Circuit switching(time division or frequency division, Centralized switching control, individual nodes are not expected to fail, Dumb terminals, complex network, only app is voice). **Packet Switching(**statistical multiplexing,Dynamic Routing, Complex hosts, dumb network, built on redundant unreliable components, support multiple applications**). Routing msg:** to build the routing table. **Robust system**:build with redundancy and adaptivity

Networking Design Principles:

Fundamental goal: develop an effective technique for multiplexed utilization of all existing networks

1. **Survivability**: continued operation despite partial failures(achieve:States are only stored at edges (hosts)and Stateless packet switches/ routers)(**availability=>**Redundancy, Absence of centralized control, dynamic routes)(fate sharing: comm. depend on the two ends)
2. Heterogeneity above IP: Support **multiple types** of communication services (TCP: reliable delivery, Connectionless datagram,QoS capability)
3. Heterogeneity below IP: Accommodate a **variety of network**s (Transport a packet,Reasonable packet size and reliabilty, Implement functionalities at host)
4. Distributed management of resources(Success in allowing multiple domains and diverse routing policies)
5. Cost effective, performance (Header size, end-to-end retransmissions, routing table lookup)
6. Low cost attachment(Hosts and Nets):Speaking IP, Host software is complex,Relies on correct implementations of host
7. Resource accountability: Challenging in the datagram model

Distance Vector Protocols; RIP Dynamic routing protocol, makes adjustments as necessary BUT bad news travels slow. Count to infinity problem can occur, to handle two-node loops can use **split horizon** (if C uses B to reach A, it will not announce the route to A to B since it uses B) or **poison reverse** (instead of not announcing, it announces infinity, more message overhead but can stop loops faster). Simple to implement, mainly used for smaller networks, Doesn't know the entire and alternate paths. Soft-state, updates every 30 seconds. Hop count is used as link cost, MAX hop count is 16. (IGRP/EIGRP) Every time we are sending updated routing table to others but in OSPF we are flooding same information every time.
(**pros**: Simple, low overhead, distributed)(**Cons**: routing loops, slow convergence, count-to-infinity)

Loop-free Routing Algorithm: Tradeoff-no loops for more memory usage and increased complexity. Router has three tables: **link-cost table**, **routing table** (each entry is dest, cost, nexthop, predecessor aka next-to-last hop, marker), **distance table** (alternate paths via neighbors). The **predecessors** allow loop detection since the entire path can be generated using the predecessors in the routing table. To **fix temporary loops**, use the **feasibility** condition: if the node you want to choose a neighbor has the min distance and is less than the **feasibility distance** (previous min distance), it is safe to use (**passive mode**). Otherwise, go **active mode**; set distance to dest to infinity (cause all packets for that destination to be dropped at this time) and ask neighbors for the route. Neighbors will repeat this process of checking feasibility condition.

Link-State Protocol: OSPF Flood every 30 mins their local link state or topology or local authority information (link state announcement);(memoryless flooding). check **seq. number**, if same drop it. If not same, flood it across the network. The update consists of **originators ip**, **list of its links and cost**, **list of subnets it connects to, sequence no and age; Reliability**:Acknowledgement sent back to make sure everyone has the same state(update state) stored in them else my network become inconsistent for next 30min. During flooding age is incremented by one every hop, in database one by every second max age is one hours after that it drops i**t;** When routers reboot don't send right away wait updates(Database description) to first come in to sync the updates(those updates when link was in fail consition). **Periodic** hello messages to monitor link status if new link use database synchronization. Changes flooded to entire network every 30 secs. Split OSPF to areas to increase **scalability**. Every router in an area knows that area's topology but it doesn't know the neighbors topology. But a **border router** knows topology of its neighboring areas. So a router doesn't know the entire path. It is expected that an operator will have no loops in its area.

Traffic engineering: Route oscillation occurs in heavy traffic all the traffic will keep switching between two paths. Routing metric can be hop count or link delay(processing, transmission, propagation and queuing delay).

Revised metric limits the maximum change in link cost per round (half a hop) to prevent wild swings(max link cost is 3 hops, min link cost depends on link type, most expensive link is 7 times of the least expensive, consider traffic only when it's moderate or heavy).

Utilization/traffic vs hop vs cost graph: if my cost is low my traffic is high and if my cost is high, traffic will be low.
(**Pros**:Fast convergence, No persistent loop, support multiple metrics)(**cons**: Complex, high overhead on messages, CPU, and memory)

Path Vector Routing(BGP): why BGP: accomodate different ISP, allow flexible Routing ploicy(increase profit, deccrease link delay), routing scalability**; BGP** Similar to Distance Vector (share info with neighbors), but spell out **entire path**(by looking at path we can **avoid loops**) and **store all backup paths**. BGP uses a **hard-state**; uses TCP to establish peering session with neighbors, and only update to **announce new route** or **withdraw(message type:open and keep alive and notification)** previously announced routes (keep-alives are sent every 30 seconds, time out after 180 sec) and negative effect under congestion. Initial routing table exchange is similar to OSPF. **Prefix:Classless Inter-Domain Routing**(More efficient address allocation, allow aggregation and Need explicit prefix length or mask)

Rather than path being based on shortest path, it is based on **attributes** sent with announcement: **AS Relationship**, **shortest ASPath**, etc.(**criteria** of route selection: local preference >> shortest AS path >>router ID).

In reality, because of path exploration BGP has a **longer convergence** delay, packet loss, and extra latency to data traffic. 4 events: Tup, Tdown, Tshort, Tlong (link up, link down, shorter path appears, forced longer path). Like RIP, bad **news travels slow; Tdown and Tlong take much longer to converge and on average, more updates, packet loss, and delay**. Path vector prevents loops, but is still vulnerable to invalid paths (**triangle example, node at center goes down**). Theoretical worst convergence time is O(n!):Full mesh, message ordering, process and propagate one update at a time, no MRAI, – No route filtering(sending whole table without filtering), Shortest ASPath, Tdown. Typical case is must faster than this worst case. **MRAI**: if a router get an update, this router does not send middle state to next router, it will wait for 30 seconds to get more update and **send final Consolidated update to next router.** This help in fast convergence. Disadvantage of MRAI is slow down correct information. MRAI doesn't apply to **first route** not to **withdrawal** announcements as well.

Damping : ARP caches negative results as well as positive results, used by worm attacks and flaky edge networks to cause flapping. **Block (damp) frequent** flaps and Reuse after the path has stabilized and Only propagate the final path. **Convergence Time** is from when the flapping stops, to when the entire network learns the stable path. How long to **dampen a route**: depend on penalty(increase everytime when router receive an update), time, cutoff and reuse threshold. Route flap damping was dropped by some ISPs because too much customer complaints customer link are not flapping but for some reason they couldn't access parts of internet which ISPs couldn't figure out so they dropped it. The convergence time depends on timer used at provider end.

Occasional flaps pass through without extra delay and Persistent flaps are suppressed as intended behaviour but due to false damping, **False damping** happens when customer sends a update but this message will propagate over the network so a router will **receive updates from various links and damp the route**. **Secondary charging** occurs when two routers have already reached beyond their cutoff frequency but one of them has reached the reuse threshold, so the path will be forwarded to router which is still in its cutoff frequency this update will result in penalty again and the second router will have to wait for a long time to receive updates. **Muffling effect** says that the first router will have the highest penalty threshold receiving many updates from a router so it doesn't matter in the entire network what penalty is going on because all internal penalty will be down by the time the first router reaches it reuse timer i.e. we achieve the intended behaviour. To damp an unstable route and have intended behaviour we have 2 sol: **selective RFD(**Decreasing trend of route preference is the sign of path exploration, wont increase the penalty**) and Robust damping:** is more general solution and can help BGP in many ways. Damping penalty should increase only after a **real flap**, not every update(Path exploration, secondary charging and potentially other interactions cause multiple updates per flap). RCN {location=A-B,, status=down, seq=1} triggers an update. RCN helps in Speed up routing convergence, Help diagnosis, fix damping. Multiple updates for same RCN will penalise only once. If a RCN send to damping algorithm we check we already has this RCN seq. or not(means already penalised or not) and then send it to other BGP processings.

AS-Policy: ISPs have competitive and cooperative relationship to deliver packet from src to dest. They must **intern-connect**, have **settlement model**(profit) and make **routing conform** to the business relationship. **Exchange router**: owned by neutral party, peers with every participating ISP, rely on exchange router for routing decision, ISP won't like the path suggested by central(exchange) router. So ISP use **exchange LANs**(not a router**),** they are fast(ethernet links), each ISP will put a router to local ethernet and ISP will have a full mesh connectivity to other ISPs. Now, ISPs have the connectivity they can make session with other contracted ISPs. Large ISP are choosy but **content providers** like Netflix and google will peer with anyone(because they want accessibility) because they make **profit** by content. Peering **reduces cost & improves data delivery performance**; private links are **high cost, configuration/maintenance overhead**. Advertise all customer routes at all peering points. No default routing; only send traffic to destinations advertised by AS. To be a peer with an ISP: Handle a single-node outage w/o traffic impact, Single AS number, Backbone capacity, staff capacity, at least 4 Peering locations, Consistent routing advertisements, Address blocks(high aggregation and no address block smaller then /24) and No default routing. AS relationships: Path selection(which path to pick as the best path:customer > peer > provider) and path export(which path to advertise to which neighbor: to control incoming traffic) . Peer only send/take customer traffic, not the providers because customer is paying money not the peer. **Valley-free path**: 0/1 uphill segment, followed by 0/1 peer-peer link, followed by 0/1 downhill segment. Uphill segment: sequence of edges that are customer-provider or sibling-sibling Downhill segment: sequence of edges that are provider-customer or sibling-sibling.To label AS graph with AS relationships, identify top of the hill. Higher node degree => bigger ISP, more likely to be provider aka top of the hill. Comparable node degrees=> more likely to be peers. So pick node with highest degree as top provider, assign transit service direction between every pair, if two ASes provide transit service for each other they are siblings, otherwise one is provider, other is customer. Why to **infer AS Relationship (Network topology):** Scientific understanding, Placement of servers, Business decisions, Security policies and Analyzing BGP convergence.

Topology: Need to define what nodes and links are (switch, router, Point of Presence (PoP: provide long distance coverage and high bandwidth), Autonomous System (AS), ISP). ISP PoP-local access points allowing users to connect to the internet with their ISP, important node in ISP. From ISP PoP, make topology by Hub-and-Spoke (easiest, simplest, but single point of failure, bandwidth limits), dual **hub-and-spoke** (higher cost but more reliable), or **levels of hierarchy**. **Backbone networks(Ebone, telsetra):** Multiple Points-of-Presence (PoPs), Lots of communication between PoPs,Need to accommodate diverse traffic demands, Need to limit propagation delay. Efforts in building provision of the network: cost, traffic volume, technical properties and good quality of service.

Active Probing (actively send traffic to know about the topology, tracerouting(To know how many nodes in the path to reach the destination, also has a TTL if it reaches zero than send back a ICMP packet to src) everything, gives you more control but is more suspicious) vs **Passive probing** (BGP monitoring, collecting updates over time). Problem with traceroute: Missing responses(Time limit exceed, firewall dropped the packet), Alias resolution, Misleading IP addresses, Angry operators who think this is an attack, etc. **Discovery** of node/link: Consult the Registration Databases(like WHOIS: can be out of date), Convert from trace-route results(but converting from routers not exact), Extract from BGP routing data(public BGP data). Measure ISP topology from multiple angles; different places will get you different routes. **Power Laws**: degree exponent(Given a graph, the CCDF, Dd, of an degree, d, is proportional to the degree to the power of the constant, D), rank exponent(Given a graph, the degree, dv, of a node v, is proportional to the rank of the node, rv to the power of the constant, R), Eigen exponent(given a graph of eigenvalues, λi, are proportional to the order, i, to the power of constant, ε). **Implications**: The majority of nodes have small degrees, a non-trivial number of nodes has very large degrees. Power law, B-A model(Incremental growth: Starts with m0 nodes, each step adds one new node with m links.); more likely to join a node to a node with higher degree. **No topology** is complete(Limited collection of paths(some links traversed), Especially links low in the AS hierarchy, backup links)

PrefixHijack: No good way to verify the content of routing announcements. **Operational Practice** to secure BGP:Protect the access of the router and BGP session between neighbor routers, Keep router configuration updated and watch it closely. **Prefix Hijack:** Making false routing announcements to attract other's traffic. Attacker can use same prefix, sub-prefix, super prefic(to hide the identity), unused or unallocated prefixes. **Damage** : Blackhole(Attacker drops all hijacked packets), Imposture(Attacker responds to hijacked traffic, we think we are talking to right guy but not) and Interception (Attacker forwards the traffic to the target prefix after viewing or modifying the information). To prevent this Monitor(collect data from different BGP router and see the information), Detect(validate data), React. Current **practice**:Hardening the router and its connections, Filtering some announcements from neighbors(unallocated and unregistered ones), Limiting the max number of prefixes that a neighbor can send within a short while, having a BGP expert standby, De-aggregating own prefixes to deal with sub-prefix hijacks. If the data path (obtained from traceroute) does not match the announced BGP path, then it might be an interception. When **interception** succeed: Goal is to deceive all neighbors except the existing next hop to the target prefix, If existing route is through a customer or peer, it's safe to announce the false route to other neighbors, If existing route is through a provider, it's safe to announce the false route to other peer or customer neighbors, Trial-and-error in practice.