==========

==========

CHAPTER 5 – NETWORK LAYER

==========

==========

GOALS:

1) Principles behind network plane (Traditional Routing Algorithms ++ SDN Controllers ++ Network Management configuration)

-SDN - (Load-sharing, firewalling, simple forwarding, NAT) ++ Match & Action (Forwarding ++ Dropping if no action ++ Modify-Field) ++ Flow Tables

2) Instantiation and Implemention in the Internet (OSPF, BGP ++ OpenFlow, ODL, and ONOS controllers ++ ICMP ++ SNMP, YANG/NETCONF)

5.1 INTRO:

1) Network-layer functions (forwarding = data plane; routing = network layer.)

2) Two Approaches to Network-layer (per router control (traditional) or software control (SDN)

--Per router control (OSPF and BGP based on this)

--SDN

5.2 Routing Algorithms:

1) Routing Protocols

-Find good 'paths' where paths = sequence of routers a packet traverses from source to destination. Good routes = fast, least congested, 'cost.' ++ If path (x,y) doesn't exist, set c(x, y)=∞.

BOOK:

-Dijkstra (See Formulas)

----Characteristics:

-Bellman-Ford - $dx(y) = \min v5c(x, v) + dv(y)$ (SEE FORMULAS)

--Characteristics:
----Iteractive, asynchronous, and distributed.
----Self-terminating / Self-stopping (Notifies neighbors only when its DV changes)
----Receives information from direct neighbors, processes, sends info out to all neighbors, stops once no more info to share
----min(v) taken over all of x's neighbors.
----From time to time, each node sends its distance vector algorithm to neighbors.
----When x receives DV estimate, it updates its own DV by BF equation
----"Good news travels fast" = routers quickly learn of good new routes.
----"Bad news travels slow" = Routers slowly update when a previously good route becomes slow - "Count to infinity problem"

-Comparison LS vs DV Algorithms:

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

2) Graph Abstraction: Link Costs

-Cost defined by network operator. Could be set to a flat value, or inversely related to bandwidth or congestion

3) Link Classifications

-Global = All routers have complete topology. (Includes "Link State" Algorithms)

vs

-Decentralized = Iterative process of computation, exchange info with neighbors. (Includes "Distance vector" Algorithms) (AKA nodes only know cost to their neighbors. Can learn least-cost to extended network via neighbors.)

-Static = Route changes slowly over time.

vs

-Dynamic = Route changes quickly - periodic updates or in response to link cost changes. (More susceptible to routing loops and oscillation.)

Bonus - Load Sensitive vs. Load Insensitive - Does route cost change based on congestion? (Modern internet routing algorithms are load-insensitive due to too many problems) (That refers to RIP, OSPF, and BGP)

LINK STATE: Dijkstra's Link State Routing Algorithm:

-Centralized

-Computes least cost path (forwarding table) from one node to all nodes

-Iterative - after k iterations, know least cost path to k destinations.

notation

- $c_{x,y}$: direct link cost from node x to y; $= \infty$ if not direct neighbors
- $D(v)$: *current* estimate of cost of least-cost-path from source to destination v
- $p(v)$: predecessor node along path from source to v
- N': set of nodes whose least-cost-path *definitively* known

-Works via link-state broadcast (All nodes have identical, complete network overview)

DECENTRALIZED - Bellman-Ford

-"Good news travels fast" =

-"Bad news travels slow" =

5.3 Intra-AS: OSPF:

KEY SLIDE POINTS:

-Scalable & realistic routing (routers not all identical ++ networks not 'flat') ++ Scale (Can't store all destinations.)

-Network of networks (Network admin controls routing on sub-network)

-AS = "Autonomous Systems" = domains.

-Intra-AS = Routing within AS (Routers in AS must run same intra-domain protocols ++ routers in different AS can run different intra-domain protocols ++ gateway router at 'edge' has links to other routers AS'es)

-Inter-AS = Routing with other AS'es (Gateways perform both inter-AS and intra-AS routing)

---Forwarding table includes intra-AS routing and inter-AS routing

---AS1 inter-domain routing needs to know which destinations are reachable from any AS2, AS3, ..., ASn that are neighbors.

-OSPF = Classic LS - Each router floods AS advertisements to all other routers in AS ++ Multiple cost type options to use (Bandwidth, delay) ++ Each router has full topology & uses Dijkstra Algorithm) ++ Security (all messages authenticated)

-Hierarchical OSPF - 2 areas: Local area & backbone ++ (Link state advertisments either flood area OR backbone - intra/inter separation) ++ area border routers(advertise distance to local area in backbone - inter) ++ local area routers (distance within area - intra) ++ Backbone router(OSPF in backbone) ++ Boundary router(connects to different AS)

TEXT EXPANDED POINTS:

-AS separates internet into multiple 'mini-networks' or subnets to avoid overwhelming memory & processing costs with 'naive' LS and DV routing. It also meets organization/ISP needs.

-OSPF (& related IS-IS) are commonly used in AS - Uses flooding advertisement and Dijkstra's Algorithm to find shortest path to all *subnets.* Link weights are set according to network administrator decision. ++ Broadcasts link states every 30 mins even if no change ++ default = no authentication (Can implement simple and MD5) ++ Allows multiple same-cost-paths (can split traffic on equal cost paths) ++ unicast/multicast support ++ Hierarchy support within AS (configure AS into subnet-like 'areas')

5.4 Routing Among the ISPs: BGP:

KEY SLIDE POINTS:

BGP = Border Gateway Patrol = THE default inter-domain routing protocol - holds the internet together ++ (Allows subnets to advertise existence and reachable destinations to internet) ++ **DECENTRALIZED** + Asynchronous.

--eBGP - obtain subnet reachability information from neighboring AS'es. (Inter) (External)

--iBGP - Propagate reachability information to all AS-internal routers. (Intra) (Internal)

--Determines best routes based on reachability/policy.

--AS advertising path means AS will forward datagrams to destination.

--BGP path advertisement = prefix (destination being advertised aka subnet/collections of subnets) & attributes(AS-PATH = list of ASes that advertisment has passed AND NEXT-HOP = specific internal router to next-hop AS)

----Inner area routers learn about advertisements from connected **edge**/'gateway' **routers**. (Obtain prefix reachability info from neighboring AS'es & determine best route)

---Policy-based routing - can accept/reject path & AS can determine whether to advertise path to neighboring AS'es.

--COMMANDS: OPEN (Open TCP connection to remote peer & authenticates sending BGP peer) ++ UPDATE (advertises new path or withdraws old) ++ KEEPALIVE (keeps connection alive in absense of updates. Also ACKs OPEN requests) ++ NOTIFICATION (reports errors, closes connection)

--Inter/Intra AS routing may be different because: policy and performance(policy dominates inter, performance dominates intra) ++ scale (hierarchical routing reduces table size/update traffic)

-Hot-Potato-Routing = Simple BGP routing chooses local gateway that has the least intra-domain cost. (Inter-domain cost doesn't matter - think business)

----Adding outside prefix in Hot-Potato:

---------1) Learn from inter-AS protocol that subnet x is reachable via multiple gateways

---------2) Use routing info from intra-AS protocol to determine costs of least-cost paths to each of the gateways.

---------3) Hot potato routing: Choose the gateway that has the smallest least cost.

---------4) Determine from forwarding table the interface I that leads to least-cost gateway. Enter (x,I) in forwarding table.

-Achieving Policy via advertisements - ISP routes only its own customers ++ Customer networks don't want to act as providers & thus don't advertise paths.

-**BGP SELECTS ROUTE BASED ON**: (This is the *real* BGP routing algorithm ++ incorporates hot potato)

--1) Policy/Local Preference - Highest policy preference routes are selected.

--2) Shortest AS-PATH

--3) Closest NEXT-HOP router: hot potato routing

--4) Other/Additional Criteria

TEXT EXPANDED POINTS:

-**OSPF vs BGP** = OSPF is for use within an AS. BGP is for use between AS'es. BGP is THE inter-AS protocol and the most important protocol (rivaled only by IP protocol.)

-Routes packets based on CIDRized prefixes (representing subnets/collections of subnets)

-A prefix with its attributes is called a 'route'

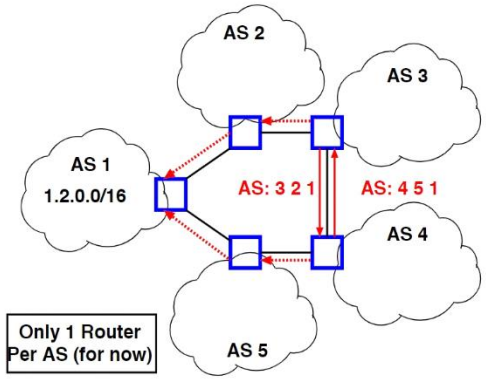
----AS-PATH - BGP attribute that contains the list of ASs through which the advertisement has passed. When prefix is passed to an AS it adds its ASN to the **front** of existing list in the AS-PATH.

----NEXT-HOP is the IP address of the router interface that begins the AS-PATH.

-BGP provides 'anycast' service (finds nearest source for content - i.e. enables a CDN) ++ (Treats paths that lead to same content as different paths to same location even if different IRL locations) (CDNs use different algorithms in practice, but DNS uses 'anycast')

BONUS: Handout

BGP Example



-**BGP and Policy**

-----BGP provides capability for enforcing various policies

-----Policies are not part of BGP - they are provided to BGP as configuration information

-----BGP enforce policies by choosing paths from multiple alternatives and controlling advertisements to other AS's.

-**BGP Path Selection**

----Policies determined by path selection

----Information based on path attributes

----Attributes + external policy information

-Customer/Provider AS Relationship:

-----Customer pays for connectivity:

-------a) E.g University of Houston contracts with AboveNet and TW Telecom

-------b) Customer is stub, provider is transit

-----Many customers are multi-homed:

-------a) E.g. AboveNet connects to Level3, Cogent...etc...

-----Typical Polices:

-------a) Provider tells all neighbors how to reach customer

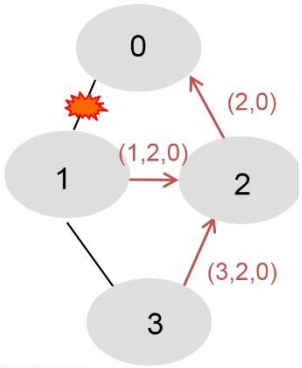
-------b) Provider prefers routes from customers ($$$)

-------c) Customer does not provide transit service.

PART 1 **PART 2**

Routing Change: Path Exploration

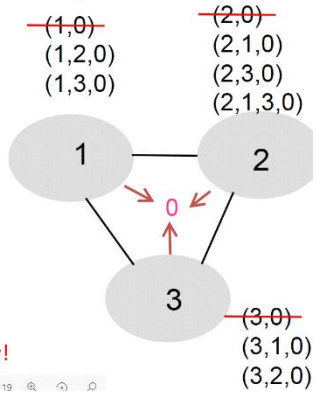
- AS 1
  - Delete the route (1,0)
  - Switch to next route (1,2,0)
  - Send route (1,2,0) to AS 3
- AS 3
  - Sees (1,2,0) replace (1,0)
  - Compares to route (2,0)
  - Switches to using AS 2



Routing Change: Path Exploration

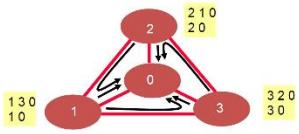
- Initial situation
  - Destination 0 is alive
  - All ASes use direct path
- When destination dies
  - All ASes lose direct path
  - All switch to longer paths
  - Eventually withdrawn
- E.g., AS 2
  - (2,0) → (2,1,0)
  - (2,1,0) → (2,3,0)
  - (2,3,0) → (2,1,3,0)
  - (2,1,3,0) → null
- Convergence may be slow!

OR NEVER! - (Open research)



Unstable Configurations

- Due to policy conflicts



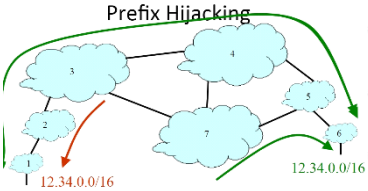
BGP Security Goals

- Confidential message exchange between neighbors
- Validity of routing information
  - Origin, Path, Policy
- Correspondence to the data path

Origin: IP Address Ownership and Hijacking

- IP address block assignment
  - Regional Internet Registries (ARIN, RIPE, APNIC)
  - Internet Service Providers
- Proper origination of a prefix into BGP
  - By the AS who owns the prefix
  - … or, by its upstream provider(s) in its behalf
- However, what's to stop someone else?
  - Prefix hijacking: another AS originates the prefix
  - BGP does not verify that the AS is authorized
  - Registries of prefix ownership are inaccurate

Prefix Hijacking

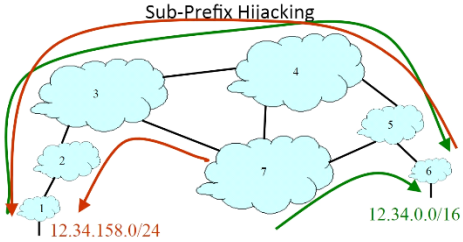


- Consequences for the affected ASes
  - Blackhole: data traffic is discarded
  - Snooping: data traffic is inspected, and then redirected
  - Impersonation: data traffic is sent to bogus destinations

Hijacking is Hard to Debug

- Real origin AS doesn't see the problem
  - Picks its own route
  - Might not even learn the bogus route
- May not cause loss of connectivity
  - E.g., if the bogus AS snoops and redirects
  - … may only cause performance degradation
- Or, loss of connectivity is isolated
  - E.g., only for sources in parts of the Internet
- Diagnosing prefix hijacking
  - Analyzing updates from many vantage points
  - Launching traceroute from many vantage points

Sub-Prefix Hijacking



- Originating a more-specific prefix
  - Every AS picks the bogus route for that prefix
  - Traffic follows the longest matching prefix

How to Hijack a Prefix

- The hijacking AS has
  - Router with eBGP session(s)
  - Configured to originate the prefix
- Getting access to the router
  - Network operator makes configuration mistake
  - Disgruntled operator launches an attack
  - Outsider breaks in to the router and reconfigures
- Getting other ASes to believe bogus route
  - Neighbor ASes not filtering the routes
  - … e.g., by allowing only expected prefixes
  - But, specifying filters on *peering* links is hard

5.5 The SDN Control Plane:

KEY SLIDE POINTS:

-Alternative to traditional, distributed, per-router control approach ++ Traditional approach had routing components in EVERY router

-SDN control plan has **Remote Controller** install forwarding tables in routers.

----Easier network management ++ greater flexibility ++ avoid router misconfigurations ++ Can 'program' routers (centralized programming easier than distributed) ++ Easy traffic engineering

----Open-interface leads to fast growth

SDN Structure:

----1) Generalized flow-based forwarding (I.e. OpenFlow) (Packet forwarding can be based on any number of header field values)

----2) Control, Data plane separation (Data Plane = Match + Action, Control Plane

----3) Control plane functions external to data-plane switches (Software on remote servers controls ++ 2 components: SDN controller & network-controlled applications)

----4) Programmable control applications (end-end paths, access control, server load balancing, etc)

----5) Distributed - Implemented on set of servers (fault tolerance ++ availability ++ performance)

-Data Plane Switches (lower level)

----Fast, simple commodity switches implement 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 can be controlled and what can't.

----Protocol for communicating with controller. (Openflow)

-SDN Controller (mid level)

----Communication ("Southbound" interface keeps controller aware of network state && allows for exchanging information)

----Network-wide management layer (Configuring flow tables in switches ++

----Maintains network state information

----Interacts with network control applications 'above' via northbound API (read/write network, state, and flow tables.

----Interacts with network switches 'below' via southbound API

----Implemented as distributed system for: performance, scalability, fault-tolerance, robustness

-Networks control apps: (top level)

----"Brains of control: Implements control functions used by lower level services

----"Unbundled" - Can be provided by 3rd party, distinct from routing vendor or SDN controller.

-SDN CONTROLLER *COMPONENTS* (Extention of mid)

----Interface layer to network control app: abstractions API (Network graph, RESTful API, intent)

----network-wide state management : state of networks links, switches, services: a distributed database (statistics, flow tables, Link-state info, host info, switch info)

----communication: communicate between SDN controller and controlled switches (OpenFlow, SNMP)

-SDN Challenges

----hardening the control plane: dependable, reliable, performance-scalable, secure distributed system - (hardening the control plane: dependable, reliable, performance-scalable, secure distributed system) ++ (dependability, security: “baked in” from day one?)

----networks, protocols meeting mission-specific requirements (e.g., real-time, ultra-reliable, ultra-secure)

----Internet-scaling: beyond a single AS

----SDN critical in 5G cellular networks

-OpenFlow Protocol

----Operates between controller and switch ++ TCP used to exchange messages ++ 3 classes of OpenFlow messages (Controller to switch ++ asynchronous ++ symmetrics (misc)) ++ Distinct from OpenFlow API (API used to specify generalized forwarding actions)

----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.)

TEXT EXPANDED POINTS:

-ODL (Northbound = REST/RESTCONF/NETCONF APIs. Mid = Ehanced Services (AAA, Devices) ++ Basic Network Functions. Southbound = config and operational data store ++ messaging ++ (OpenFlow, NETCONF, SNMP, OSVDB))

-ONOS - (Northbound = REST API && Intent) ++ (Mid = Hosts, Paths, Flow Rules, Devices, Links, Statistics, Topology) ++ (Soundbound = Device, Link, Host, Flow, Packet, OpenFlow, NETCONF, OVSDB)

----Unique features - Intent framework transparently requests high level services ++ Distributed mid-level core ++ Southbound features are abstracted & transparent.

5.6 ICMP: The Internet Control Message Protocol:

KEY SLIDE POINTS:

-ICMP: "Internet Control Message Protocol" - Used by internet/routers to communicate network level info. (error reports, unreachable hosts, port & protocol stuff ++ echo/ping - packets with increasing TTL - record RTTs after packet arrives at source.)

-Commands:

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

5.7 Network Management and SNMP:

KEY SLIDE POINTS:

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."

-Components/Framework of Network Management:

----Managing Server (Application, typically with network managers.)

----Network Management Protocol (Used by manager to: query, configure, manage device, manage data server, set events)

----Managed Device (Equipment with configurable components)

----Data (*Device state*, configuration data, operational data, device statistics)

-Network Operator Approaches to Management:

----1) CLI (command line interface)

----2) SNMP/MIB (operator queries/sets devices data (MIB) using Simple Network Management Protocol (SNMP))

----3) 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

SNMP

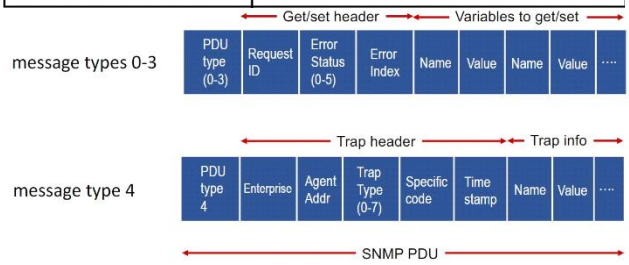
-SNMP Protocol:

----2 ways of conveying MIB info & commands:

--------1) Request/Response

--------2) "Trap Message" (When some event/error occurs on a managed device, it messages controller)

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



-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

NETCONF

-1) goal: actively manage/configure devices network-wide

-2) goal: actively manage/configuredevices network-wide

----actions: retrieve, set, modify, activate configurations

----atomic-commit actions over multiple devices

----query operational data and statistics

----subscribe to notifications from devices

-3) remote procedure call (RPC) paradigm

----NETCONF protocol messages encoded in XML

----Exchanged over secure, reliable transport (e.g., TLS) protocol.

-**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

-**YANG**:

----Data modeling language used to specify structure, syntax, semantics of NETCONF network management data

----Yang Description generates XML document describing device, capabilities

----Expresses constraints in valid NETCONF configuration ++ ensure NETCONF configurations satisfy correctness, consistency constraints

5.BONUS) **gRPC DISCUSSION:**

-**gRPC INFO**:

--RPC = REMOTE PROCECURE CALLS - One process calls a function just like it would call a local function. The RPC protocol ships those calls to a remote process - called server - where execution happens. It sends response back to client.

----Supports many languages

----Extremely fast and efficient

----Open Source software

-gRPC is Performant and Efficient

-BUILT ON HTTP/2 (Streaming, multiplexing, header compression, binary framing)
----Main objective is to reduce webpage load times.
----Load times improved by binary framing - (HTTP/1.1 - It's based on request/response model) ++ (HTTP/2 - data customer wants to send is converted into binary form - much more efficient!)
----Binary framing can have multiple streams ++ A stream carries multiple message corresponding to logical request/response to message.

----Frame is smallest unit of transition in HTTP/2. Every frame contains data of one type.

----HTTP/2 gets its performance boost because of multiple streams working in parallel.

-HIERARCHY - Frame (Smallest unit) -> Message (multiple frames + header/message/control frames) -> Streams (Client receives multiple messages from one file - can have multiple streams open) -> Data from the streams is multiplexed in TCP

-Uses Protocol Buffers:
----Serialization time, compact wire format, client/server compute time, network throughput
----Very high performing serialization time that gRPC leverages

--Protocol Buffers - **Language/Platform neutral** mechanism for serializing structured data
----Specifies syntax of IDL (high level language - write definition and compiler generates code for multiple languages)
----Proto compiler generates code for multiple languages
----Languages specific runtime libraries
----Serialization format
----Forward and backward compatibility

-Continually benchmarked and improved:

----Benchmarking running on containers nowadays.

-gRPC is General Purpose

-Speaks many languages

-Continuous tests for interoperability among many languages

-Runs everywhere
----Now supports Linux on ARM

-gRPC is Extensible:

-Load Blaancing
-Security
-Observability
-Service Discovery
-Codecs
-Compression
-Transports
-And more...

-gRPC NOTES:

-gRPC does handle:
----Serialization and
----Managing the execution of the task
----Finding the server(s) that execute the remote function
-gRPC does NOT handle:
----Routing the packets from server to client

Using PRC is recommended when:
----A corporate application that needs to communicate with HR and sales applications.
----Centralized Authentication service responsible for validating the credential for multiple applications
----In app purchase service for mobile applications
Using RPC is NOT recommended when:
----A CDN server that serves static files for multiple corporate applications

RPC Facts:
----It is NOT bound to using HTTP as the communication protocol (TRUTH: _)
----In this architecture, the client is looking for service instead of server.
----REST API is NOT an example of RPC (TRUTH: _)
----RPCs do NOT only use JSON to transfer messages (TRUTH: _)

-gRPC uses the HTTP 2 protocol between the sender and the client.

-gRPC Deployment Example (Each Task is running in a different language)

-Why is gRPC better than REST/JSON?

----Performance
----Built-in Security
----Built-in Interop Between Languages
----Type Safety
----Streaming Support

-RPC types:

Unary - Client sends one request and receives one response from server

Server streaming - Client sends one request and receives many responses from server (EXAMPLE: Want to watch server stock prices)

Client streaming - Client sends many requests and receives one response from server (EXAMPLE: Client wants to upload file to storage server - sends bits until done and server responds to say operation succeeded/failed)

Bi-Directional Stream - Client sends many requests and receives many responses from server (EXAMPLE: Online chat)

Ch 5) **FORMULAS AND KEY CONCEPTS**:

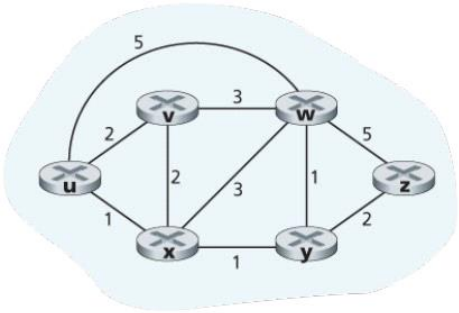
1) **Dijkstra's Algorithm Formula**

notation

- $c_{x,y}$: direct link cost from node x to y; $= \infty$ if not direct neighbors
- $D(v)$: *current* estimate of cost of least-cost-path from source to destination v
- $p(v)$: predecessor node along path from source to v
- N': set of nodes whose least-cost-path *definitively* known

Dijkstra's link-state routing algorithm

```
1  Initialization:
2   N' = {u}                    /* compute least cost path from u to all other nodes */
3   for all nodes v
4    if v adjacent to u         /* u initially knows direct-path-cost only to  direct neighbors  */
5       then D(v) = c_{u,v}     /* but may not be minimum cost!                                  */
6    else D(v) = ∞
7
8   Loop
9    find w not in N' such that D(w) is a minimum
10   add w to N'
11   update D(v) for all v adjacent to w and not in N' :
12       D(v) = min ( D(v),  D(w) + c_{w,v} )
13   /* new least-path-cost to v is either old least-cost-path to v or known
14   least-cost-path to w plus direct-cost from w to v */
15  until all nodes in N'
```



step	N'	D (v), p (v)	D (w), p (w)	D (x), p (x)	D (y), p (y)	D (z), p (z)
0	u	2, u	5, u	1,u	∞	∞
1	ux	2, u	4, x		2, x	∞
2	uxy	2, u	3, y			4, y
3	uxyv		3, y			4, y
4	uxyvw					4, y
5	uxyvwz					

-For each iteration, find the lowest cost, previous node from the set of nodes N'

-Ties are broken arbitrarily

-Note in the HW, you have to fill up the entire table - in this version, you don't (???)

-Time complexity O(n^2) in our method - O(n log(n)) if more efficient implementation.

-Overall message complexity O(n^2)

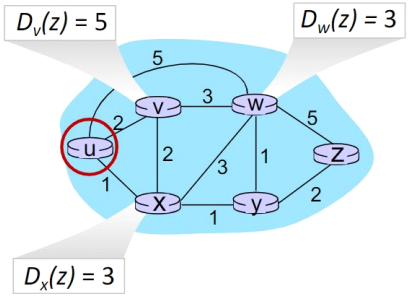
-Oscillations possible.

2) **BELLMAN-FORD EQUATION**

Bellman-Ford equation

Let $D_x(y)$: cost of least-cost path from x to y.
Then:

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



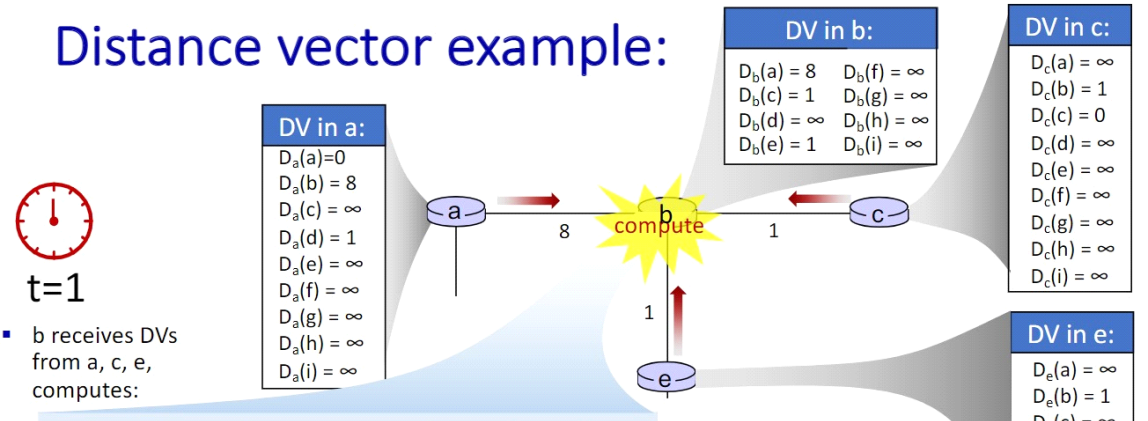
Bellman-Ford equation says:

$$D_u(z) = \min \{ c_{u,v} + D_v(z), \; c_{u,x} + D_x(z), \; c_{u,w} + D_w(z) \}$$
$$= \min \{2 + 5, \; 1 + 3, \; 5 + 3\} = 4$$

node achieving minimum (x) is next hop on estimated least-cost path to destination (z)

All nodes:
- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors

Distance vector example:



t=1

- b receives DVs from a, c, e, computes:

$D_b(a) = \min\{c_{b,a}+D_a(a), c_{b,c}+D_c(a), c_{b,e}+D_e(a)\} = \min\{8,\infty,\infty\} = 8$
$D_b(c) = \min\{c_{b,a}+D_a(c), c_{b,c}+D_c(c), c_{b,e}+D_e(c)\} = \min\{\infty,1,\infty\} = 1$
$D_b(d) = \min\{c_{b,a}+D_a(d), c_{b,c}+D_c(d), c_{b,e}+D_e(d)\} = \min\{9,2,\infty\} = 2$
$D_b(e) = \min\{c_{b,a}+D_a(e), c_{b,c}+D_c(e), c_{b,e}+D_e(e)\} = \min\{\infty,\infty,1\} = 1$
$D_b(f) = \min\{c_{b,a}+D_a(f), c_{b,c}+D_c(f), c_{b,e}+D_e(f)\} = \min\{\infty,\infty,2\} = 2$
$D_b(g) = \min\{c_{b,a}+D_a(g), c_{b,c}+D_c(g), c_{b,e}+D_e(g)\} = \min\{\infty,\infty,\infty\} = \infty$
$D_b(h) = \min\{c_{b,a}+D_a(h), c_{b,c}+D_c(h), c_{b,e}+D_e(h)\} = \min\{\infty,\infty,2\} = 2$
$D_b(i) = \min\{c_{b,a}+D_a(i), c_{b,c}+D_c(i), c_{b,e}+D_e(i)\} = \min\{\infty,\infty,\infty\} = \infty$

DV in b:
$D_b(a) = 8$ $D_b(f) = 2$
$D_b(c) = 1$ $D_b(g) = \infty$
$D_b(d) = 2$ $D_b(h) = 2$
$D_b(e) = 1$ $D_b(i) = \infty$

This example is representative for the process ^ Rotate between nodes and update based on the minimum value.

-t = 0 means each node only has its own state (I.e. It's own distances to its neighbors). Number of hops = t.

-From time to time, each node sends its distance vector algorithm to neighbors.

-When x receives DV estimate, it updates its own DV by BF equation

-"Good news travels fast" and "Bad news travels slow" (Count to infinity problem - Solved by "Poisoned Reverse" - if x routes through y to get to z, x will lie to y that x -> = infinity. This holds until the path from x -> y -> z is no longer fastest for x. The x tells the truth.)

-Iteractive, asynchronous, and distributed.

Ch 5) **GLOSSARY**:

-LS = Link State Algorithm

-DV = Distance Vector Algorithm

-OSPF = Open Shortest Path First. Covered in 5.3

-BGP = Border Gateway Patrol = THE default inter-domain routing protocol

----BGP ROUTE - A prefix with its attributes is called a 'route'

----AS-PATH - BGP attribute that contains the list of ASs through which the advertisement has passed. When prefix is passed to an AS it adds its ASN to the existing list in the AS-PATH

----NEXT-HOP = the IP address of the router interface that begins the AS-PATH.

-OpenFlow = Match + Action.

-ODL = "Open Daylight" SDN Controller.

-ONOS = SDN Controller

-ICMP = The Internet Control Message Protocol:

-SNMP =

-YANG = Data modeling language used to specify structure, syntax, semantics of NETCONF network management data

NETCONF -

-AS = "Autonomous Systems" = A network of networks where each AS is a mini-network.

-RIP = Routing Information Protocol = Classic DVs: DVs exchanged every 30 seconds. No longer widely used. (Apparently still common according to the HW - should we trust text or HW?)

-EIGRP = Enhanced Interior Gateway Routing Protocol = DV-based and former Cisco property.

-OSPF = Open Shortest Path First = LS routing ++ IS-IS protocol (ISO standard, not RFC standard) aka OSPF

-OVSDB = The Open vSwitch Database Management Protocol (OVSDB) is used to manage data center switching, an important application area for SDN technology.

==========

==========

CHAPTER 6 – LINK LAYER

==========

==========

NETWORKS FINAL CHAPTER 6 NOTES

6.0) **GOALS**:

-Learn how packets are sent across individual links in end-end communication path.

----How are they encapsulated in link-layer frames for transmission over single link? ++ What are the link layer protocols used in different links? ++ How are transmission conflicts in broadcast resolved ++ What is link-layer addressing & how does it operate with network-layer addressing? ++ What is the difference between a switch and a router?

----2 types of Link-Layer channels - 1) broadcast channels (multiple hosts & needs MAP to coordinate) ++ 2) Point-to-Point communication

----Error correction ++ VLANs ++ Multiple-Access-Networks ++ Switched LANs (Ethernet most prevelant)

6.1) **Introduction to the Link Layer**

KEY SLIDE POINTS:

-Terminology: Node ++ Link ++ Link-Layer-Frame (SEE GLOSSARY)

-Datagram may be transferred by many different link protocols between ends ++ each protocol provides different services

-SERVICES - **Framing & Link Access** (header, trailer, datagram ++ Channel Access if shared medium ++ MAC address identifies source) ++ **Reliable Delivery** (Adjacent nodes - both link & end-end reliably because no guaranteed all links error-check.))

-----SERVICES CONT - **FLOW CONTROL** (Pacing between adjacent sending/receiving nodes) ++ **ERROR DETECTION** (Caused by signal attenuation, noise ++ Receive detects errors, signals retransmission, or drops frame) ++ **ERROR CORRECTION** (Receiver identifies & corrects error ++ No retransmission) ++ **HALF-DUPLEX & FULL-DUPLEX** (Half-duplex = nodes at both end can transmit, not at same time though.)

-WHERE? - Link layer implemented in every host in NIC (Network Interface Card) or chip. (Includes physical layer) ++ Attaches host system buses ++ Combination of hardware, software, firmware.

-IMPELMENTATION - **Sender** (encasulates datagram in frame ++ adds error-checking bits, reliable data transfer, flow control, etc) ++ **Receiver** (Looks for errors, reliable data transfer, flow control, etc ++ extracts datagram to upper layer)

6.2) **Error-Detection and Correction Techniques**

KEY SLIDE POINTS:

-EDC (Header Field - Error Detection and correction bits) ++ larger EDC = better detection!

-Parity Checking

----Single bit that ensures the number of 1s is either even/odd in rows and/or columns based on policy)

----If row or column breaks rule, error has occurred

----Errors may easily go undetected

----FEC = "Forward Error Correction" - The ability of the receiver to both detect and correct errors is known as forward error correction

-Internet Checksum

----Treats UDP segment as 16-bit integers

----Sender adds the 16-bit integers together (If carry at far end, wrap it around to LSB on right) ++ Take 1s compliment of the result ++ Store in checksum field

----Receiver adds the 16-bit integers together & adds checksum - should get all zeros - otherwise error!

-Cyclic Redundancy Check

-Checksum is fast and simple - CRC is more complex but can be performed in dedicated hardware in adapters in link-layer.

-GENERATOR - sender and receiver agree on (r + 1) generator G pattern.

-Append r additional bits R to D such that the resulting (d + r) is easily divisible by G. (No remainder) using modulo-2 arithmetic.

-(Check FORMULA #1)

-If remainder exists - ERROR!

-DETECTS BURST ERRORS OF FEWER THAN (r + 1) BITS!

----Detects burst errors of greater than (r + 1) bits with probability 1 - 0.5^r

----Detects all odd numbers of bit errors.

-CRC FORMULA:

----D*(2^r) XOR R = nG

----D*(2^r) = nG XOR R (Alt)

----R = remainder[d*(2^r) /G]

6.3) **Multiple Access Links and Protocols**

KEY SLIDE POINTS:

-Two link types

----Point to Point

----Broadcast (Shared wire/medium) ++ Needs MAP (Multiple Access Protocol) to avoid & deal with **collisions**. (Simultaneous transmission = information lost)

-Ideal MAP: (Given channel of rate R bps)

----When one node wants to transmit, it can send at rate R.

----When M nodes want to transmit, each can send at average rate R/M

----Fully Decentralized (no special 'coordinator' node nor clock/slot synchronization)

----Simple

-MAC protocols: Taxonomy

--1) Channel Partitioning

----Divide channel into smaller pieces (time slots (TDM), frequency (FDM), code (CSMA)) and allocate pieces to nodes

----EXAMPLES: TDMA, FDMA, CSMA & CSMA/CD

----a) **TDMA** allows access in 'rounds' ++ must wait for turn always ++ inefficient because unused slots go idle.

----b) **FDMA** divides access into frequency bands ++ access evenly divided ++ inefficient when few stations have to speak.

----c) **CDMA** allows nodes to transmit simultaneously and restores the message based on the sender's code. Only works if codes are orthogonal. (Mainly chapter 7)

--2) Random Access

----Channel not divided ++ transmits at full rate ++ allow collisions but recovers. (Collision detection & collision avoidance key)

----EXAMPLES: ALOHA ++ slotted ALOHA ++ CSMA ++ CSMA/CD ++ CSMA/CA (wifi)

----a) **SLOTTED ALOHA** - When node obtains fresh frame, transmit in next slot ++ if no collision, can send more packets ++ if collision, node retransmits frame in subsequent slots with probability p until success.

-Assumes all frames are same size, time divided into equal slots, Nodes transmit only at slot beginning, nodes are synchronized, if 2+ nodes collide then all nodes detect collision.

-Randomizing retransmission prevents deadlocks.

-Pros: Single node can continually transmit at full rate ++ highly decentralized - only slots in nodes need to be in sync ++ simple

-Cons: Wasted slots (collisions and idle) ++ nodes may detect collisions in less time than to transmit a packet ++ clock synchronization

-Efficiency 1/3 = 0.37%

----b) **Pure ALOHA**

-No slot synchronization.

-Collision frequency increases

-Efficiency = 18%

----c) **CSMA** (Carrier Sense Multiple Action)

-Simple CSMA - listens before transmission - if channel idle, send frame - if busy, wait.

-CSMA/CD - CSMA with Collision Detection - If collision detected, abort & retransmit at random later time ++ Easy in wired, hard in wireless.

--Collisions can still occur - transmission delay means channel might incorrectly be sensed as idle.

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, …, 2^m-1}*. NIC waits *K*·512 bit times, returns to Step 2
   - more collisions: longer backoff interval

- T_{prop} = max prop delay between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame

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

- efficiency goes to 1
  - as t_{prop} goes to 0
  - as t_{trans} goes to infinity
- better performance than ALOHA: and simple, cheap, decentralized!

-The set of k in BEB grows exponentially from {0} --> {0, 1} --> {0,1,2,3} ---> etc...

--3) <u>Turning-Taking</u>

-Nodes take turns, but nodes with more to send can take longer turns

-Aims for best of both worlds - when one node active, transmit at R. When M nodes active, transmit at R/M.

----a) **Polling**

-Master node invites other nodes to take turns ++ Used mostly with dumb devices

-CONCERNS: Polling overhead ++ Latency ++ Single point of failure(master)

----b) **Token Passing**

-Control token passed from one node to next sequentially.

-Token message

-Concerns: Token overhead, Latency, Single point of failures (token)

6.4) **Switched Local Area Networks**

KEY POINTS:

1) Addressing & ARP

-**MAC Addressing**

----Similar to 32-bit IP Address used in layer-3 (internet)

----Portable, permanent MAC addresses used 'locally' to get frame from one interface to another physically-connected interface (same subnet, in IP-addressing sense)

----48-bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable

----e.g.: 1A-2F-BB-76-09-AD (hexadecimal base-16. Each 'numeral' represents 4 bits)

----Each interface on LAN has both IP address and MAC address.

----IEEE ensures uniqueness - manufacturers buy address space.

----Portable

-**ARP - Address Resolution Protocol**

----Solves: How to find MAC address given IP address?

----Each IP node (host/router) on LAN has table with IP/MAC address mapping for *some* nodes (Each entry has Time To Live (TTL))

----For A to find B's address with ARP: Sends broadcast query with B's IP address ++ B replies with ARP response giving MAC address ++ A receives B reply and adds B to ARP table

--a) Routing to different subnet: Sending datagram from A to B through R. (Assume A knows R's IP and MAC)

----a1) A sends (A-->B) datagram to R's MAC address ++ R determines outgoing interface & passes datagram to link layer ++ Creates link-layer frame to carry A-->B with B's MAC address ++ Transmits link-layer frame.

2) Ethernet

-Dominant Wired LAN technology. (First widely used ++ Cheap, Simple, Fast ++ Multiple speeds, single chip)

-**Topology**:

----(Old) Bus - Connects all nodes in same collision domain (Collision possible)

----(Current) Switched - Modern ++ Active link-layer switch in center ++ Each 'spoke' runs separate protocol (no collisions)

-**Frame Structure**

-Preamble (Used to synchronize sender/reciever clock rates)

-Addresses (byte source, destination MAC addresses ++ if matching destination address or broadcast address (ARP) data is passed to network layer protocol) ++ else packet discarded)

-Type (Indicates higher layer protocol ++ used to demultiplex at receiver)

-CRC (Cyclic Redundancy Check - frame dropped if ereror)

-**Additional Ethernet INFO**

----Connectionless (no handshaking between sending and receiving NICs)

----Unreliable (receiving NIC doesn't send ACKs or NAKs to sending NIC ++ data dropped in frames only recovered if sender uses higher layer rdt, else lost)

----Ethernet's MAC protocol - Unslotted CSMA/CD with binary backoff

----Many standards: Varying speeds, different physical media.

3) Switches

-General

----Link-Layer device - Stores & forwards Ethernet Frames ++ examine incoming frame's MAC address ++ Selectively forward frame to one-or-more outgoing links when frame is to be forwarded on segment ++ Alternately filter/discard frame.

----Uses CSMA/CD to access segment

----Transparent - host unaware of switches

----Plug & Play, Self-Learning - Switches do not need to be configured

-**Switches: Multiple Simultaneous Transmissions**

----Hosts have dedicated, direct switch connection

----Switches buffer packets

----Ethernet protocol used on each incoming link (No collisions ++ fully duplex ++ Each link is own collision domain)

-**Switch forwarding table**

----Used to find reachable hosts ++ Contains (MAC address of host, interface to reach host, time stamp)

----Entries are created/maintained via Self-Learning ++ Host learns which hosts can be reached through which interface when host sends frame.

----Records sender/location pair in switch table

----If destination known: Send packet ++ If destination unknown, flood

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 */
```

-**Switches: Properties**

----Elimination of Collisions - Significant performance improvement

----Heterogenous Links - Links are isolated from one another, allowing different technologies to be mixed

----Management - Enhanced Security ++ Can automatically disconnect malfunctioning adapter ++ Gathers statistics.

-**Switches vs Routers**

(Both 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

4) <u>VLANs</u>

-Port-based VLAN: switch ports grouped (by switch management software) so that singlephysical switch operates as multiple virtual switches (Can also define VLAN based on MAC addresses of endpoints instead of ports)

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

-Used to solve scaling ++ efficiency, security, privacy issues ++ Logical addressing (I.e. Person from CS department moves to EE physical space but wants to remainlogically attached to CS switch)

-**Characteristics**

----Traffic Isolation (Frames in one VLAN can only reach ports within the VLAN ++ Can also define VLAN based on MAC addresses of endpoints instead of ports)

----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)

----Trunk Port (Carries frames between VLANS defined over multiple physical switches ++ Frames forwarded between VLANs can't be vanilla 802.1 frames - Needs VLAN ID Info ++ 802.1q adds header fields for frames forwarded between trunk ports)

6.5) **Link Virtualization: A Network as a Link Layer**

KEY SLIDE POINTS:

Multiprotocol label switching (MPLS)

-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

-AKA "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 differfrom those of IP

----Use destination andsource 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 anddestination address

----Fast Reroute - Can precompute backup routes in case of 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

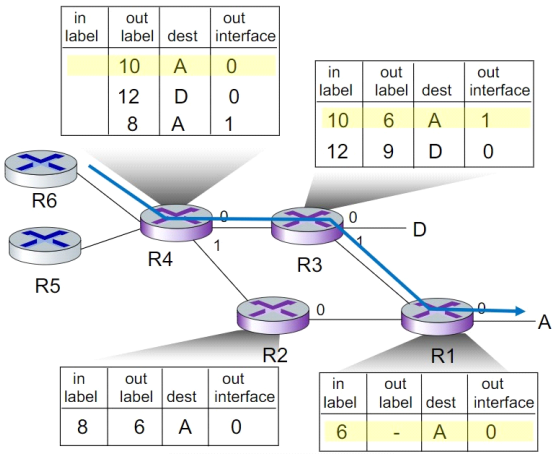


Image: R3 is advertising to R4 that it can route to destinations A and B and incoming frames with dest 10 & 12 respectively will be switched to those destinations.

6.6) **Data Center Networking**

KEY SLIDE POINTS:

Data Centers:

-Composed of 10k - 100ks of hosts, usually closely coupled and in close proximity

-Challenges - Multiple Applications, each serving massive numbers of clients ++ Reliability ++ Managing/Balancing load ++ Avoiding bottlenecks (processing, networking, data)

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) Switches (one per rack ++ 40-100Gbps Ethernet to blades)

-Server Racks (20-40 server blades: hosts)

Datacenter networks: Multipath

-Rich connections between switches & racks (increased throughput & reliability)

----(Web of redundant connections between levels)

Datacenter networks: application-layer routing

-Load balancer receives external client requests --> Directs workload to data center --> Returns results to external client (Data center is transparent to client.)

Datacenter networks: protocol innovations

-Goals (Cost reduction, Centralized SDN Management, Virtualization)

-Link Layer: RoCE: remote DMA (RDMA) over Converged Ethernet

-Transport Layer: **ECN** (explicit congestion notification) used in transport-layer congestion control (DCTCP, DCQCN) ++ Eperimentation with **hop-by-hop** (backpressure) congestion control

----Transport times are very small and congestion protocals must quickly catch even small loss.

-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.

6.7) **Retrospective: A Day in the Life of a Web Page Request**

KEY SLIDE POINTS:

(FINISH IF TIME)

Ch 6) **FORMULAS**:

1) **Cyclic Redundancy Check**

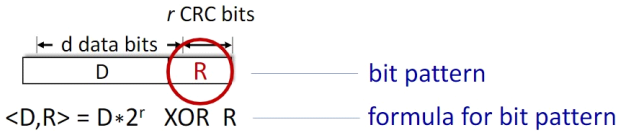
-Checksum is fast and simple - CRC is more complex but can be performed in dedicated hardware in adapters in link-layer.

-GENERATOR - sender and receiver agree on (r + 1) generator G pattern.

-Append r additional bits R to D such that the resulting (d + r) is easily divisible by G. (No remainder) using modulo-2 arithmetic.

-If remainder exists - ERROR!

- D: data bits (given, think of these as a binary number)
- G: bit pattern (generator), of *r+1* bits (given)



goal: choose *r* CRC bits, R, such that <D,R> exactly divisible by G (mod 2)

- receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
- can detect all burst errors less than r+1 bits
- widely used in practice (Ethernet, 802.11 WiFi)

EXAMPLE

Figure 6.7 illustrates this calculation for the case of $D = 101110$, $d = 6$, $G = 1001$, and $r = 3$. The 9 bits transmitted in this case are 101 110 011. You should check these calculations for yourself and also check that indeed $D \cdot 2^r = 101011 \cdot G$ XOR R.

We want:

$D{\cdot}2^r$ XOR $R = nG$

or equivalently:

$D{\cdot}2^r = nG$ XOR R

or equivalently:

if we divide $D{\cdot}2^r$ by G, want remainder R to satisfy:

$$R = \text{remainder}\left[\frac{D{\cdot}2^r}{G}\right]$$

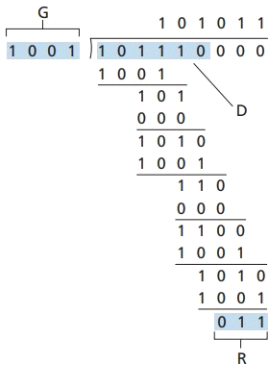


Figure 6.7 ♦ A sample CRC calculation

--CRC TEXT FORMULA:

----D*(2^r) XOR R = nG -----> D*(2^r) = nG XOR R (Alt)

----(Alt) R = remainder[d*(2^r) /G] by modulo %2 binary division

--Module %2 binary division:

----Standard division but with no carries. Use XOR instead of standard subtraction.

----Remainder = error!

-(To generate problem - append L - 1 bits where L = Divisor)

-DETECTS BURST ERRORS OF (r-1) BITS!

Ch 6) **GLOSSARY**:

ARP Packet = An ARP Packet queries all the other hosts and routers on the subnet to determine the MAC address corresponding to the IP address that is being resolved. (Between Network and Link layers)

DOCSIS = data over cable service interface specificaiton (Uses FDM, TDM, and Random Access)

ECMP = qual Cost Multi Path Routing - Prforms a randomized next-hop selection along the switches between source and destination

EDC = Header Field - Error Detection and correction bits

FEC = "Forward Error Correction" - The ability of the receiver to both detect and correct errors is known as forward error correction

Link-Layer-Frame = Layer-2 Frame

MAP = Medium Access Protocol - How do multiple hosts access same broadcast channel? (Aloha ++ Slotted Aloha ++ CSMA ++ CSMA/CD ++)

MPLS = Multiprotocol label switching

NIC = Network Interface Card

NODE = Any device that runs a link-layer protocol - hosts/routers/switches/Wifi access points.

----Links = Communication channels that connect adjacent nodes along the communication path

Top of Rack (TOR) Switches - One per server rack in data center

==========

==========

CHAPTER 7 – WIRELESS AND MOBILE NETWORKS

==========

==========

7.0) **GOALS**:

Two major challenges - Wireless & Mobility (NOTE: Wireless is not always mobile)

7.1) **Introduction to the Link Layer** (564)

KEY SLIDE POINTS:

-**Wireless**

----Wireless Links and network characteristics

----Wifi: 802.11 wireless LANs

----Cellular networks: 4G and 5G

-**Mobility**

----Mobility management: Principles ++ Practice (4G/5G Networks ++ Mobile IP)

----Impact on Higher Principles

-Elements of a wireless network

----Wired Network Infra-Structure ++ Wireless hosts (End systems that run applications)

----Base Station: Connected to Wired Network ++ Relay between wired network & wireless hosts in 'area' (I.e. Cell towers, 802.11 access points)

----Wireless Link - "Backbone Link" ++ Connects wireless device to base station ++ Multiple Access Protocol (MAP) coordinates link access ++ Various transmission rates/distances/frequency bands

-Infrastructure Mode - Base station connects mobiles to wireless network ++ "Handoff" - Mobile changes base station providing connection into wired network

-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

-**Classification** (# hops & infrastructure/not)

	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

7.2) **Wireless Links and Network Characteristics** (568)

KEY SLIDE POINTS:

Differences from Wired Links (Challenging!)

-Decreased signal strength - "Path loss" - Radio signal attenuates as it propagates through matter

-Interference from other sources - wireless network frequencies (e.g., 2.4 GHz) shared by many devices (e.g., WiFi, cellular, motors)

-Multipath propagation: radio signal reflects off objects ground, arriving at destination at slightly different times

SNR - Signal to Noise ratio

-Larger SNR = Easier to extract signal from noise (Good)

-SNR versus BER tradeoffs:

----Higher bit transmission rate = higher BER

----Given physical layer: Increase power -> increase SNR->decrease BER

----Dynamic SNR - Given SNR: choose physical layer that meets BER requirement, giving highest throughput (SNR may change with mobility: dynamically adapt physical layer (modulation technique, rate))

----Formula: SNR = 20*log10(amplitude_signal)/(amplitude_noise)

Dilemmas:

-Multiple senders/receivers creates problems (Beyond Multiple Access)

-Hidden Terminal Problem (A - B - C ---> A and C aware of B but not each other - interference at B!)

-Signal Attenuation (Hidden terminals A and C interfere with each other at B)

Code Division Multiple Access (CDMA)

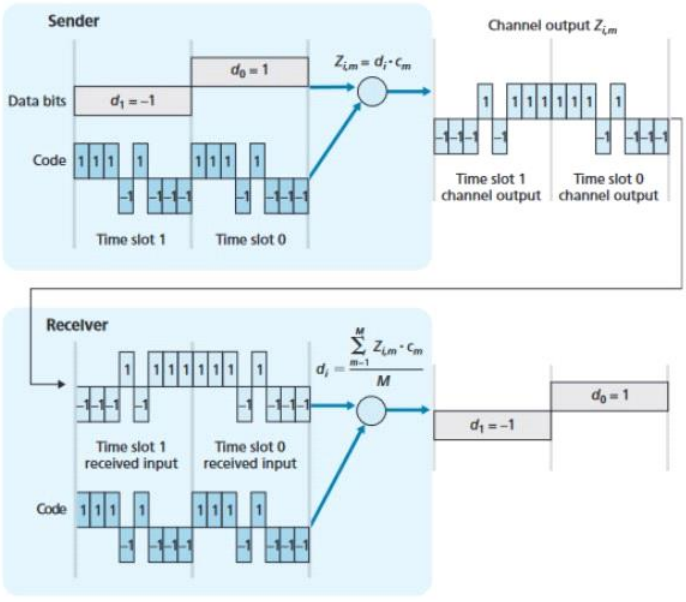
-Unique “code”assigned to each user; i.e., code set partitioning

----Users share frequency but have different "Chipping" sequence (code) to encode data

----Allows multiple senders to coexist as long as codes are **orthogonal**.

-Encoding: Inner product: (original data) X (chipping sequence)

-Decoding: Summed inner-product: (encoded data) X (chipping sequence) -- Use code on data to restore original message.



7.3) **WiFi: 802.11 Wireless LANs** (574)

KEY SLIDE POINTS:

802.11 Architecture

-All 802.11 standards use **CSMA/CA** for multiple access and have **base-station** and **ad-hoc network** versions

-Wireless host communicates with base station/"access point"

-Provides **Basic Service Set (BSS)** (aka 'cell' or "fundamental building block") in infrastructure mode that contains: Wireless hosts ++Access Point ++ Ad hoc mode (hosts only)

Standards:

IEEE 802.11 standard	Year	Max data rate	Range	Frequency
802.11 b	1999	11 Mbps	30 m	2.4 Ghz
802.11 g	2003	54 Mbps	30 m	2.4 Ghz
802.11 n (WiFi 4)	2009	600	70 m	2.4, 5 Ghz
802.11 ac (WiFi 5)	2013	3.47 Gpbs	70 m	5 Ghz
802.11 ax (WiFi 6)	2020 (expected)	14 Gbps	70 m	2.4, 5 Ghz
802.11 af	2014	35–560 Mbps	1 Km	unused TV bands (54–790 MHz)
802.11 ah	2017	347 Mbps	1 Km	900 Mhz

802.11: Channels, association

-Spectrum divided into channels at different frequencies

-802.11 has frequency range of 2.4 GHz to 2.4835 GHz (85 MHz band)

-WiFi jungle is any physical location where a wireless station receives a sufficiently strong signal from two or more APs

-802.11 defines 11 channels - (channels non-overlapping if separated by 4+ channels - only 1, 6, 11 are fully non-overlapping trio)

--AP admin chooses frequency for AP

--Interference possible if two APs use same channel

-Arriving host must **associate** with AP (Creates 'virtual wire' between itself and AP)

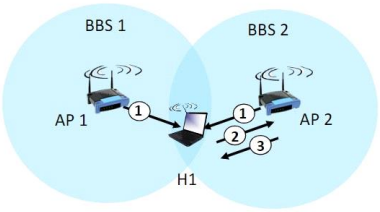
--Hosts Scans channel and listens for **beacon frames** sent by AP containing AP's name (SSID) and MAC address.

--Selects AP to associate with.

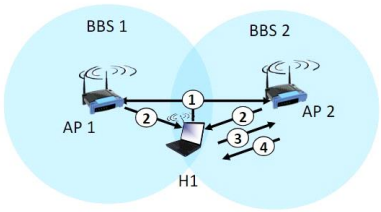
--May perform authentication (Chapter 8)

--Run DHCP to get IP address in AP's subnet

802.11: 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:
(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

-Active scanning requires the 'handshake' Associate Request frame because otherwise the AP won't know it was selected for association

-Authentication may depend on MAC address or username/password

-Authentication server usually separate from AP ++ serves many APs.

IEEE 802.11: Multiple Access Protocol

-Goal: *Avoid* Collisions (Cannot detect/stop them reliably)

-CSMA -> CSMA/**CA** -> CSMA with **Collision Avoidance** -> Can't reliably sense collisions (hidden terminal, fading, etc)

----1) If sense channel idle for DIFS (short waiting period before sending) then transmit entire frame (no CD)

----2) If sense channel busy then start random backoff time using binary exponential backoff; timer counts down ONLY while channel idle; transmit when timer expires; if no ACK, increase random backoff interval, repeat 2

----3) If frame received OK receiver returns ACK after SIFS (Short waiting period after receiving) (ACK needed due to hidden terminal problem)

----4) If source has another packet, it starts at step 2 -

----NOTE) SIFS = the Short Inter-frame Spacing = When destination receives a frame that passes the CRC, it waits a short periods of time known as SIFS and then sends back an ACK frame. If transmitting station

doesn't receive ACK within given time, it assumes error & retransments. If ACK still isn't received after fixed number of retransmissions, the transmitting station gives up and discards the frame.

----NOTE 2) CSMA/CA waits before transmitting because otherwise multiple waiting hosts send at once, resulting a in collision. Trivial in CSMA/CD but serious in wireless.

-**Avoiding Hidden Terminal Collision - RTS-CTS**:

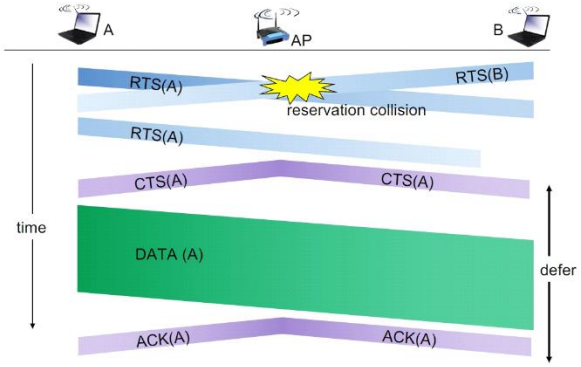
----Sender first transmits small request-to-send(RTS) packet to AP using CSMA to **reserve access**.

----RTSs may still collide with each other (but they're short)

----AP broadcasts clear-to-send CTS in response to RTS

----CTS heard by all nodes ++ sender transmits data frame ++ other stations defer transmissions

Collision Avoidance: RTS-CTS exchange



NOTE - Two nodes using 802.11 and directional antennas can act as a point to point communication system over 10s of miles.

802.11 frame: Addressing

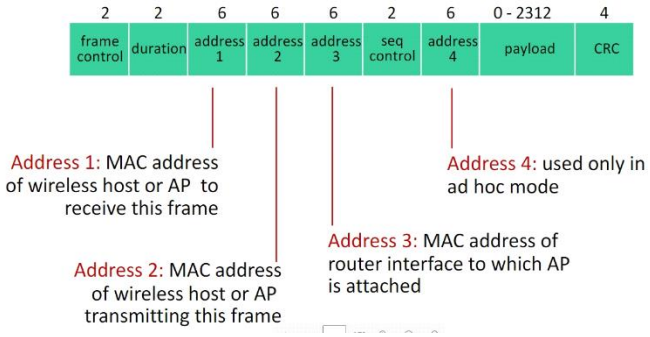
Address 1: MAC address of receiving Host or AP

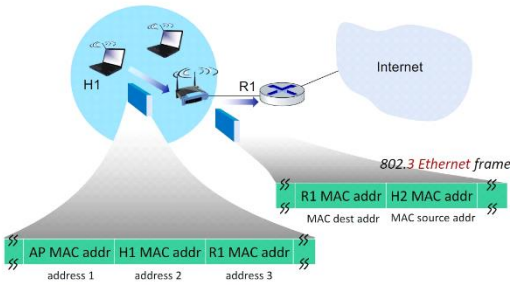
Address 2: MAC address of wireless host or AP transmitting this frame - source H1

Address 3: MAC address of router interface to which AP is attached (Tells AP which router to send to. Note that when R1 sends to H1, AP attaches R1 to Address 3)

(Address 4 used when APs forward frames to each other in ad hoc mode - not the focus)

802.11 frame: addressing





Seq control - Orders packets by number

802.11: mobility within same subnet

-CASE: Assume H1 remains in same IP subnet - then IP address can remain same

-Switch: which AP is associated with H1? (Self-learning (Ch. 6): switch will see frame from H1 and "remember" which switch port can be used to reach H1)

-How does switch detect 'handover'? ++ Can update the table by having AP2 immediately send message to old AP with new association to update old AP's forwarding table.

802.11: advanced capabilities

-Rate Adaptation - Base station, mobile dynamically change transmission rate as mobile moves to improve SNR.

-SNR decreases, BER increases as node moves away from base station

-When BER becomes too high, switch to lower transmission rate with lower BER. Alternately, if many packets in a row succeed in not being dropped, increase SNR.

-Power management - Node tells AP it will **sleep** until next frame ++ AP doesn't transmit frames before node wakes up ++ 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 there are beacon frames to be sent. Else sleep again.

Personal area networks (Piconets): Bluetooth

-Small range - less than 10 m diameter

-Replacement for cables (mouse, keyboard, headphones)

-Ad hoc: no infrastructure

-2.4-2.5 GHz ISM radio band, up to 3 Mbps

-Master controller / Clients devices ++ master polls clients, grants requests for client transmissions

-TDM (Time Division Multiplexing) - 625 µsec sec. slot

-FDM - sender uses 79 frequency channels in known, pseudo-random order slot-to-slot (spread spectrum) (See FHSS below)

-FHSS = Frequency-Hopping Spread Spectrum - channel frequency changes in a known but pseudo-random manner from slot to slot. This form of channel hopping reduces interference to only a few slots.

-Parked mode: clients can “go to sleep” (park) and later wakeup (to preserve battery)

-Bootstrapping: nodes self-assemble (plug and play) into piconet - Self-organizing

- "Neighbor Discovery Problem" - Central Controller finds other devices in range by broadcasting series of 32 inquiry messages each on different frequency channels up to 128 times.

-"Bluetooth Paging" - Central Controller invites clients to piconet.

7.4) **Cellular Networks: 4G and 5G** (595)

KEY POINTS:

-THE solution for wide-area mobile internet. (APs are too short-ranged)

-Widespread deployment & use ++ More mobile-broadband connection devices than fixed ++ 4G available 90% of the time in the US (97% in Korea)

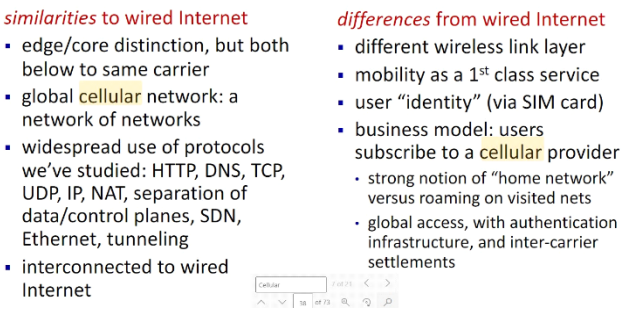
-Cell = Geographic Coverage Area in an LTE network

-Tranmission rates up to 100s Mbps.

-Technical standards - 3GPP = 3rd Generation Partnership Project

-4G - LTE = Longterm Evolution Standards

-Comparison/Similarities/Differences wired internet and 4G/5G



Elements of 4G Architecture

-Mobile devices (smartphones, tablet, laptop, IoT, 4G LTE Radio)

-UE = User Equipment

-IMSI = 64 International Mobile Subscriber Identity

-Base station - Edge of network ++ Manages wireless radio resources ++ Coordinates Device authentication ++ Similar to Wifi AP BUT (Active role in user mobility ++ coordinates with nearby base stations to optimize radio use)

-LTE Jargon - eNode-B = Base Station on LTE network

ELEMENTS OF ARCHITECTURE CONTINUED:

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

-P-GW = PDN Gateway - gGateway to mobile cellular network ++ Similar to other internet gateway routers from internet perspective ++ NAT services ++ (LAST ELEMENT ENCOUNTERED BEFORE INTERNET)

-S-GW = Serving Gateway in LTE mobile cellular network

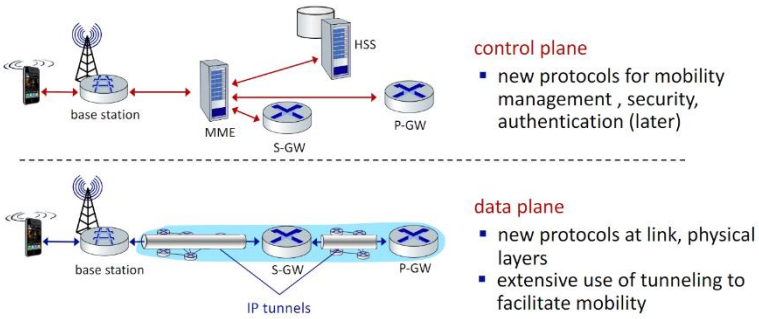
-MME = Device authentication in LTE (device-to-network, network-to-device) coordinated with mobile home network HSS ++ Mobile Device management (Device handover between cells ++ Tracking/paging device location) ++ Path (tunneling) setup from mobile device to P-GW

----Note, when device changes cells, only the tunnel terminating at the base station needs to be moved.

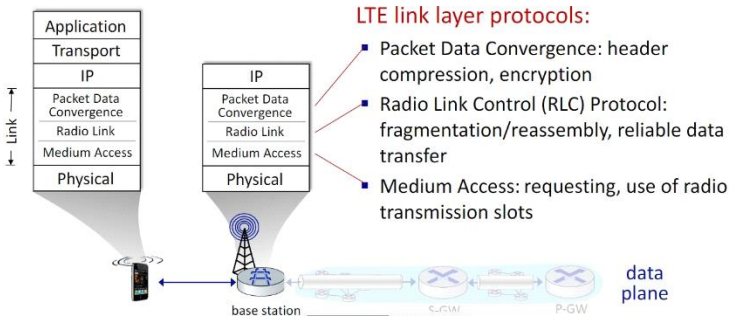
----MME tracks device while it is sleeping & when it moves cells ++ Locates device by **paging**.

LTE Element	Description	Similar WLAN function(s)
Mobile device (UE: User equipment)	End user's IP-capable wireless/mobile device (e.g., smartphone, tablet, laptop)	Host, end-system
Base Station (eNode-B)	Network side of wireless access link into LTE network	Access point (AP), although the LTE base station performs many functions not found in WLANs
The Mobility Management Entity (MME)	Coordinator for mobile device services: authentication, mobility management	Access point (AP), although the MME performs many functions not found in WLANs
Home Subscriber Server (HSS)	Located in a mobile device's *home* network, providing authentication, access privileges in home and visited networks	No WLAN equivalent
Serving Gateway (S-GW), PDN-Gateway (P-GW)	Routers in a cellular carrier's network, coordinating forwarding to outside of the carrier's network	iBGP and eBGP routers in access ISP network
Radio Access Network	Wireless link between mobile device and a base station	802.11 wireless link between mobile and AP

LTE: data plane control plane separation



LTE data plane protocol stack: first hop (new protocols mostly in link, physical layers & mobility management) (Key terms: packet data convergence, Radio link control (RLC), encryption)



-LTE Radio Access Network - Downstream channel (FDM, TDM within frequency channel ++ OFDM - orthogonal frequency division multiplexing)

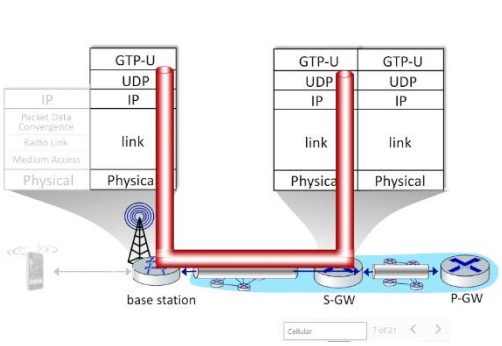
-Orthogonal = minimal interference from other channels ++ upstream (FDM, TDM similar to OFDM)

-OFDM - each active mobile device allocated two or more 0.5 ms time slots over 12 frequencies

----Scheduling algorithm not standardized - up to operator

----100's Mbps per device possible

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

-BS broadcasts primary synch signal every 5 ms on all frequencies

----BSs from multiple carriers may be broadcasting synch signals

-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

-Mobile selects which BS to associate with (e.g., preference for home carrier)

-More steps still needed to authenticate, establish state, set up data plane

NETWORK ATTACHEMENT:

-Attachment to a Base Station

-Mutual Authentication (Network checks if attaching device has IMSI identification)

-Mobile-device-to-PDN-gateway Data Path Configuration

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

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

-All IP: Carriers interact with each other & public internet at exchange points

FUTURE: 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 antennae

-**Millimeter wave frequencies** - Much higher data rates, but over shorter distances

----Pico-cells: cells diameters: 10-100 m

----Massive, dense deployment of new base stations required

-**3 standards**!

----eMBB (Enhanced Mobile Broadband) - Provides for increased bandwidth for higher download and upload speeds, as well as a moderate reduction in latency when compared to 4G LTE. eMBB enables rich media applications, such as mobile augmented reality and virtual reality, as well as mobile 4K resolution and 360° video streaming.

----URLLC (Ultra Reliable Low-Latency Communications) - URLLC is targeted towards applications that are highly latency-sensitive, such as factory automation and autonomous driving. URLLC is targeting latencies of 1msec. As of this writing, technologies that enable URLLC are still being standardized.

----mMTC (Massive Machine Type Communications). mMTC is a narrowband access type for sensing, metering, and monitoring applications. One priority for the design of 5G networks is to lower barriers for network connectivity for IoT devices. In addition to lowering latency, emerging technologies for 5G networks are focusing on reducing power requirements, making the use of IoT devices more pervasive than has been with 4G LTE.

CAPACITY FORMULA - capacity = cell density * available spectrum * spectral efficiency

7.5) **Mobility Management: Principles** (610)

KEY SLIDE POINTS:

What is mobility (Spectrum: Least mobility top to greatest mobility bottom)

----1) Device moves between networks, but powers down while moving

----2) Device moves within same AP in one provider network

----3) **Device moves among APs in one provider network**

----4) **Device moves among multiple provider networks, while maintaining ongoing connections**

-FOCUS ON LAST TWO ^

Mobility approaches:

-Routers have many advantages but do not scale well to billions of mobile devices.

-The correct approach is to let end-systems handle it - Functionality at the "edge"

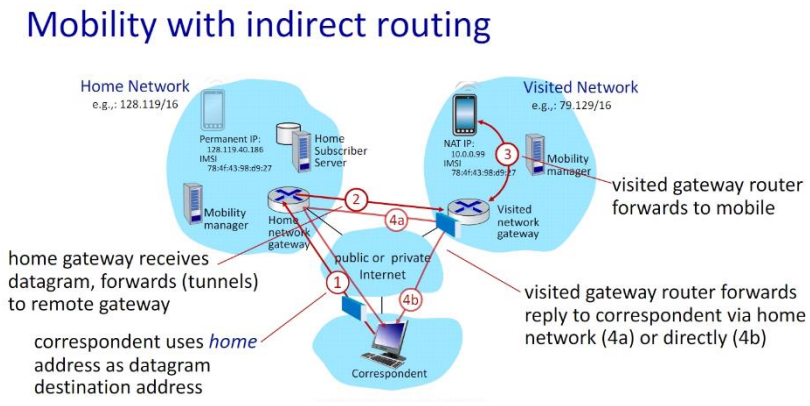
----Indirect routing: communication from correspondent to mobile goes through home network, then forwarded to remote mobile.

----Direct routing: correspondent gets foreign address of mobile from home, then sends directly to mobile.

Visited Networks

-Home Network HSS stores identity and info ++ Knows location of mobile on visited Network

-Visited Network = Mobile associates with visited mobility manage ++ But visited mobility manager registers mobile's location with home HSS



-Visited network can either forward data to the home network, which forwards it to the client OR it can forward data to the client directly.

7.6) **Mobility Management in Practice** (619)

KEY SLIDE POINTS:

7.6.1 Mobility Management in 4G/5G Networks

-Four Major steps in mobile user streaming video & switching cells.

----1) Base Station Association - Covered earlier ++ Mobile identifies itself with IMSI

----2) Control Plane Configuration - MME, home HSS establish control-plane state - indicates the mobile is resident in visited network ++ MME handles the authentication (likely but not necessarily retrieving this info from HSS) and informs the HSS that the mobile is in visited network. ++ Alternately **local breakout** uses visited P-GW, but rarely used)

----3) Data-plane configuration:

--------MME configures forwarding tunnels for mobile

--------Visited & home network establish tunnels from home P-GW to mobile (S-GW in resident/visited network but P-GW in home network)

----4) Mobile device changes its point of attachment to visited network

-Configuration:

-Mobile communicates with local MME via BS control-plane channel

-MME uses mobile's IMSI info to contact mobile's home HSS:

----Retrieve authentication, encryption, network service informati

----Home HHS knows mobile now resident in visited network

-BS, mobile select parameters for BS-mobile data-plane radio channel

-S-GW to BS tunnel: when mobile changes base stations, simply change endpoint IP address of tunnel

-S-GW to home P-GW tunnel: implementation of indirect routing

-Tunneling via GTP (GPRS tunneling protocol): mobile's datagram to streaming server encapsulated using GTP inside UDP, inside datagram

-Handover Process

----1) Current (source) BS selects target BS, sends Handover Request messageto target BS

----2) Target BS pre-allocates radio time slots, responds with HR ACK with info for mobile

----3) Source BS informs mobile of new BS (mobile can now send via new BS -handover lookscomplete to mobile)

----4) Source BS stops sending datagrams to mobile, instead forwards to new BS (who forwards to mobile over radio channel)

----5) Target BS informs MME that it is new BS for mobile

--------MME instructs S-GW to change tunnel endpoint to be (new) target BS

----6) Target BS ACKs back to source BS: handover complete, source BS can release resources

----7) Mobile's datagrams now flow through new tunnel from target BS to S-GW

7.6.2 Mobile IP

-Hypothetical system that works but is not widely adopted.

-Standardized before mobile/4G support for Internet protocols

-<u>Mobile IP architecture:</u>

----Indirect routing to node (via home network) using tunnels

----Mobile IP home agent: combined roles of 4G HSS and home P-GW

----Mobile IP foreign agent: combined roles of 4G MME and S-GW

----Protocols for agent discovery in visited network, registration of visited location in home network via ICMP extensions

Mobile IP Architecture

4G/5G element	Mobile IP element	Discussion
Home network	Home network	
Visited network	Foreign network	
IMSI identifier	Permanent IP address	Globally unique routable address information
Home Subscriber Service (HSS)	Home agent	
Mobility Management Entity (MME)	Foreign agent	
Data plane: indirect forwarding via the home network, with tunneling between the home and visited network, and tunneling within the network in which the mobile device resides	Data plane: indirect forwarding via the home network, with tunneling between the home and visited network	
Base station (eNode-B)	Access Point (AP)	No specific AP technology is specified in Mobile IP
Radio Access Network	WLAN	No specific WLAN technology is specified in Mobile IP

-3 main pieces

Agent discovery. Mobile IP defines the protocols used by a foreign agent to advertise its mobility services to a mobile device that wishes to attach to its network. Those services will include providing a care-of-address to the mobile device for use in the foreign network, registration of the mobile device with the home agent in the mobile device's home network, and forwarding of datagrams to/from the mobile device, among other services.

Registration with the home agent. Mobile IP defines the protocols used by the mobile device and/or foreign agent to register and deregister a care-of-address with a mobile device's home agent.

Indirect routing of datagrams - Mobile IP also defines the manner in which datagrams are forwarded to mobile devices by a home agent, including rules for forwarding datagrams and handling error conditions, and several forms of tunneling

7.7) **Wireless and Mobility: Impact on Higher-Layer Protocols** (626)

KEY SLIDE POINTS:

-Logically, impact should be minimal ++ (Best effort service model unchanged ++ TCP/UDP can and do run over wireless & mobile)

-HOWEVER Performance Suffers:

----Packet loss/delay due to handover loss and bit errors (Discarded packets ++ Delays for link layer transmissions)

----TCP interprets loss as congestion, will decrease congestion window un-necessarily. (Assumes corruption and handover loss is congestion)

---------3 Fixes - Local recovery(Recovers from bit erros at point of origin) ++ TCP sender awareness of wireless links (TCP given info that the link in use is wireless & adjusted accordingly) ++ Split connection (Split connection into two segments: Mobile --> Access point AND Access Point --> Other end point)

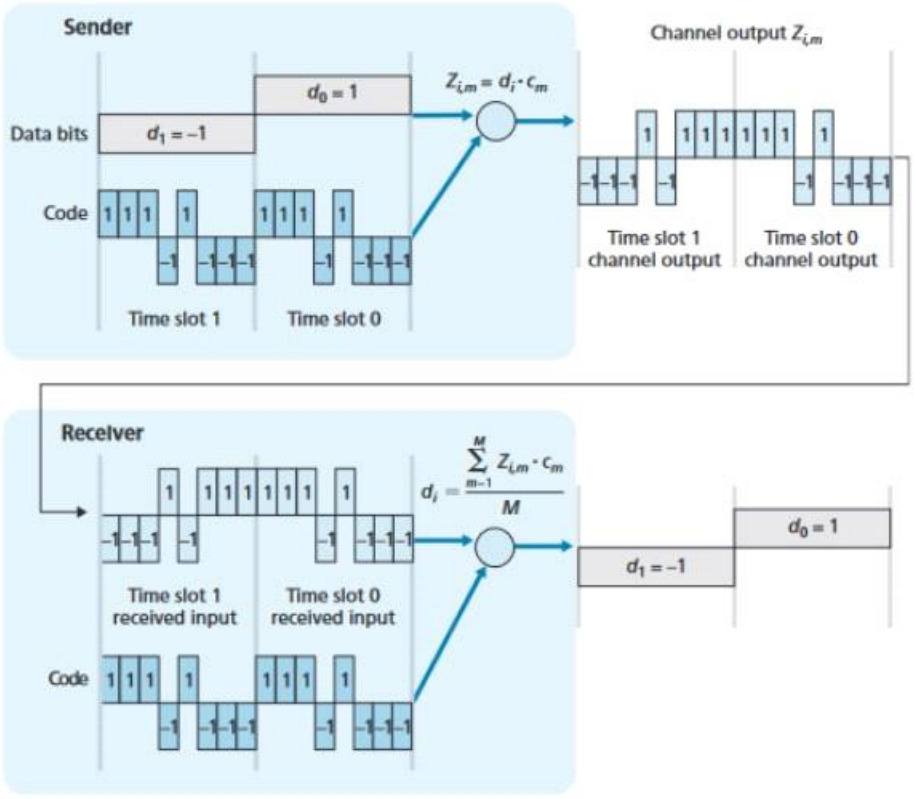
----Delay impairments for real-time traffic.

----Bandwidth a scare resource for wireless links.

Ch 7) **FORMULAS**:

0.) SNR = 20*log10(amplitude_signal)/(amplitude_noise)

1) **CDMA**



2) **Capacity Formula** (LTE & 5G)

capacity = cell density * available spectrum * spectral efficiency

Ch 7) **GLOSSARY**:

3GPP = 3rd Generation Partnership Project

AP = Access Point = Base station

ARQ = Link layer acknowledgement/retransmission scheme used by 802.11 MAC Protocol. Uses DIFS and SIFS(?)

BER = Bit Error Rate (Wireless Transmission Error) (7.2 and 7.3)

BSS = Basic Service Set (BSS) - Fundamental building block of 802.11 architecture

CDMA = Code Division Multiple Access

Cell = Geographic Coverage Area in an LTE network

CSMA/**CA** = CSMA with Collision Avoidance

DIFS = Distributed Inter-frame Space = In CSMA/CA - If initially the station senses the channel idle, it transmits its frame after a short period of time known as the DIFS.

eNode-B = Base Station on LTE network

HSS = Home Subscriber Service - LTE architechture stores info about mobile devices for which the HSS's network is their "home network"

IMSI = 64 International Mobile Subscriber Identity in LTE network (7.4)

LTE = Longterm Evolution Standards

MIMO: multiple directional antennae in 5G network

MME = Device authentication in LTE (device-to-network, network-to-device) coordinated with mobile home network HSS

OFDM - Orthogonal frequency division multiplexing used in LTE network.

P-GW = PDN Gateway - Gateway to LTE mobile cellular network ++ Similar to other internet gateway routers ++ NAT services

Piconet = A Bluetooth ad hoc network - A "personal net" with 10s of meters of range or less.

RTS = "Request to Send" Packet - Used in Collision Avoidance (7.3)

S-GW = Serving Gateway in LTE mobile cellular network

SIFS = the Short Inter-frame Spacing = When destination receives a frame that passes the CRC, it waits a short periods of time known as SIFS and then sends back an ACK frame. (7.3)

SNR = Signal to Noise ratio (7.2 and 7.3)

UE = User Equipment in LTE (4G/5G)

WiFi jungle = Any physical location where a wireless station receives a sufficiently strong signal from two or more APs

==========

==========

CHAPTER 8 – Security in Computer Networks

==========

==========

NETWORKS FINAL CHAPTER 8 NOTES

8.0) **GOALS**:

UNDERSTAND:

-Cryptography (Many uses beyond confidentiality) ++ Authentication ++ Message Integrity.

-Security in Practice (Firewalls & Intrusion Detection Systems ++ Security in Application/Transport/Network/Link layers)

8.1) **What Is Network Security?** (640)

KEY SLIDE POINTS:

Properties of Secure Communication.

-1) Only sender, intended receiver should "understand" message contents

----Sender encrypts message

----Receiver decrypts message

-2) Authentication: sender, receiver want to confirm identity of each other

-3) Message integrity: sender, receiver want to ensure message not altered (in transit, or afterwards) without detection

-4) Access and availability: services must be accessible and available to users

-5) Operational security - firewalls and intrusion detection systems are used to counter attacks against an organization's network. A firewall sits between the organization's network and the public network, controlling packet access to and from the network. An intrusion detection system performs "deep packet inspection," alerting the network administrators about suspicious activity.

"Bobs & Alices" = Entities that need secure communication.

----Real-lifeBobs and Alices!

----Web browser/server for electronic transactions (e.g., on-line purchases)

----On-line banking client/server

----DNS servers

----BGP routers exchanging routing table updates

----Other examples?

Network Attack Types

-Eavesdrop: intercept messages

-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)

8.2) **Principles of Cryptography** (642)

KEY SLIDE POINTS:

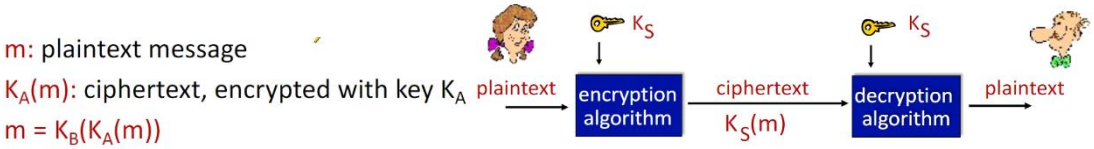
Two Types of Keys:

Public key systems = A pair of keys is used. One of the keys is known to both Bob and Alice (indeed, it is known to the whole world). The other key is known only by either Bob or Alice (but not both)

Symmetric key systems = Alice's and Bob's keys are identical and are secret

8.2.1 Symmetric Key Cryptography

General Terminology --> Symmetric Key Image



-<u>Breaking an encryption scheme</u>

----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)

CIPHERS

-<u>Substitution cipher</u>

----Monoalphabetical (Substitute one letter for another)

--------Encryption Key (Mapping from set of 26 letters to different set of 26 letters)

----n-substitution cipers (Polyalphabetical)

----Cycling pattern

----Encryption Key - n substitution ciphers, and cyclic pattern (Key need not be just n-bit pattern)

Block Ciphers

input	output	input	output
000	110	100	011
001	111	101	010
010	101	110	000
011	100	111	001

Table 8.1 ♦ A specific 3-bit block cipher

-NOTE k = 3 --> Therefore 3-bit block cipher --> permutes in 2^k! ways.

-Cipher-Block Chaining

----PROBLEM: If we apply a block cipher as described by simply chopping up the message into k-bit blocks and independently encrypting each block, a subtle but important problem occurs.

----We might have identical blocks, allowing decryption via pattern analysis.

----Therefore, add randomness.

----The basic idea is as follows. The sender creates a random k-bit number r(i) for the ith block and calculates c(i) = KS(m(i) $\oplus$ r(i)). Note that a new k-bit random number is chosen for each block. The sender then sends c(1), r(1), c(2), r(2), c(3), r(3), and so on. Since the receiver receives c(i) and r(i), it can recover each block of the plaintext by computing **m(i) = KS(c(i)) $\oplus$ r(i)**

----NOTE: If two plaintext blocks m(i) and m(j) are the same, the corresponding ciphertext blocks c(i) and c(j) will be different (as long as the random numbers r(i) and r(j) are different, which occurs with very high probability).

----To reduce number of bits sent: Create **Initialization Vector (IV)**

---- Initialization Vector (IV) - Random k-bit string ++

---------For the first block, the sender calculates m(1) $\oplus$ c(0), that is, calculates the exclusive-or of the first block of cleartext with the IV. It then runs the result through the block-cipher algorithm to get the corresponding ciphertext block; that is, **c(1) = KS(m(1) $\oplus$ c(0)).** The sender sends the encrypted block c(1) to the receiver.

--------For the ith block, the sender generates the ith ciphertext block from **c(i) = KS(m(i) $\oplus$ c(i - 1)).**

--------When the receiver receives c(i), it decrypts it with KS to obtain s(i) = m(i) $\oplus$ c(i - 1); since the receiver also knows c(i - 1), it then obtains the cleartext block from m(i) = s(i) $\oplus$ c(i - 1).

-Symmetric key crypto: DES (Data Encryption Shared)

-US encryption standard

-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: encrypt 3 times with 3 different keys

-AES: Advanced Encryption Standard

-Symmetric-key NIST standard, replaced DES

-Processes data in 128-bit blocks

-NOTE: 128, 192, or 256 bit keys

-Brute force decryption (try each key) taking 1 sec on DES, takes 149 trillion years for AES

8.2.2 Public Key Encryption

Problem: Symmetric key crypto requires sender/receiver to agree on code

-Sender/Receiver might not have met

-PUBLIC KEY CRYPTOGRAPHY

-Sender/Reicever do NOT share same key

-Receiver's public encryption key known to all ++ Receiver's private decryption key known only to self.

-Requirements

requirements:

1. need $K_B^+(\cdot)$ and $K_B^-(\cdot)$ such that
   $K_B^-(K_B^+(m)) = m$
2. given public key K_B^+, it should be impossible to compute private key K_B^-

RSA

-Modular arithmetic:

----x mod n = remainder of x when divide by n

----facts:

--------[(a mod n) + (b mod n)] mod n = (a+b) mod n
--------[(a mod n) -(b mod n)] mod n = (a-b) mod n
--------[(a mod n) * (b mod n)] mod n = (a*b) mod n

----thus:
--------(a mod n)dmod n = **a^d mod(n)**

example: x=14, n=10, d=2:

$(x \bmod n)^d \bmod n = 4^2 \bmod 10 = 6$

$x^d = 14^2 = 196 \quad x^d \bmod 10 = 6$

-RSA: getting ready

----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: Creating public/private key pair

----1) Choose two large prime numbers p, q.(e.g., 1024 bits each)

----2) Compute n = pq, z = (p-1)(q-1)

----3) Choose e (with e < n) that has no common factors with z (e & z are “relatively prime”).

----4) Choose d such that ed - 1 is exactly divisible by z. (in other words: e*d mod(z) = 1).

----5) Public key is (n,e). Private key is (n,d).

-RSA: encryption, decryption

-Given (n,e) and (n,d) as computed above

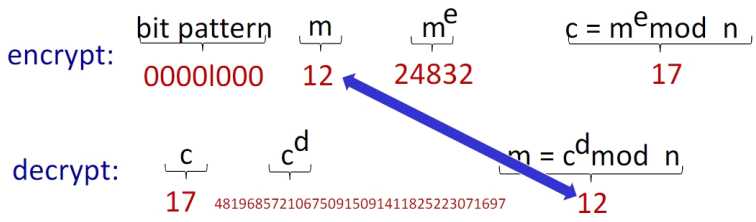
0. given (*n,e*) and (*n,d*) as computed above
1. to encrypt message *m (<n)*, compute
   $c = m^e \bmod n$
2. to decrypt received bit pattern, *c*, compute
   $m = c^d \bmod n$

magic happens! $m = (\underbrace{m^e \bmod n}_{c})^d \bmod n$

RSA example:

Bob chooses *p=5, q=7*. Then *n=35, z=24*.
e=5 (so *e, z* relatively prime).
d=29 (so *ed-1* exactly divisible by z).

encrypting 8-bit messages.



RSA Proof - Why Does RSA Work?

Why does RSA work?

- must show that $c^d \bmod n = m$, where $c = m^e \bmod n$
- fact: for any x and y: $x^y \bmod n = x^{(y \bmod z)} \bmod n$
  - where n= pq and z = (p-1)(q-1)
- thus,

$$
\begin{aligned}
c^d \bmod n &= (m^e \bmod n)^d \bmod n \\
&= m^{ed} \bmod n \\
&= m^{(ed \bmod z)} \bmod n \\
&= m^1 \bmod n \\
&= m
\end{aligned}
$$

RSA IMPORTANT PROPERTY

The following property will be *very* useful later:

$$\underbrace{K_B^-(K_B^+(m))}_{\text{use public key first, followed by private key}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{use private key first, followed by public key}}$$

result is the same!

-Property follows directly from modular arithmetic:

-(m^e mod(n))^d mod(n) = m^(ed) mod(n)

= m^(de) mod(n)

= (m^d mod(n))^e mod(n)

Why is RSA secure?

-Given Bob's key, intruder has to find factors of n without knowing the two factors p and q

----Fact: factoring a big number is hard

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, KS

----Bob and Alice use RSA to exchange a symmetric session key KS

----Once both have KS, they use symmetric key cryptography

8.3) **Message Integrity and Digital Signatures** (656)

KEY SLIDE POINTS:

Cryptographic technique analogous to hand-written signatures:

-Sender (Bob) digitally signs document: he is document owner/creator.

-Verifiable, nonforgeable: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 KB, creating "signed" message, KB-(m)

Digital signatures

- 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'

non-repudiation:
- ✓ Alice can take m, and signature $K_B^-(m)$ to court and prove that Bob signed m

8.3.1 Cryptographic Hash Functions

-PROBLEM: 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 msuch that x = H(m)

-NOTE: Internet Checksum = Poor hash function

----Internet checksum has some properties of hash function: (Produces fixed length digest (16-bit sum) of message ++ Is many-to-one)

----Too easy to end up with duplicate hash values.

----IT SHOULD BE COMPUTATIONALLY INFEASIBLE TO FIND ANY TWO DIFFERENT MESSAGES x AND y SUCH THAT H(x) = H(y) (In other words an attacker shouldn't be able to substitute one message for another)

8.3.2 Message Authentication Code

-(ch 8) MAC = Message Authentication Code is message m, where s is concatenated with m to create m + s, and the hash H(m + s) is calculated (for example, with SHA-1). H(m + s) is called the **MAC**.

-Alice then appends the MAC to the message m, creating an extended message (m, H(m + s)), and sends the extended message to Bob.

-Bob receives an extended message (m, h) and knowing s, calculates the MAC H(m + s). If H(m + s) = h, Bob concludes that everything is fine.

-MOST POPULAR MAC IS HMAC - HMAC actually runs data and the authentication key through the hash function twice

-PROBLEM: How to share MAC authentication key?

----Require digital signature.

8.3.3 Digital Signatures

<u>Public Key Certification</u>

-Why Use Certified public keys - Needed to ensure Bob's identity while using a public key - Trudy cannot take the key to pose as Bob to Alice and vice versa.

-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

---

8.4) **End-Point Authentication** (666)

KEY POINTS:

GOAL: Bob wants "Alice" to prove she really is Alice.

-FAILURE: It is not enough to state identity - can be faked.

-FAILURE: It is not enough to include IP address & identity - IP address can be spoofed.

-FAILURE: Password + IP address + Identity - fails because Trudy can record and playback interaction to Bob.

-FAILURE: Encrypting the password doesn't prevent a playback interaction.

-**BETTER**: Use a 'nonce' - A number (R) only used once-in-a-lifetime (Alice sends message to Bob. Bob chooses a nonce and sends it to Alice. ++ Alice encrypts the nonce using Symmetric Key A&B ++ Bob decrypts the message - If it equals the nonce, Alice is authenticated.

-FLAW - Public Key - Trudy can act as a middleman, posing as Alice to Bob and vice versa. She can recover the message m like this without being detected by Bob or Alice.

----SOLVE WITH DIGITAL SIGNATURES

8.5) **Securing E-Mail** (671)

KEY SLIDE POINTS:

8.5.1 Secure E-Mail

-When security is implemented at one layer, all layers above it have the same security (Note that lower level security can't provide user-level security)

GOAL: Confidentiality - Alice wants to send confidentiale-mail, m, to Bob.

-Alice:

----Generates random symmetricprivate key, KS

----Encrypts message with KS (for efficiency)

----Also encrypts KS with Bob's public key

----Sends both KS(m) and KB+(KS) to Bob

-Bob:

----Uses his private key to decrypt and recover KS

----Uses KSto decrypt KS(m) to recover m

GOAL: Message Integrity, Authentication:

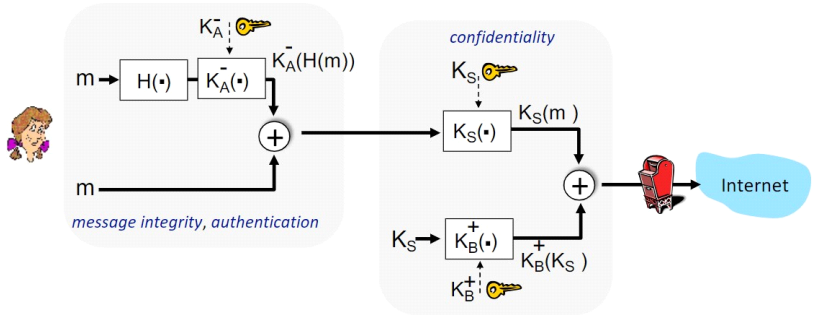
-Alice:

----Alice digitally signs hash of her message with her private key, providing integrity and authentication

----Sends both message (in the clear) and digital signature

GOAL: All 3 now! - Confidentiality, Message Integrity, Authentication

Alice sends m to Bob, with *confidentiality, message integrity, authentication*



Alice uses three keys: her private key, Bob's public key, new symmetric key

What are Bob's complementary actions?

8.5.2 PGP

PGP software uses MD5 or SHA for calculating the message digest; CAST, triple-DES, or IDEA for symmetric key encryption; and RSA for the public key encryption. When PGP is installed, the software creates a public key pair for the user. The public key can be posted on the user's Web site or placed in a public key server. The private key is protected by the use of a password. The password has to be entered every time the user accesses the private key. PGP gives the user the option of digitally signing the message, encrypting the message, or both digitally signing and encrypting.

8.6) **Securing TCP Connections: TLS** (676)

KEY SLIDE POINTS:

-**TLS** - (USED IN TCP) Transport-layer security (Widely deployed security protocol above the transport layer ++ Supported by almost all browsers, web servers: https (port 443))

-Provides:

----Confidentiality: via symmetric encryption

----Integrity: via cryptographic hashing

----Authentication: via public key cryptography.

-History:

----Early research, implementation: secure network programming, secure sockets

----Secure socket layer (SSL) deprecated [2015]

----TLS 1.3: RFC 8846 [2018]

8.6.1 The Big Picture

-Pieces: Handshake ++ Key Derivation ++ Data Transfer ++ TLS Record

----Handshake: Alice, Bob use their certificates, private keys to authenticate each other, exchange or create shared secret
----Key Derivation: Alice, Bob use shared secret to derive set of keys

----Data transfer: Stream data transfer: data as a series of records (not just one-time transactions)
----Connection closure: special messages to securely close connection

Handshake

-t-tls: initial handshake:

----Bob establishes TCP connection with Alice
----Bob verifies that Alice is really Alice
----Bob sends Alice a master secret key (MS), used to generate all other keys for TLS session
----potential issues:
-------3 RTT before client can start receiving data (including TCP handshake)

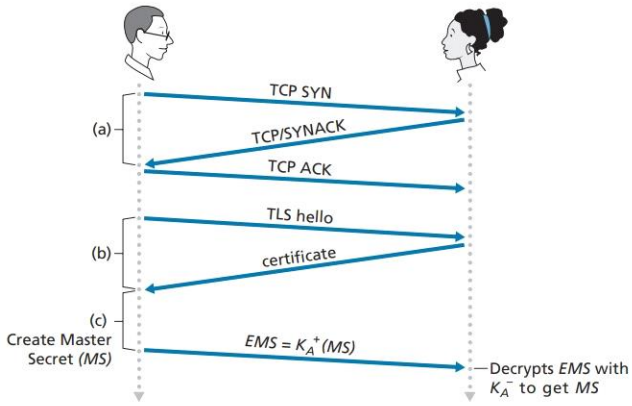


Figure 8.25 ♦ The almost-TLS handshake, beginning with a TCP connection

Key Derivation

-t-tls: cryptographic keys:

----Considered bad to use same key for more than one cryptographic function (Different keys for message authentication code (MAC) and encryption)

----Four keys:

--------Kc: encryption key for data sent from client to server

--------Mc: MAC key for data sent from client to server

--------Ks: encryption key for data sent from server to client

--------Ms: MAC key for data sent from server to client

----Keys derived from key derivation function (KDF):

--------Takes master secret and (possibly) some additional random data to create new keys

Data Transfer

-t-tls: encrypting data:

----Recall: TCP provides data bytestream abstraction

----Q: Can we encrypt data in-stream as written into TCP socket?

--------A: Where would MAC go? If at end, no message integrity until all data received and connection closed!

--------Solution: break stream in series of “records”:

----------Each client-to-server record carries a MAC, created using Mc

----------Receiver can act on each record as it arrives

----t-tls record encrypted using symmetric key, Kc, passed to TCP:

-To prevent middleman Trudy from inserting/deleting/replacing TCP segments, TLS using sequence numbers.

TLS Record

-t-tls: encrypting data (more):

----Possible attacks on data stream?

--------DANGER: Re-ordering: man-in middle intercepts TCP segments and reorders (manipulating sequence #s in unencrypted TCP header)

--------Replay

----Solutions:

--------Use TLS sequence numbers (data, TLS-seq-# incorporated into MAC) (Can't switch packets around)

--------Use nonce (Can't intercept and fake identity)

Figure 8.26 ♦ Record format for TLS

8.6.2 A More Complete Picture

TLS 1.3 Cipher Suite

-"Cipher suite": algorithms that can be used for key generation, encryption, MAC, digital signature

-TLS: 1.3 (2018):more limited cipher suite choice than TLS 1.2 (2008)

----Only 5 choices, rather than 37 choices

----Requires Diffie-Hellman (DH) for key exchange, rather than DH or RSA

----Combined encryption and authentication algorithm ("authenticated encryption") for data rather than serial encryption, authentication
--------(4 based on AES)

----HMAC uses SHA (256 or 284) cryptographic hash function

TLS 1.3 Handshake

-1) The client sends a list of cryptographic algorithms it supports, along with a client nonce.

-2) From the list, the server chooses a symmetric algorithm (for example, AES) and a public key algorithm (for example, RSA with a specific key length), and HMAC algorithm (MD5 or SHA-1) along with the HMAC keys. It sends back to the client its choices, as well as a certificate and a server nonce.

-3) The client verifies the certificate, extracts the server's public key, generates a Pre-Master Secret (PMS), encrypts the PMS with the server's public key, and sends the encrypted PMS to the server.

-4) Using the same key derivation function (as specified by the TLS standard), the client and server independently compute the Master Secret (MS) from the PMS and nonces. The MS is then sliced up to generate the two encryption and two HMAC keys. Furthermore, when the chosen symmetric cipher employs CBC (such as 3DES or AES), then two Initialization Vectors (IVs)—one for each side of the connection—are also obtained from the MS. Henceforth, all messages sent between client and server are encrypted and authenticated (with the HMAC).

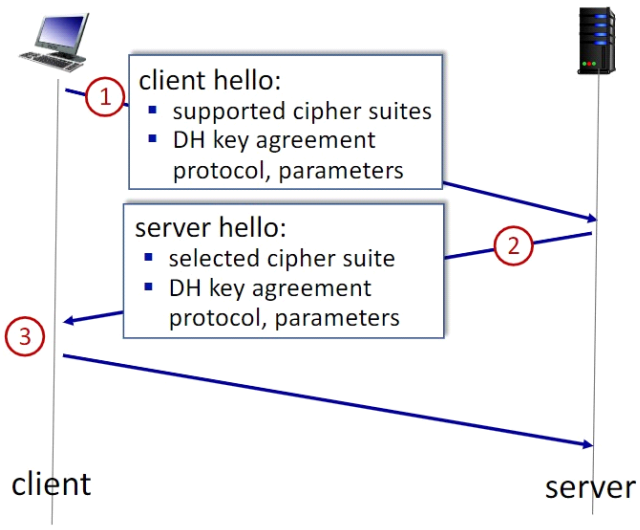
-5) The client sends the HMAC of all the handshake messages.

-6) The server sends the HMAC of all the handshake messages.

----NOTE: Last two steps protect the handshake from tamperating (I.e. Trudy might try to delete all the strong algorithms from the client's offered list. Tampering will be discovered on HMAC comparison.

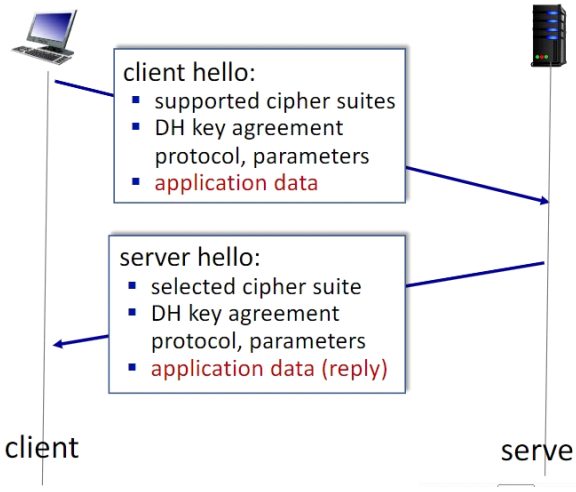
----NOTE: The sequence numbers stop segment replay attack ++ The nonces stop connection replay attack.

TLS 1.3 handshake: 1 RTT



1. client TLS hello msg:
   - *guesses* key agreement protocol, parameters
   - indicates cipher suites it supports
2. server TLS hello msg chooses
   - key agreement protocol, parameters
   - cipher suite
   - server-signed certificate
3. client:
   - checks server certificate
   - generates key
   - can now make application request (e.g.., HTTPS GET)

TLS 1.3 handshake: 0 RTT



- initial hello message contains encrypted application data!
  - "resuming" earlier connection between client and server
  - application data encrypted using "resumption master secret" from earlier connection
- vulnerable to replay attacks!
  - maybe OK for get HTTP GET or client requests not modifying server state

Connection Closure

-t-tls: connection close:

----Truncation attack:

--------Attacker forges TCP connection close segment

--------One or both sides thinks there is less data than there actually is

----Solution: record types, with one type for closure

--------Type 0 for data; type 1 for close (If close not declared and information stops, something went wrong)

----MAC now computed using data, type, sequence #num

8.7) **Network-Layer Security: IPsec and Virtual Private Networks** (683)

KEY SLIDE POINTS:

8.7.1 IPsec and Virtual Private Networks (VPNs)

-IPSec - Provides datagram-level encryption, authentication, integrity

----For both user traffic and control traffic (e.g., BGP, DNS messages)

-Two “modes”:

----Transport mode: Only datagram payload is encrypted & authenticated.

----Tunnel mode:

--------Entire datagram is encrypted, authenticated

--------Encrypted datagram encapsulated in new datagram with new IP header, tunneled to destination

8.7.2 The AH and ESP Protocols

Two IPsec protocols

-Authentication Header (AH) protocol [RFC 4302]:

----Provides source authentication & data integrity but not confidentiality

-Encapsulation Security Protocol (ESP)

----Provides source authentication, data integrity, and confidentiality

----More widely used than AH

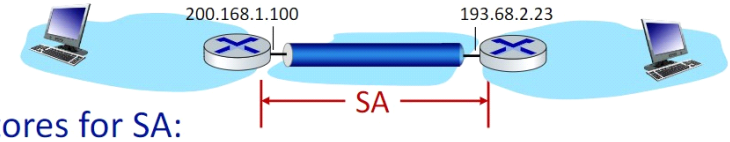
8.7.3 Security Associations

-Before sending data, security association (SA) established from sending to receiving entity (An **SA is one directional!** You need TWO SAs to achieve bidirectional communication.)

-Ending & receiving entitles maintain state informationabout SA

----Recall: TCP endpoints also maintain state info

----IP is connectionless; IPsec is connection-oriented!



R1 stores for SA:

- 32-bit identifier: *Security Parameter Index (SPI)*
- origin SA interface (200.168.1.100)
- destination SA interface (193.68.2.23)
- type of encryption used
- encryption key
- type of integrity check used
- authentication key

Sending Router will **maintain state information** about this SA, which will include:

• A 32-bit identifier for the SA, called the Security Parameter Index (SPI)

• The origin interface of the SA (in this case 200.168.1.100) and the destination

interface of the SA (in this case 193.68.2.23)

• The type of encryption to be used (for example, 3DES with CBC)

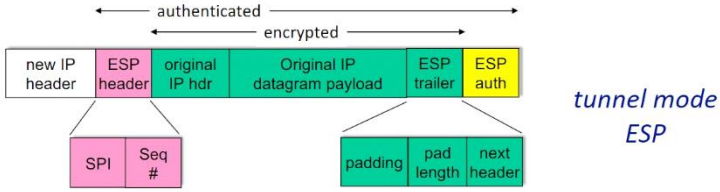
• The encryption key

• The type of integrity check (for example, HMAC with MD5)

• The authentication key

8.7.4 The IPsec Datagram

IPsec datagram



- ESP trailer: padding for block ciphers
- ESP header:
  - SPI, so receiving entity knows what to do
  - sequence number, to thwart replay attacks
- MAC in ESP auth field created with shared secret key

-ESP trailer: padding for block ciphers
-ESP header:
----SPI, so receiving entity knows what to do
----Sequence number, to thwart replay attacks
-MAC in ESP auth field created with shared secret key

ESP tunnel mode: actions

-at R1: (R1 ----> R2)

----Appends ESP trailer to original datagram (which includes original header fields!)

----Encrypts result using algorithm & key specified by SA

----Appends ESP header to front of this encrypted quantity

----Creates authentication MAC using algorithm and key specified in SA

----Appends MAC forming payload

----Creates new IP header, new IP header fields, addresses to tunnel endpoint

IPsec sequence numbers

-For new Security Assocation (SA), sender initializes seq. # to 0

-Each time datagram is sent on SA:
----Sender increments seq # counter
----Places value in seq # field

-Goal:
----Prevent attacker from sniffing and replaying a packet
----Receipt of duplicate, authenticated IP packets may disrupt service

-Method:
----Destination checks for duplicates
----Doesn't keep track of all received packets; instead uses a window

IPsec security databases

-Security Policy Database (SPD) - SPD: "what" to do

----Policy: for given datagram, sender needs to know if it should use IP sec
----Policy stored in security policy database (SPD)
----Needs to know which SA to use:
-------May use: source and destination IP address; protocol number

-Security Assoc. Database (SAD) - SAD: "how" to do it.

----Endpoint holds SA state in security association database (SAD)
----When sending IPsec datagram, R1 accesses SAD to determine how to process datagram
----When IPsec datagram arrives to R2, R2 examines SPI in IPsec datagram, indexes SAD with SPI, processing
----Datagram accordingly.

Summary of IPsec Services

Trudy sits somewhere between R1, R2. She doesn't know the authentication and encryption keys

--TRUDY CANNOT:
---- see original contents of datagram, source, dest IP address, transport protocol, application port.
----Flip bits without detection
----Masquerade as R1 using R1's IP address
----Replay a datagram

8.7.5 IKE: Key Management in IPsec

-PROBLEM: Manual keying is impractical for VPN with 100s of endpoints

-INSTEAD: Use IPsec IKE (Internet Key Exchange)

-IKE: PSK and PKI:
-Authentication (prove who you are) with either:
----Pre-shared secret (PSK) or
----With PKI (pubic/private keys and certificates).

-PSK: both sides start with secret:
----Run IKE to authenticate each other and to generate IPsec SAs (one in each direction), including encryption, authentication keys

-PKI: both sides start with public/private key pair, certificate
----Run IKE to authenticate each other, obtain IPsec SAs (one in each direction).
----Similar with handshake in SSL.

-IKE phases

--IKE has two phases:

----Phase 1: establish bi-directional IKE SA
--------Note: IKE SA different from IPsec SA
--------AKA ISAKMP security association
----Phase 2: ISAKMP is used to securely negotiate IPsec pair of SAs

-Phase 1 has two modes: aggressive mode and main mode
----Aggressive mode uses fewer messages
----Main mode provides identity protection and is more flexible

IPsec summary

-IKE message exchange for algorithms, secret keys, SPI numbers

-Either AH or ESP protocol (or both)
----AH provides integrity, source authentication
----ESP protocol (with AH) additionally provides encryption

-IPsec peers can be two end systems, two routers/firewalls, or a router/firewall and an end system

---

8.8) **Securing Wireless LANs and 4G/5G Cellular Networks** (691)

KEY SLIDE POINTS:

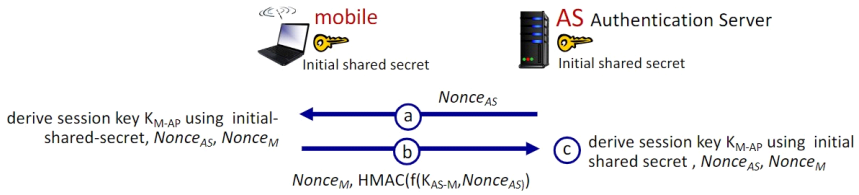
8.8.1 Authentication and Key Agreement in 802.11 Wireless LANs

-Arriving mobile must:
----Associate with access point: (establish) communication over wireless link
----Authenticate to network

-1) Discovery of security capabilities:
----AP advertises its presence, forms of authentication and encryption provided
----Device requests specific forms authentication, encryption desired
----Although device, AP already exchanging messages, device not yet authenticated, does not have encryption keys

-2) Mutual authentication and shared symmetric key derivation:
----AS, mobile already have shared common secret (e.g., password)
----AS, mobile use shared secret, nonces (prevent relay attacks), cryptographic hashing (ensure message integrity) to authenticating each other
----AS, mobile derive symmetric session key

802.11: WPA3 handshake



ⓐ AS generates $Nonce_{AS}$, sends to mobile

ⓑ mobile receives $Nonce_{AS}$
- generates $Nonce_M$
- generates symmetric shared session key $K_{M\text{-}AP}$ using $Nonce_{AS}$, $Nonce_M$, and initial shared secret
- sends $Nonce_M$, and HMAC-signed value using $Nonce_{AS}$ and initial shared secret

ⓒ AS derives symmetric shared session key $K_{M\text{-}AP}$

-a) AS generates Nonce(AS), sends to mobile

-b) Mobile receives Nonce(AS)

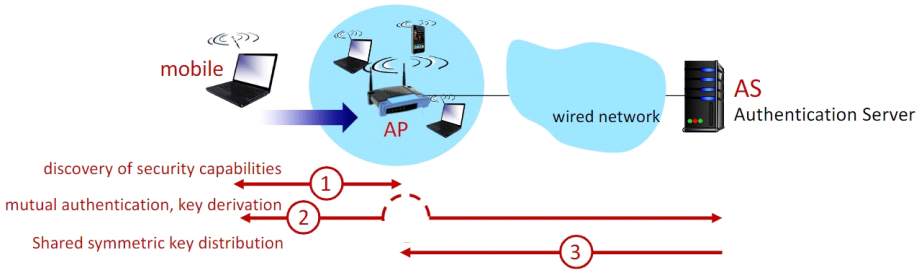
----Generates Nonce(M)

----Generates symmetric shared session key K(M-AP) using Nonce(AS), Nonce(M), and initial shared secret

----Sends Nonce(M), andHMAC-signed value using Nonce(AS) and initial shared secret

-c) AS derives symmetric shared session key K(M-AP)

802.11: authentication, encryption



③ shared symmetric session key distribution (e.g., for AES encryption)
- same key derived at mobile, AS
- AS informs AP of the shared symmetric session

-3) Shared symmetric session key distribution (e.g., for AES encryption)
----Same key derived at mobile, AS
----AS informs AP of the shared symmetric session

-4) Encrypted communication between mobile and remote host via AP
----Same key derived at mobile, AS
----AS informs AP of the shared symmetric session

-EAP = "Extensible Authentication Protocol" (EAP) [RFC 3748] defines end-to-end request/response protocol between mobile device, AS

Mutual Authentication and Shared Symmetric Session Key Derivation

802.11 Security Messaging Protocols

8.8.2 Authentication and Key Agreement in 4G/5G Cellular Networks

Authentication, encryption in 4G LTE

-Arriving mobile must:
----Associate with BS: (establish) communication over 4G wireless link
----Authenticate itself to network, and authenticate network

-Notable differences from WiFi
----Mobile's SIMcard provides global identity, contains shared keys
----Services in visited network depend on (paid) service subscription in home network

CONNECTION

-Mobile, BS use derived session key K(BS-M) to encrypt communications over 4G link

-MME in visited network + HHS in home network, together play role of WiFi AS
----Ultimate authenticator is HSS
----Trust and business relationship between visited and home networks

Authentication, encryption: from 4G to 5G

-4G: MME in visited network makes authentication decision
-5G: home network provides authentication decision
----Visited MME plays "middleman" role but can still reject

-4G: uses shared-in-advance keys
-5G: keys not shared in advance for IoT

-4G: device IMSI transmitted in cleartext to BS
-5G: public key crypto used to encrypt IMSI

8.9) **Operational Security: Firewalls and Intrusion Detection Systems** (699)

KEY SLIDE POINTS:

8.9.1 Firewalls

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

Why Use Firewalls?

-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 (Traditional) packet filters
----Stateful packet filters
----Application gateways

Stateless/Traditional Packet Filters

-Internal network connected to Internet via router firewall

-Filters packet-by-packet, decision to forward/drop packet based on:

----1) Source IP address, destination IP address
----2) TCP/UDP source, destination port numbers
----3) ICMP message type
----4) TCP SYN, ACK bits

--Example 1: block incoming and outgoing datagrams with IP protocol field = 17 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

--**Stateless packet filtering: more examples**

Stateless packet filtering: more examples

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

-OTHER:

Policy: Allow HTTPS web access only ++ Firewall Setting: Accept all incoming packets from any address, port 443)

--**Access Control Lists**

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

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

-First two lines allow TCP traffic (web traffic) as long as the traffic originated from INSIDE the network.

-If the traffic originated from outside, ACK bit won't be set & firewall drops the packet.

PROBLEM: A malicious attacker can manually set flag bit of outside traffic packet to ACK, thereby FOOLING the firewall. Solved by Stateful Packet Filters below

Stateful Packet Filters

-PROBLEM: stateless packet filter = a heavy handed tool
----Admits packets that “make no sense,” e.g., dest port = 80, ACK bit set, even though no TCP connection established:

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

-SOLUTION: 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

-Stateful packet filtering - Augmented ACL:

Stateful packet filtering

ACL augmented to indicate need to check connection state table before admitting packet

action	source address	dest address	proto	source port	dest port	flag bit	check connection
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	X
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---	
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----	X
deny	all	all	all	all	all	all	

-Note that the "Check Connection" field must be set for outside traffic to enter now.

Application Gateway

-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 Routers/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

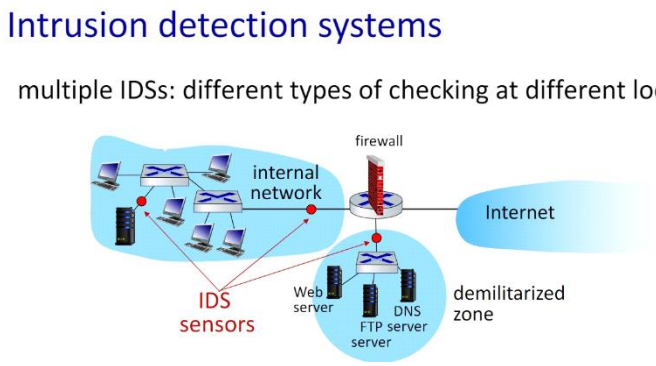
-Many highly protected sites still suffer from attacks

8.9.2 Intrusion Detection Systems

-PROBLEM with Packet filtering:
----It only operates on TCP/IP headers
----No correlation check among sessions

-SOLUTION: 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

Intrusion detection systems image:



-DMZ = Demilitarized Zone = Lower-security region which is protected only by the packet filter, but also monitored by IDS sensors. Note that the DMZ includes the organization's servers that need to communicate with the outside world, such as its public Web server and its authoritative DNS server

TWO TYPES OF IDS

-**Signature-Based System** = Store "Signatures" aka rules pertaining to specific intrusion activity
----May relate to rules about a specific packet's attributes OR to a series of packets.
----Sniff packets individually

----CONS: Resource intensive & only works on recorded attack types ++ IDS can become overwhelmed and miss legitimate threats ++ May give false alarms if legitimate packet has similar attributes to a signature

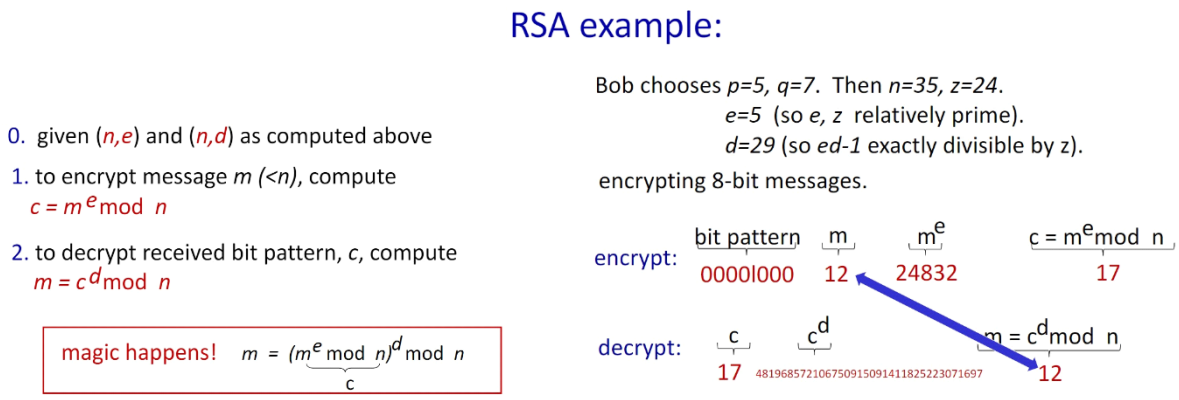
-**Anomaly-Based System**

----Creates 'traffic profile' and looks for statistically unusual traffic.

----Can detect new attack types

Ch 8) **FORMULAS**:

1a) **RSA Encryption - encryption, decryption, example**



1b) **RSA: Creating public/private key pair**

-1) Choose two large prime numbers p, q.(e.g., 1024 bits each)

-2) Compute n = pq, z = (p-1)(q-1)

-3) Choose e (with e < n) that has no common factors with z (e & z are “relatively prime”).

-4) Choose d such that ed - 1 is exactly divisible by z. (in other words: e*d mod(z) = 1).

-5) Public key is (n,e). Private key is (n,d).

1c) **RSA Proof**

Why does RSA work?

- must show that $c^d \bmod n = m$, where $c = m^e \bmod n$
- fact: for any x and y: $x^y \bmod n = x^{(y \bmod z)} \bmod n$
  - where n= pq and z = (p-1)(q-1)
- thus,

$$
\begin{aligned}
c^d \bmod n &= (m^e \bmod n)^d \bmod n \\
&= m^{ed} \bmod n \\
&= m^{(ed \bmod z)} \bmod n \\
&= m^1 \bmod n \\
&= m
\end{aligned}
$$

$$[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n$$
$$[(a \bmod n) - (b \bmod n)] \bmod n = (a - b) \bmod n$$
$$[(a \bmod n) \cdot (b \bmod n)] \bmod n = (a \cdot b) \bmod n$$

MODULO ARITHMETIC PROPERTIES

1d) RSA FIND d:

----Find totient_n = numbers less than 'n' and relatively prime to it (I.e. They share no common factors)
----mmi = e^-1 % totient_n[i]

-----There's no great way to go about doing this. Good Luck.

2) **Substitution Ciphers**

--a) Caeser Cipher

-Substitute alphabet letters k bits to the right to encrypt.

-Substitute alphabet letters k bits to the left to encrypt.

--b) Monoalphabetical Cipher

-Substitute any letter of the alphabet with another.

--c) Polyalphabetic Encryption

-Use multiple different Monoalphabetic Ciphers based on letter positioning

Ch 8) **GLOSSARY**:

3DES = More secure variant of DES

ACL (8.9) = Firewall's table of rules, applied top to bottom to incoming packets: (action, condition) pairs: looks like OpenFlow forwarding (Ch. 4)! (IMAGE in 8.9)

AES = Advanced Encryption Standard (Symmetric Key Encryption Scheme - Cannot brute force < 129 trillion years)

Ciphertext = Encrypted text

DES (8.2) = Data Encryption Shared (Symmetric Key Encryption Scheme - Can brute force it in a day (56 bit specific?))

DMZ (8.9) = Demilitarized Zone = Lower-security region which is protected only by the packet filter, but also monitored by IDS sensors. Note that the DMZ includes the organization's servers that need to communicate with the outside world, such as its public Web server and its authoritative DNS server

EAP = "Extensible Authentication Protocol" (EAP) [RFC 3748] defines end-to-end request/response protocol between mobile device, AS (8.8)

Initialization Vector - Random k-bit strings of numbers or letters (Used in cryptography)

IPsec IKE = "Internet Key Exchange" - Automatic method for setting SA rules in IPSec VPN.

Key = String of numbers or characters used as an input to an encryption algorithm

(Ch. 8) MAC = Message Authentication Code is message m, where s is concatenated with m to create m + s, and the hash H(m + s) is calculated (for example, with SHA-1). H(m + s) is called the **MAC**.

MD5 = Widely used hash function in Authentication/Message Integrity

Nonce = A number (R) only used once-in-a-lifetime. Used to avoid playback attack from working, as the nonce will not be the same in subsequent sessions.

Plaintext = Regular, non-encrypted text

Public key systems = A pair of keys is used. One of the keys is known to both Bob and Alice (indeed, it is known to the whole world). The other key is known only by either Bob or Alice (but not both)

SAD = Security Assoc. Database - Endpoint in IPSec connection holds SA state in security association database (SAD). Router accesses SAD to determine how to process datagram.

SA = Security Association - One way connection in an IPSec connection (Two-way in the case of IKE)

SHA-1 = Widely used hash function in Authentication/Message Integrity

SPD = Security Policy Database - Used to store policy for given datagram - should it use IPSec or not?

SPI = "Security Parameter Index" - 32 bit identifier for an SA (Security Association) in a VPN.

Symmetric key systems = Alice's and Bob's keys are identical and are secret

TLS - Transport-layer security (Widely deployed security protocol above the transport layer ++ Supported by almost all browsers, web servers: https (port 443))

VPN = Virtual Private Networks - Establishes a private network using public networks.

==========

==========

FORMULAS AND INFORMATION

==========

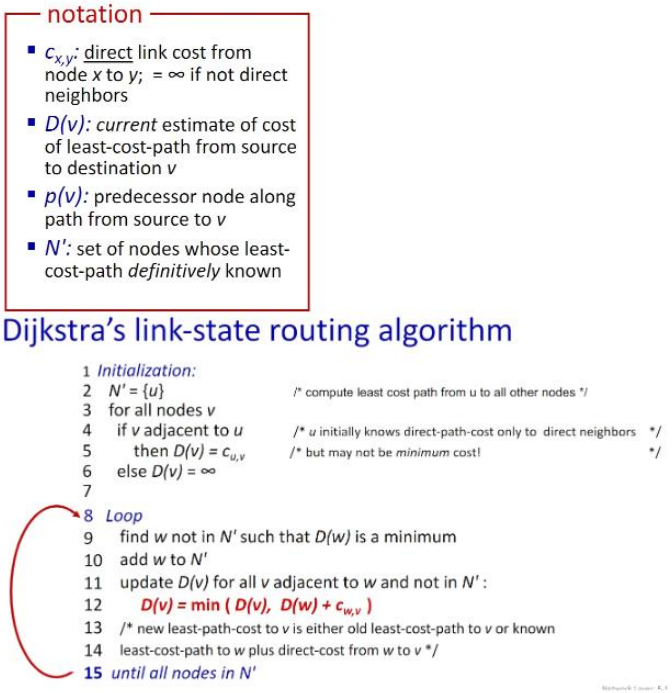
==========

Final Exam Notes

Chapter 5

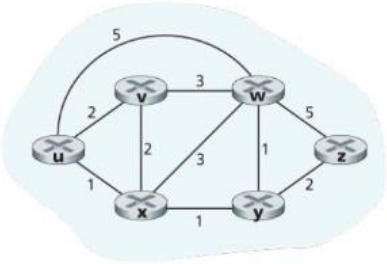
Dijkstra's Routing Algorithm : Link State Routing

1. For each iteration, find the lowest cost, previous node from the set of nodes N'



2. Ties are broken arbitrarily

This is an example of the notation for the homework question.



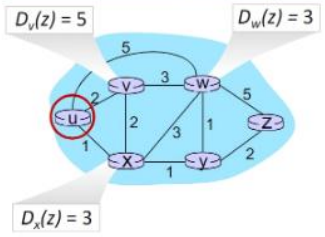
step	N'	D (v), p (v)	D (w), p (w)	D (x), p (x)	D (y), p (y)	D (z), p (z)
0	u	2, u	5, u	1,u	∞	∞
1	ux	2, u	4, x		2, x	∞
2	uxy	2, u	3, y			4, y
3	uxyv		3, y			4, y
4	uxyvw					4, y
5	uxyvwz					

Bellman-Ford : Distance Vector Routing

Bellman-Ford equation

Let $D_x(y)$: cost of least-cost path from x to y.
Then:

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



Bellman-Ford equation says:

$$D_u(z) = \min \{ c_{u,v} + D_v(z), \; c_{u,x} + D_x(z), \; c_{u,w} + D_w(z) \}$$
$$= \min \{2 + 5, \; 1 + 3, \; 5 + 3\} = 4$$

node achieving minimum (x) is next hop on estimated least-cost path to destination (z)

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

Open Shortest Path First - OSPF - Intra AS

- OSPF protocol is also used to supply link metrics for MPLS routers.

Global and Decentralized Algorithms

Route Oscillations

Border Gateway Protocol - BGP - Inter AS

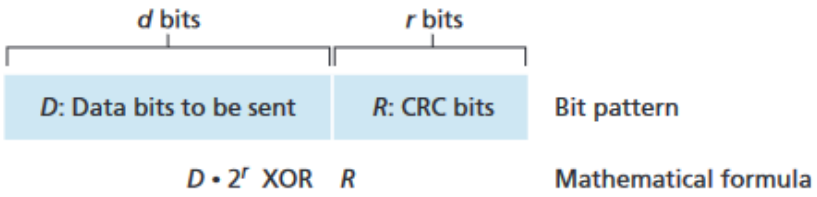
Autonomous Systems (Domains) and Gateways

Software defined network - SDN

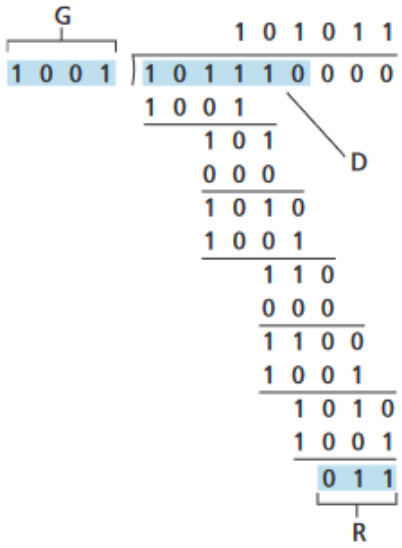
- Separation of data plane and control plane
- Network control functions: external to data-plane switches
- A programmable network

Chapter 6

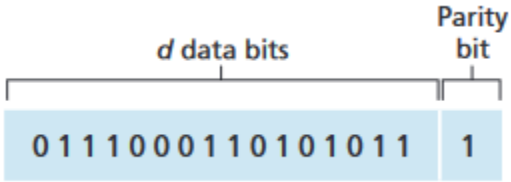
Cyclic Redundancy Check



D=101110
G=1001
Formula to compute R
D*2r XOR R = nG
R = remainder ((D* 2r)/G)



Parity bits



The sender will add either a 0 or 1 to make the number of 1's even or odd based on a predetermined method. This can be translated to a 2D grid as well. Two-dimensional parity can also detect (but not correct!) any combination of two errors in a packet. The ability of the receiver to both detect and correct errors is known as forward error correction (FEC)

MAC Protocols

- Channel Partitioning
  - TDMA - Time Division Multiple Access: allows turns, must wait for turn
  - FDMA - Frequency Division Multiple Access: divides access into frequency bands; access evenly divide
  - CDMA - Code-Division Multiple Access: allows nodes to transmit simultaneously and restores the message based on the sender's code

- Random Access
    - SLOTTED ALOHA
    - Pure ALOha
    - CSMA(/CD) - Carrier Sense Multiple Action (Collision Detection)

Slotted Aloha:

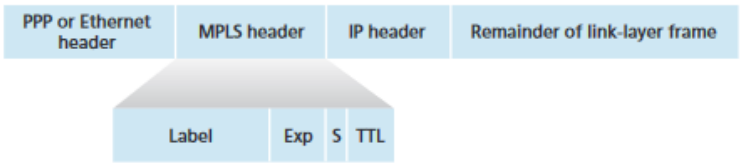
- When the node has a fresh frame to send, it waits until the beginning of the nextslot and transmits the entire frame in the slot.
- If there isn't a collision, the node has successfully transmitted its frame and thus need not consider retransmitting the frame. (The node can prepare a new frame for transmission, if it has one.)
- If there is a collision, the node detects the collision before the end of the slot. The node retransmits its frame in each subsequent slot with probability p until the frame is transmitted without a collision.
- The efficiency of ALOHA is the % chance of a successful transmission on any time slot. With the optimal *p* value the efficiency peaks at ~37% (1/e).

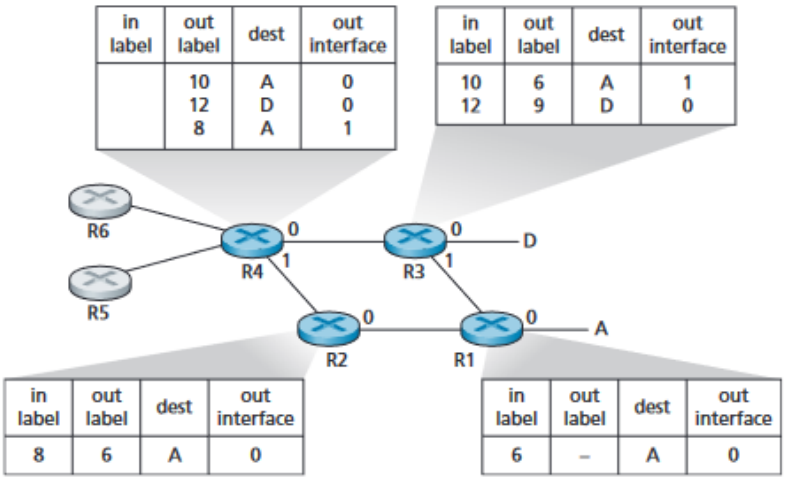
CSMA/CD:

- Listen before speaking. If someone else is speaking, wait until they are finished. This is called carrier sensing
- If someone else begins talking at the same time, stop talking. This is called collision detection
- Efficiency = 1/(1+5dprop/dtrans)
- Collisions are the result of transmission propagation delay. A node mistakenly decides the channel is idle when it is not because the current signal had not physically reached it yet.

MPLS - Multiprotocol Label Switching

- Routers running MPLS (label-switch routers) can decide where to forward a packet on more packet fields than just it's destination address like with regular IP routers.
- Label-switch routers can have multiple forwarding table entries for the same destination, with each of them matching to a tuple of packet header fields. Similar to how TCP has a socket matched for each (source IP, source socket, dest. IP, dest. socket) 4-tuple.





Ethernet

- Main MAC Random Access Protocol for shared wired channels
- Uses unslotted CSMA/CD
- Binary/Exponential Backoff: After a host detects it's n'th collision, it will randomly select a value '*k*' from a set {0, 1, 2, … 2^(n) - 1}. Then it will wait k*512 bit time.

LAN Switch

The switch of the link layer.

- They are directly connected to either a host or another LAN switch.
- Are 'plug-and-play', do not need to be given a configured forwarding table.
- 'Self Learning': the switch populates it's own forwarding table as Frames are sent to it. Recording the source IP & MAC address and mapping it to the interface/adapter it came from.
- 'Transparent': Nodes (Hosts & Routers) don't need to state the switch's address when sending a Frame into the LAN.
- Switch Broadcast: When a switch receives a Frame with a destination MAC address it does not have in its switch table, it broadcasts the frame to all it's interfaces (except for the one it received the frame from).

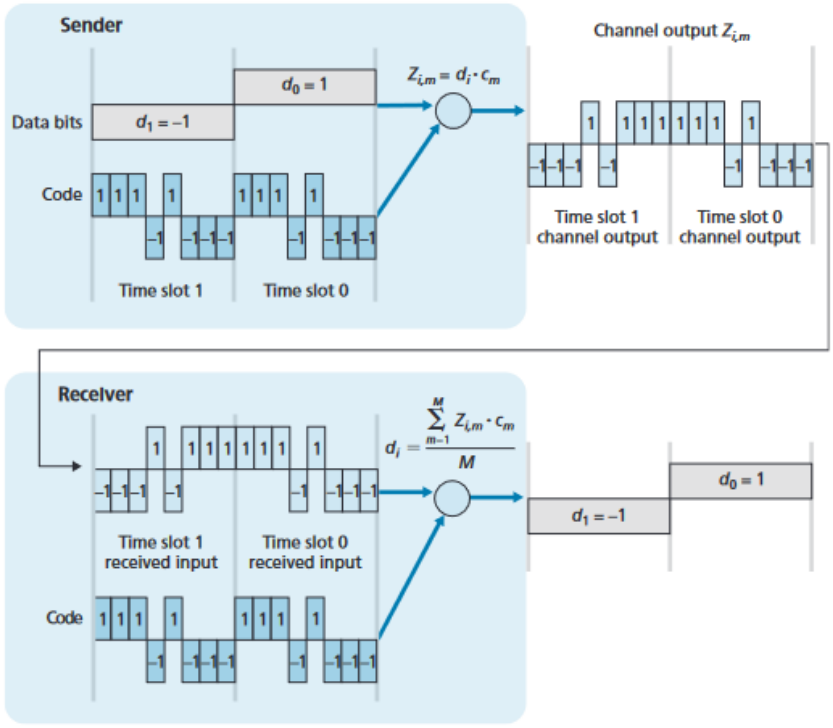
VLAN

- Virtual LAN networks can be implemented by grouping a switche's interfaces into exclusive groups, Frames from the interfaces of one group cannot be sent out of interfaces of another group.
- When a VLAN spans multiple switches, the are connected by a 'Trunk Port' and a VLAN ID field is provided in the frame header to navigate to the correct switch.

Recommended reads:
6.2, 6.3, 6.4, 6.7

Chapter 7

Wireless CDMA:



Frame format for wireless networks:

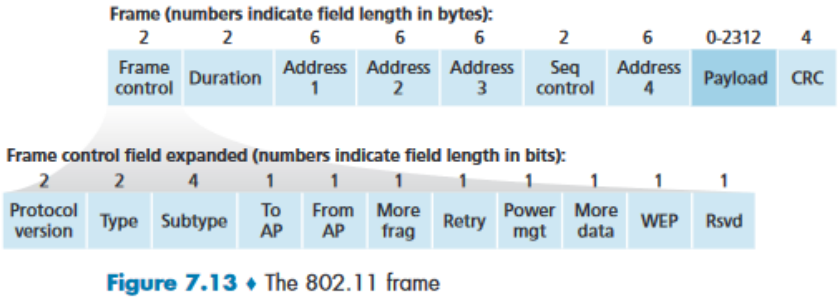


Figure 7.13 ♦ The 802.11 frame

Address 1: MAC address of the wireless station
Address 2: MAC address of the station that transmits the frame
Address 3: Address 3 contains the MAC address of this router interface

ETX - Expected transmission count
Bidirectional link: ETX=1df dr
df = forward = (received/sent)
dr = reverse = (received/sent)

Chapter 8

Types of key systems

- symmetric key systems, keys are identical and are secret.

- public key systems, a pair of keys is used. One of the keys is known to both and the other key is known only to one, but not both

- Symmetric key system: Caesar cipher
    - Caesar cipher would work by taking each letter in the plain-text message and substituting the letter that is *k* letters later (allowing wraparound;) in the alphabet
- An improvement on the Caesar cipher is the **monoalphabetic cipher**, which also substitutes one letter of the alphabet with another letter of the alphabet. However, rather than substituting according to a regular pattern, any letter can be substituted for any other letter, as long as each letter has a unique substitute letter, and vice versa
- Types of attacks on Symmetric key systems
    - Ciphertext-only attack
    - Known-plaintext attack
    - Chosen-plaintext attack
- **Polyalphabetic cipher**
- In modern times we use **Block ciphers** for symmetric key encryption. It's used in things such as Pretty Good Privacy (PGP), TLS, and IPsec.
    - The message to be encrypted is processed in blocks of k bits. For example, if k = 64, then the message is broken into 64-bit blocks, and each block is encrypted independently. To encode a block, the cipher uses a one-to-one mapping to map the k-bit block of cleartext to a k-bit block of ciphertext. Let's look at an example. Suppose that k = 3, so that the block cipher maps 3-bit inputs (cleartext) to 3-bit outputs (ciphertext). One possible mapping is given in Table 8.1. Notice that this is a one-to-one mapping; that is, there is a different output for each input. This block cipher breaks the message up into 3-bit blocks and encrypts each block according to the above mapping. You should verify that the message 010 110 001 111 gets encrypted into 101 000 111 001.

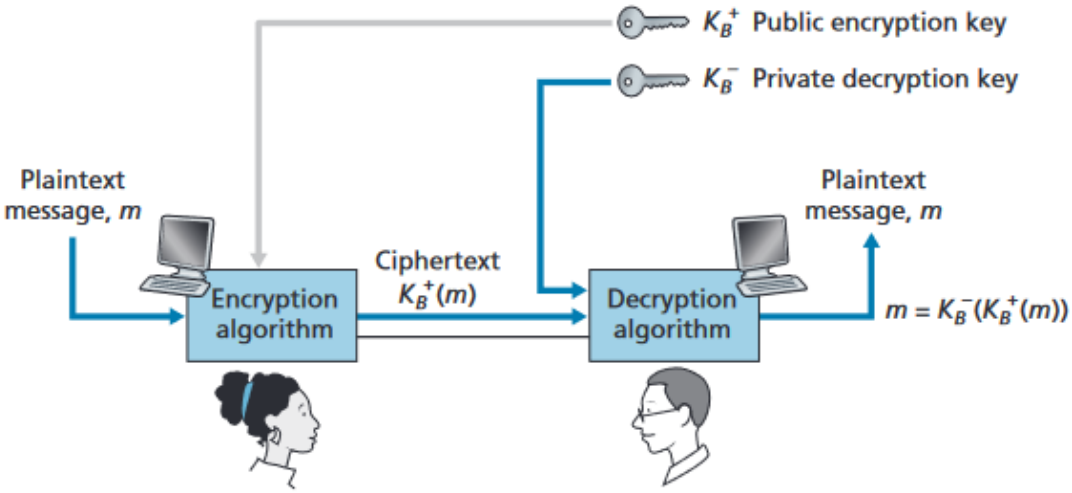
input	output	input	output
000	110	100	011
001	111	101	010
010	101	110	000
011	100	111	001

- **Cipher-Block Chaining**

- **Public key encryption**
    - To generate the public and private RSA keys, perform the following steps:
        i. Choose two large prime numbers, p and q. How large should p and q be? The larger the values, the more difficult it is to break RSA, but the longer it takes to perform the encoding and decoding. RSA Laboratories recommends that the product of p and q be on the order of 1,024 bits.
        ii. Compute $n = pq$ and $z = (p - 1)(q - 1)$.

iii. Choose a number, e, less than n, that has no common factors (other than 1) with z. (In this case, e and z are said to be relatively prime.) The letter e is used since this value will be used in encryption.
iv. Find a number, d, such that ed - 1 is exactly divisible (that is, with no remainder) by z. The letter d is used because this value will be used in decryption. Put another way, given e, we choose d such that
ed mod z = 1
v. The public key that Bob makes available to the world, K + B, is the pair of numbers (n, e); his private key, K - B, is the pair of numbers (n, d)

- To decrypt requires the use of the private key
  i. m = cdmod n
- To encrypt does not require the use of the private key
  i. c = memod n
- c = ciphertext; m=cleartext;



TLS-TCP Handshake

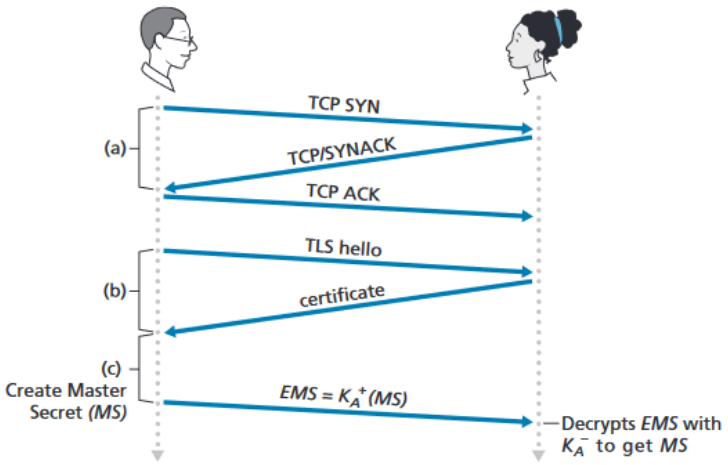


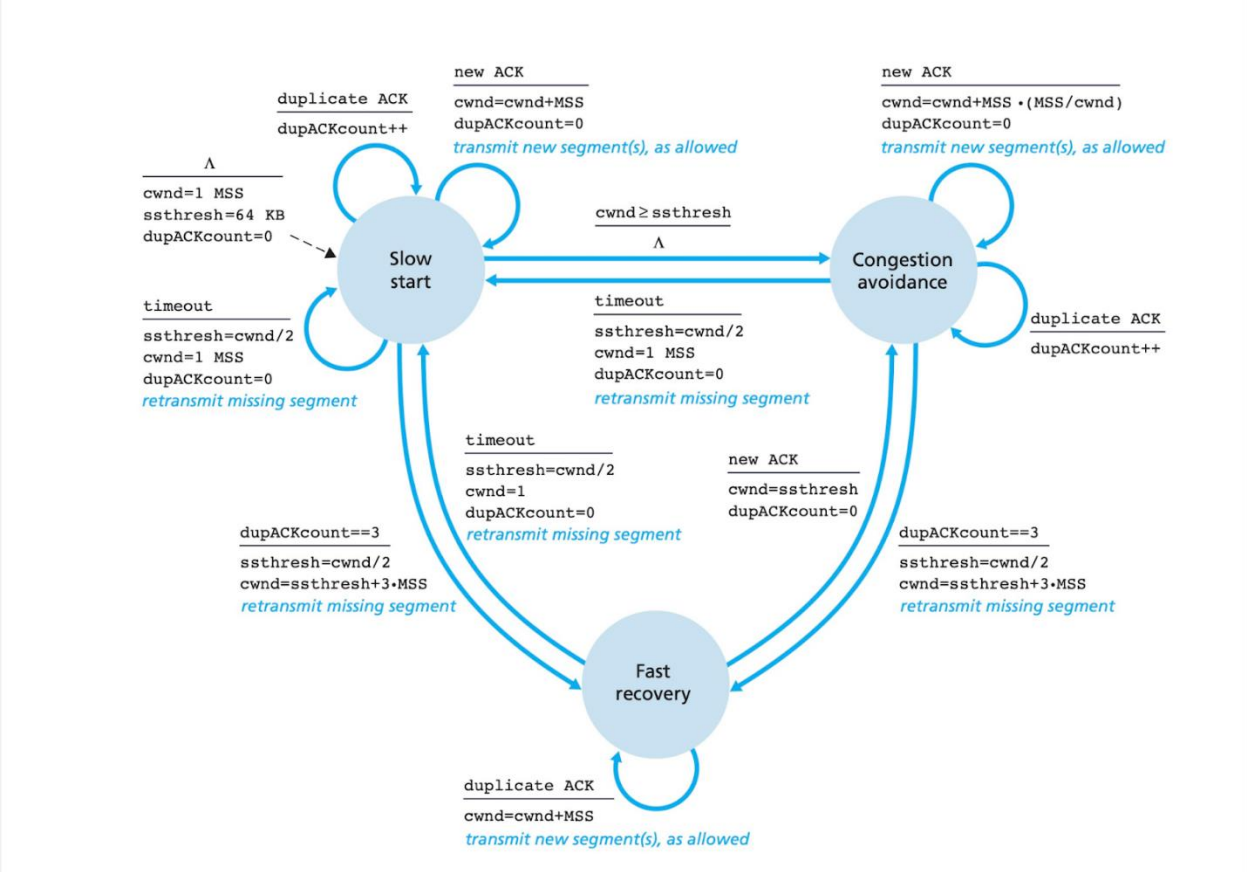
Figure 8.25 ♦ The almost-TLS handshake, beginning with a TCP connection

Need to look at:

- Network-Layer Security: IPsec and VirtualPrivate Networks
- Securing Wireless LANs and 4G/5G Cellular Networks
- Operational Security: Firewalls and Intrusion Detection Systems
- Message Integrity and Digital Signatures

Midterm questions

TCP Reno:



1) SLOW START

- Beginning cwnd = 1 MSS
- Increases by 1 MSS for every time a transmitted segment is first acknowledged. (This occurs for each MSS transmitted, effectively doubling every time as long as successful)
- Timeout resets cwnd to 1 and ssthresh to previous_cwnd/2
- 3 ACK results in Fast Recovery Mode beginning. ssthresh = ceil(ssthresh/2) and cwnd = ssthresh + 3 → Moves to congestion avoidance
- Slow start ends when cwnd >= ssthresh → Congestion Avoidance mode begins.

2) CONGESTION AVOIDANCE

- On entry to CA mode - cwnd is set to ceil(cwnd/2) from when congestion was last encountered.
- Increases cwnd by 1 MSS for every successful RTT (AKA the entire set of MSS sendings must be successful to increase by 1.)
- Timeout resets cwnd to 1 and ssthresh to ceil(previous_cwnd/2) → Moves to slow start
- 3ACK results in Fast Recovery Mode beginning. ssthresh = ceil(sstresh/2) and cwnd = ssthresh + 3 → Returns to congestion avoidance

----If more than 3 ACKS somehow, add more than 3 to first cwnd.

3) FAST RECOVERY

- On entering (see previous steps) set ssthresh = ceil(sstresh/2) and cwnd = new_ssthresh + 3 (one time only to fix 3ACK, then cwnd = ssthresh) → Returns to congestion avoidance
- cwnd receives +1 for each duplicate ACK (including those after Fast Recovery starts)
- Timeout resets cwnd to 1 and ssthresh to ceil(previous_cwnd/2) → Moves to slow start
- New ACK → cwnd = ssthresh, dupACKCount = 0 → Moves to Congestion Avoidance

Subnets:

IPV4 subnets

- IPV4 uses a numerical address made of 32 bits, separated into 4 eight bit numbers, so 8bits . 8bits . 8bits . 8bits, meaning that addresses range from 0.0.0.0-255.255.255.255 for a total of 4,294,967,296 different addresses.
- CIDR-Classless Inter-Domain Routing
- In CIDR for subnets you get notation such as 192.168.0.1/24. Where the /24 tells you how much of the address is the prefix for the network you're on.

The way I like to think of it is, the /x portion is the number of bits that are locked in place and won't change for this subnet regardless of host. So in this example
X=locked; O=unlocked
192.168.0.0/24
XXXXXXXX-XXXXXXXX-XXXXXXXX-OOOOOOOO
Meaning that the first 24 bits are locked and won't change, in binary this would be
11000000 10101000 00000000 00000000 → 11000000 10101000 00000000 11111111
In decimal
192.168.0.0 → 192.168.0.255
Meaning that you can have up to 256 addresses on this subnet.

- To determine how many hosts,
- (TotalBits - CIDRBits)^2 = Number of Hosts
- (32-24)^2 = 8^2 = 256

The first and last address are reserved for the router and broadcast address.

If you wanted to break 192.168.0.0/24 into 3 smaller subnets. This subnet can only be broken down further by locking the last 8 bits.

10000000 = 128
01000000 = 64
00100000 = 32
00010000 = 16
Etc…

First divide the number of addresses by the amount of subnets you want to make, (256/3)=85.33.. This isn't a power of 2, so determine whether the closest subnet above or below it will create 3 or more subnets. If we only lock the first bit, you get

IP/CIDR	Address range	Broadcast
192.168.0.0/25	192.168.0.0 - 192.168.0.127	192.168.0.127
192.168.0.128/25	192.168.0.128 - 192.168.0.255	192.168.0.255

So, /25 isn’t enough

IP/CIDR	Address range	Broadcast
192.168.0.0/26	192.168.0.0 - 192.168.0.63	192.168.0.63
192.168.0.64/26	192.168.0.64 - 192.168.0.127	192.168.0.127
192.168.0.128/26	192.168.0.128 - 192.168.0.191	192.168.0.191
192.168.0.192/26	192.168.0.192 - 192.168.0.255	192.168.0.255

192.168.0.X/26 will work, as it breaks the original address into 3 or more subnets and is the largest possible subnet. Usable addresses are the Address range minus the first and last address.