Black box (unknown environment)
  - White box (known environment)
  - Gray box (partially known environment—to model insider threat agents, for instance)

Exercise Types

- Red team
  - Performs the offensive role
- Blue team
  - Performs the defensive role
- White team
  - Sets the rules of engagement and monitors the exercise
- Purple team
  - Exercise set up to encourage collaboration
  - Red and blue teams share information and debrief regularly
  - Might be assisted by a facilitator

Passive and Active Reconnaissance

- Pen testing and kill chain attack life cycle
- Reconnaissance phase
  - Passive techniques unlikely to alert target
  - Active techniques are detectable
- Open Source Intelligence (OSINT)
- Social engineering
- Footprinting

- Info gathering
  - Passive
    - Open Source Intelligence (OSINT)
    - Google hacking
    - Info gather on job posting/resume
    - Email harvesting
    - Whois enumeration
  - Active
    - port scanning check for open port
    - Vulnerbility scanner

-THE END-