Network Layer #1

1. Network-layer

- transport segment from sender to receiver
    - sending: encapsulates segments into datagrams
    - receiving: delivers segments to transport layer
    - network layer protocols in every host, router
        - router examines “header field” of IP datagrams passing through it
- Service:
    - individual datagram: guaranteed delivery with less than 40 msec delay
    - flow of datagrams: in-order/ guaranteed min. bandwidth / restrictions on changes
- Planes:
    - Data plane-Forwarding:
        - local per router function
        - from router’s input port to output port
    - Control plane- Routing:
        - network-wide logic
        - route taken from source host to destination host

2. Two control-plane approaches

- Per-router Control Plane => Traditional routing algorithms
    - local forwarding table updated by routing algorithms in every router
    - routers exchange information with neighbors
- Logically Centralized Control Plane => Software-defined networking(SDN)
    - distinct/remote controller(SDN server) interact with local control agents(CAs)
    - local control agents deliver information to remote controllers & receive the forwarding table(local flow table: headers|counters|actions)

3. Router Architecture

- routing processor: software; millisecond; in control plane; 生成&更新路由表
- high-speed switching fabric: hardware; nanosecond; in data plane

4. Input Port

- line termination: in physical layer; bit-level reception 接受物理信号
- link layer protocol: in data link layer(e.g. Ethernet); 执行处理数据链路层行为
- lookup, forwarding: decentralized switching
    - using IP header field values, lookup outport port using local forwarding table in input port memory (“match + action”
    - Destination-based forwarding(多用&传统法): forward based only on destination IP address
        - Longest Prefix Matching
            - often performed using **TCAM(Ternary Content Addressable Memories)**
            - retrieve address in one clock cycle
    - Generalized forwarding: forward based on any set of header field values
- queuing:

- HOL(head-of-the-line) blocking: a queued datagram at front of queue to prevent others in queue from moving forward

5. Switching Fabrics:

- N inputs, desirable switching rate = N*line rates
- 3 Types:
  - Switching via memory: 1st generation
    - under direct control of CPU
    - 复制 2 次:从输入端口复制到 memory, 再复制到输出端口
    - speed limited by memory bandwidth
  - Switching via a Bus
    - 不需要路由选择处理器干预
    - 输入端口预先计划一个内部标签，匹配的输出端口可保存
    - datagram from import port memory is copied directly to output port memory via a shared bus (share 1 copy)
    - switching speed limited by bus bandwidth
  - Switching via Interconnection Network/ Crossbar
    - 纵横式网络能并行转发多个分组
    - advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric

6. Output Port

- Buffering: v(arrive) > v(output line)
  - datagrams can be lost due to congestion, lack of buffers
  - how much?
    - Rule of thumb: B = RTT(round-trip time)* C(link capacity)
    - 250ms * 10Gpbs = 2.5Gbit buffer
    - with N flows: RTT*C/sqrt(N)
- Scheduling Mechanisms:
  - Scheduling: choose next packet to send on link
  - FIFO
  - Priority: send highest priority queued packet
  - RR(Round Robin): 红，绿，红，绿
  - WFQ(Weighted Fair Queuing): generalized RR; each class gets weighted amount of service in each cycle

Network Layer #2

1. Network Layer Protocols:

- routing protocols:
  - path selection
  - RIP, OSPF, BGP
- IP protocols:
  - addressing conventions
  - datagram format
  - packet handling conventions
- ICMP protocol:
  - error reporting
  - router "signaling"

2. IP

TTL(8bit): max number remaining hops (i--)

3. IP fragmentation, reassembly

- transmission links have: MTU(max. transfer size)- largest possible "link-level" frame
  e.g. Ethernet 1500 Bytes, Wi-Fi 2304 Bytes
- divided(fragmented) within network; reassembled only at final destination
- IP header bits used to identify, order related fragements
- frag_flag = 0: last fragmented datagram

4. IP addressing: 32-bit identifier for host, router interface

- interface: connection b/t host/router and physical link
  - routers typically have multiple interfaces
  - host typically has 1/2 interfaces e.g. wired Ethernet, Bluetooth
  - actually connected by: Ethernet switches / WiFi base station
- IP addresses associated with each interface

- Subnet
  - Device interfaces with same subnet part (高位一致 high order bits)of IP addresses
  - Can physically reach each other without intervening router

- CIDR: Classless Inter-Domain Routing
  a.b.c.d/x: x is #bits of subnet part
  CIDR 划分法 网络号|主机号 任意位数

4. DHCP: Dynamic Host Configuration Protocol

- 当我们进入网络，连上 DHCP 服务器，其会从 IP 池中找一个 IP 返回给我们的设备；当离开这个网络区域时，DHCHP 会检测到，并将 IP 回收到 IP 池中

- host dynamically obtain IP form network server when joining network

    - can renew its lease
    - allow reuse (only hold address while connected)
    - additional information can returned together:
        - address of first-hop router for client(called default gateway 默认网关)
        - name and IP address of DNS server
        - Network mask

- DHCP view:
    - host broadcasts: “DHCP discover”
    - server broadcasts back: “DHCP offer”
    - host: “DHCP request”
    - server: “DHCP ack”

5. Hierarchical addressing:
- How does network get subnet part of IP address?
    → get allocated portion of its provider ISP’s address space
    → ISP 是通过ICANN (Internet Corporation for assigned Names & Numbers)
- Hierarchical addressing allows route aggregation / efficient advertisement of routing information[换 ISP 的组织地址单另 or，由于 longest prefix matching 可以直接]

6. Public的反面Private addresses:
- that are used only within a network(subnets)

7. NAT(Network address translation) 网络地址
- 有设定专用的私网地址
- 私网内所有的用户共享一个公网
- 通过一个 NAT translation table 来翻译 WAN side(outside)和 LAN side(inner)
- 映射时 ip 和 port 都会被替换
- 优点:
    - one public IP address for all devices
    - local 改, public 不用改
    - 改 ISP(public)不用改 local
    - security plus
- 缺点
    - violates end-to-end argument
        - routers should only process up to layer 3
        - address shortage can be solved by IPv6
        - NAT must be taken into consideration by app designers e.g. P2P applications
        - NAT traversal: clients cannot connect to server behind NAT

8. IPv6
- Header format helps speed processing/ forwarding

- header changes to facilitates QoS
- fixed-length 40 byte header
- no fragmentation allowed
- *Priority: identify priority among datagrams in flow*
- *flow label: identify datagrams in same flow*
- *next header: identify upper layer protocol for data*

9. Generalized forwarding and SDN(software defined network)
- each router contains a flow table that is computed and distributed by a logically centralized routing controller

[OpenFlow Data Plane Abstraction 自定义？]
- flow: defined by header fields
- generalized forwarding: simple packet-handling rules
    - Pattern: match values in packet header fields
    - Actions for matched packet: drop/forward/modify…

OpenFlow Table:
Rule + Action + Stats(Packet + byte counters – 统计# pkts are matched with)

Network Layer #3

1. Routing protocol: determine "good" paths
- Path = sequence of routers, from source to destination
- "good": least cost, fastest, least congested
- User preference:
    - file transfer: high bandwidth
    - communication: low delay (avoid satellite links)
    - money transfer: security

2. Basic Graph Theoretic Notations
- G= (N,A)

  N: # nodes A: #arcs/edges e.g. 单个点也是graph N = {1}; A = empty set
- Let G be a connected graph(N,A)
    - G contains a spanning tree
    - |A| >= |N|-1 如果小于- 有点没连上
    - 当|A| = |N|-1 → G is a tree
- path: a walk with no repeated node
- cycle: 落脚点是起始点；至少 3 个点 n1-n2-n3. . nL 1=L L>=4
- a graph is connected if 任意两点之间有 path
- tree: connected path without cycle
- spanning tree of connected graph G: 满足 3 个条件
    - a subgraph of G
    - contains all nodes
    - is a tree

3. Link state routing: all nodes know network topology and link costs (have the same information via "link state broadcast"

Dijkstra's Algorithm:
complexity: n nodes; O(n^2)
Oscillations 震荡 possible： e.g. when link cost equals amount of carried traffic

4. **Distance vector routing**: Bellman-Ford equation
- each node sends its own distance vector estimate to neighbors
- a node x receives new DV estimate form neighbor, it updates

$$d_x(y) = \min_v\{c(x,v) + d_v(y)\}$$

cost from neighbor v to destination y

cost to neighbor v

min taken over all neighbors v of x.

5. Count-to-Infinity Problem
- Slow convergence under topology change
- the more unequal the link costs, the longer the convergence time is

A solution: Poisoned reverse
If Z routes through Y to get to X:
Z tells Y: my distance to X is infinite (so Y won't route to X via Z)
*looping with more than 3 nodes cannot be solved;

Centralized Routing	Decentralized Routing
A central entity calculates all paths b/t source and destination nodes; then distributes routing information to all the nodes	Each node exchanges cost and routing information; keep exchanging until routing table converges
Problem: single point of failure; complexity	Problem: convergence and sub-optimality(due to delayed information)
	Bellman-Ford algorithm requires very little information to be stored at the network nodes: - length/cost of outgoing links - identity of every node - cost(shortest path) of immediate neighbors to the destination

Decentralized Routing:
- Each node maintains 2 tables:
  - Distance table
  - Routing table

- Initialization (assuming distance to destination: *d*)
  - $D_v^{(0)} = \infty, v \neq d$
  - $D_d^{(0)} = 0$
- Iterative steps (at h-th step)
  - $D_d^{(h+1)} = 0$ cost from w to d
  - $D_v^{(h+1)} = \min_{w \in N(v)} [D_w^{(h)} + l(v,w)]$, for each $v \neq d$ $(*)$ v : nei

cost from v to d

where $N(v)$ is the set of neighbors of node v

$l(v,w) = \infty$, if w is not a neighbor of v

  - Algorithm is well suited for distributed computing, since (*) can be executed at each node v in parallel.

*is executed at each node v **in parallel and simultaneously**
Pros: the algorithm terminates at most N-1 iterations
Cons:
- how to make all the nodes agree to start/stop each iteration ?? 同步不简单
- how to abort the algorithm and start a new version 特别是 cost changes 时

⇨ Asynchronous 异步 Distributed Algorithm
Basic idea: from time to time, execute the following iteration at each node v

$$D_v = \min_{w \in N(v)} [l(v,w) + D_w] \qquad (+)$$

- each node uses the latest D_w received from its neighbors
- no need of synchronization at all nodes
- only requirement is that a node v will eventually execute the B-F equation and transmit this information to its neighbors

	LS (link-state)	DV(distance vector)
Message complexity	N nodes, E links, O(NE) messages sent	Exchange b/t neighbors only
Speed of convergence	O(n^2) requires O(NE) messages May have oscillations	Convergence time varies May have routing loops Count-to-infinity problem “bad news traverse slowly”
Robustness (if router malfunctions)	Node can advertise incorrect link cost; each node computes only its own table	Node can advertise incorrect path cost; each node’s table used by others; Error propagates through network

Network Layer #4

1. Internet Approach to scalable routing:

- scale: billions of destinations cannot all be stored in routing tables
- Administrative autonomy:
    - Internet = network of networks
    - **Network under the control of one administrative entity**

2. Autonomous System(AS) = domains 自治系统
- Aggregate routers into regions

1) **Intra-AS routing 里**: routing within a single AS
- routing among hosts, routers in the same AS("network")
- All routers in AS must run the same intra-domain protocol
    - **interior gateway protocols(IGP)**
    - Most common intra-AS routing protocols:
        - **RIP: Routing Information Protocol – DV**
        - **OSPF: Open Shortest Path First - LS**
        - **IGRP: Interior gateway routing protocol - DV**
- Routers in different AS can run different intra-domain routing protocol
- Gateway router: at "edge" of its own AS, has link(s) to router(s) in other ASes[can be understood by both sides]

2) **Inter-AS routing间**: routing among ASes
- Gateways perform inter-domain routing (as well as intra-domain routing)

Forwarding table: configured by both 2 routings
- intra-AS routing: determine entries for destinations within AS
- intra & inter: determine entries for external destinations

3**. OSPF : Open Shortest Path First – LS**
- "open": publicly available
- link-state algorithm + routing computation using Dijkstra's algorithm
- link state packet dissemination & topology map at each node
- each router floods OSPF link-state advertisements to all other routers in entire AS
    - carried in OSPF messages directly over IP (rather than TCP or UDP)
    - link state: for each attached link
- a similar protocol: IS-IS routing protocol
    - standardized by ISO based on OSPF(nearly identical)

链路状态算法为 OSPF 使用，它首先通过向全网广播自己和邻近节点的链路状态，从而使得全网每一个节点都生成一个连通图，最后通过 Dijkstra 算法，计算每个节点为 root 节点到全网任意一个节点的最短路径，最后通过回溯，找到该节点要到 AS 内任意一个目标的下一跳节点，再将这些信息整合成该节点的路由表

Advanced features:

- **Security**
  - All OSPF messages authenticated (to prevent malicious intrusion) 可以设置数据包验证，避免非法数据包更新到路由器的路由表中
- **Multi-path support**
  - Multiple same-cost path allowed (only one path in RIP)
- **Multi-metric support**
  - set different link cost based on ToS(Type of service)
    - satellite link cost is set low for best effort ToS
    - high for real-time ToS
- **Multicast extension**: send to multiple destinations at the same time
- **Hierarchical OSPF**: to support a network of large domain
  - **two**-level hierarchy: local area 1, 2, 3… + backbone
  - area border routers: "summarize" distances to networks in own area, advertise to other area border routers
  - backbone routers: run OSPF routing limited to backbone
  - Boundary routers: connect to other ASes

4. BGP: Border Gateway Protocol

- the de facto inter-domain routing protocol
- "Glue" that holds Internet together
- provides each AS a means to
  - allow subnet to advertise to rest of Internet: "I am here"
  - determine "good" routes to other networks based on reachability information and policy [can set as you want]
- eBGP: obtain subnet reachability information form neighboring ASes
- iBGP: propagate reachability information to all AS-internal routers

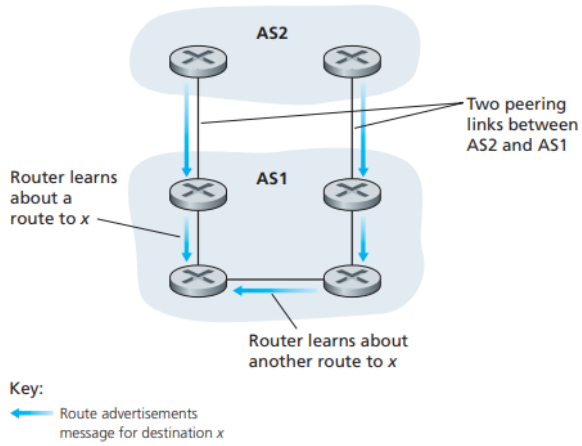
BGP session: two BGP routers exchange BGP messages over semi-permanent TCP connection
BGP is a "path vector" protocol: advertising paths to different dest. network prefixes

"route" = prefix + attributes
Attributes:

- **AS_PATH:** list of ASes through which prefix advertisement has passed
  e.g. AS2, AS3, X (X is prefix, e.g. 138.16.64/24)
- **NEXT_HOP:** indicates specific internal-AS router to next-hop AS

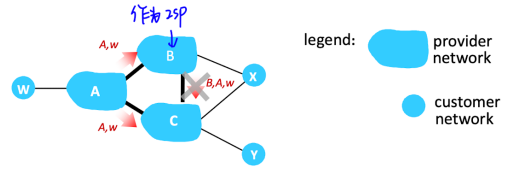
as-path 最主要的作用就是避免从自己发出去的 Route，又被发回来，导致 Route 被不断循环传输，当 BGP Peer 发现发给自己的 Route 包含自己的 AS number 时，会丢弃该 Route

next-hop 则是，当 Route 传输给邻近 AS 的 BGP Peer 时，将把自己的 ip 写入到该 Route 的 next-hop 域．AS1 和 AS2 有两个 BGP 连接，此时 AS2 将 Route 信息同步给 AS1，在 AS1 左下角的 BGP Peer 会收到两个 prefix 和 as-path 相同，但是 next-hop 不同的 route，此时该路由器会根据自己和 next-hop 的距离，决定通过 AS2 要走哪些路径，他会选择路径最短的一条。



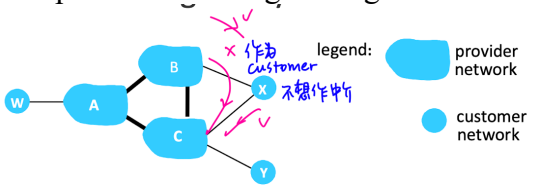
Policy-based routing:

- gateway receiving route advertisement uses import policy to accept/decline path (e.g. never route through AS Y)
- AS policy also determines whether to advertise path to other neighboring ASes or not



Policy: an ISP may want to route traffic to/from its customer networks (It does not want to carry transit traffic between other ISPs)

- A advertises path Aw to B and to C
- B *chooses not to advertise* BAw to C:
  - B gets no "revenue" for routing CBAw, since none of C, A, w are B's customers
  - C does not learn about CBAw path
- C will use route Aw (not using B) to get to w.



- Similarly, a dual-homed customer that is attached to two networks, e.g., X,
  - does not want to route from B to C via X
  - .. so X will not advertise to B a route to C (and vice versa)

Gateway router may learn about multiple paths to destination AS, selects route based on:

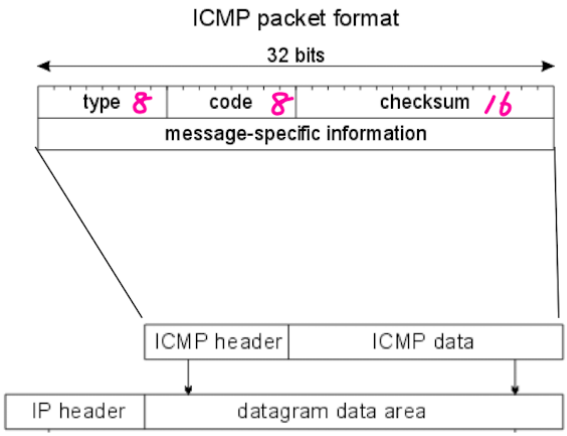
- Local preference value attribute: ***policy decision***
- Shortest AS-PATH: shortest # of AS
- Closest NEXT_HOP router:
  - hot potato routing: choose the local gateway that has the least intra-domain cost 里成本最小

Why different intra-, inter- AS routing?

	Intra-AS 间	Inter-AS
Policy	Everyone is under the control of single admin, no policy decision needed	Admin wants to control over how its traffic routed who routes through its network
Scale	Hierarchical routing saves table size, reduced update traffic	
Performance	Focus on performance	Policy may dominate over performance

5. ICMP: Internet Control Message Protocol

- Error reporting: unreachable host, network, port, protocol
- Echo request/reply: used by ping
- ICMP message: type, code+ first 8 bytes of IP datagram causing error
- ICMP messages carried in IP datagrams: "Network-layer above IP"



Traceroute and ICMP:

- Source sends series of UDP segments to destination with unlikely port number
    - first set has TTL = 1, second set has TTL = 2….
    - when TTL = n, router discards datagram and sends source an ICMP message (type 11, code 0, TTL expired) which includes name of router and IP address
- Stopping criteria:
    - UDP segments eventually arrives at destination host, which returns ICMP "port unreachable" message(type 3, code 3)
    - source stops

6. SNMP: Simple Network Management Protocol
- Protocol to manage data in MIB(Management Information Base)
- Managed devices monitored by agents, who collect data
- Managing entity collect agents' information
- 2 modes

1. request / response mode

2. trap mode (agent 自己 upload)