Feladat-konzultáció

- Általában 9-17 között vagyok a tanszéken (I.B.128 vagy I.B.111)
- Biztos időpont (email egyeztetés szükséges)

Forgalomirányítás (routing) alapjai

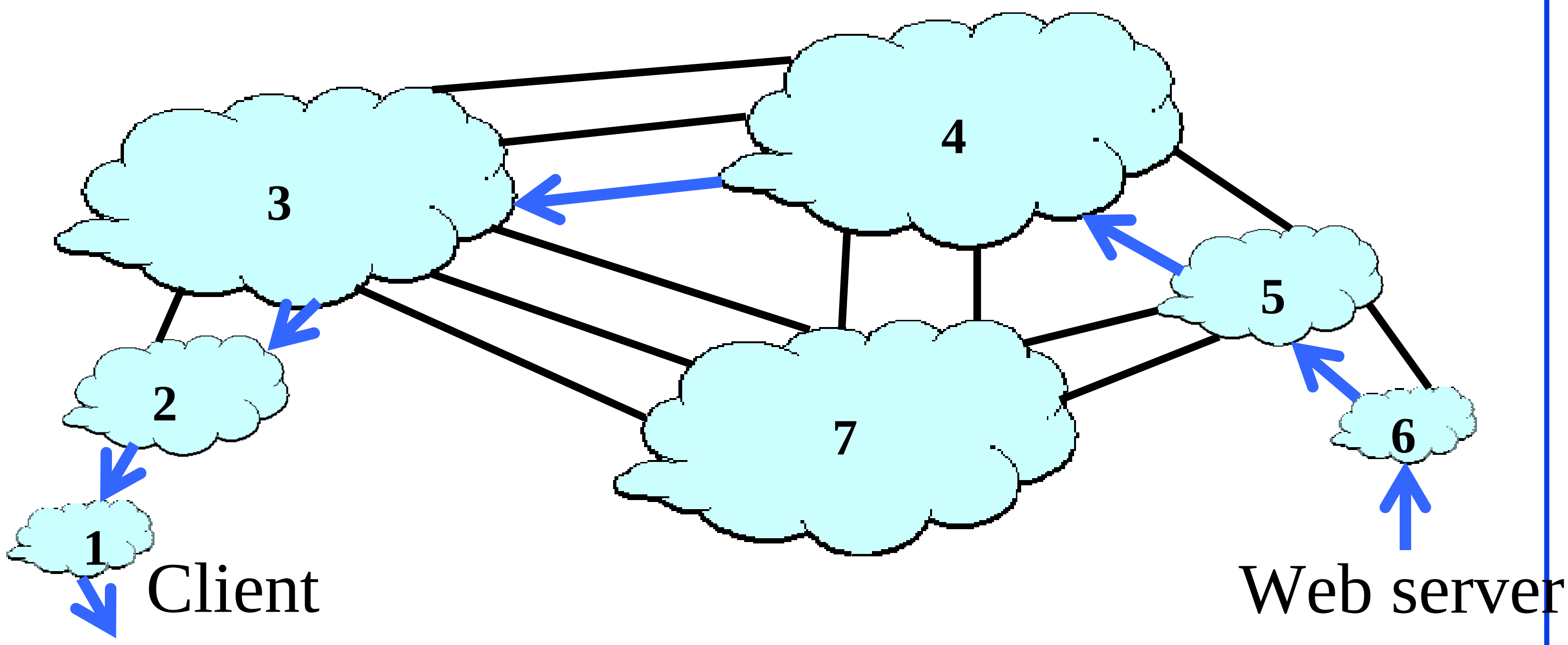
Téma

- **Routing vs Forwarding**
- **Forwarding table vs Forwarding in simple topologies**

- Routers vs Bridges:
- Routing Problem
- Telephony vs Internet Routing
- Source-based vs Fully distributed Routing
- Distance vector vs Link state routing
- Addressing and Routing: Scalability

Internet útvonalirányítás

- Internet egyik lényeges komponense
  - Routerek irányítják a forgalmat
  - Szolgáltatók szabályozhatják a forgalmat a hálózatának linkein
  - Üzleti kapcsolatok kialakítása: tranzit (előfizetéses) és peering



Kerülő útas (indirection) forgalmi irányítás

- Kerülő útvonal a routerek sorozata
  - *Elosztott*
- FIB (Forwarding table): lokális feltérképezés (mapping)
- Globális állapot (distance vectors, link state) feldolgozása a hasznos lokális feltérképezéshez (table entries)
- Felmerült problémák
  - **Konzistencia**: út (series of indirections) vezessen az adott célállomáshoz!
  - **Teljesség (Completeness)**: legyen út az összes célállomáshoz
  - **Efficiency/Performance:**
    - Az e2e (end to end) út hatékonysága, az e2e út kapcsolata a QoS nyújtásához és a hálózat erőforrásához (data plane),
    - Címek összegzése (control plane)

Routing vs. Forwarding

- Forwarding: select an output port based on destination address and routing table
  - Data-plane function: *using* the indirection (in tables)
  - Often implemented in hardware

- Routing: process by which routing table is *built..*
  - … so that the series of local forwarding decisions takes the packet to the destination with high probability, and …(reachability condition)
  - … the path chosen/resources consumed by the packet is *efficient* in some sense… (optimality and filtering condition)

  - Control-plane function: *setting* up indirections
  - Implemented in software

Forwarding Table

- Can display forwarding table using "netstat -rn"
  - Sometimes called "routing table"

Destination	Gateway	Flags	Ref	Use	Interface
127.0.0.1	127.0.0.1	UH	0	26492	lo0
192.168.2.	192.168.2.5	U	2	13	fa0
193.55.114.	193.55.114.6	U	3	58503	le0
192.168.3.	192.168.3.5	U	2	25	qaa0
224.0.0.0	193.55.114.6	U	3	0	le0
default	193.55.114.129	UG	0	143454	

Forwarding Table *Structure*

- Fields: *destination, gateway, flags, ...*
- ***Destination***: can be a host address or a network address. If the ‘H’ flag is set, it is the host address.
- ***Gateway:*** router/next hop IP address. The ‘G’ flag says whether the destination is directly or indirectly connected.
- U flag: Is route up ?
- G flag: router (indirect vs direct)
- H flag: host (dest field: host or n/w address?)

Where do Forwarding Tables Come From?

- Routers have forwarding tables
  - Map prefix to outgoing link(s)
- Entries can be statically configured
  - E.g., “map 12.34.158.0/24 to Serial0/0.1”
- But, this *doesn’t adapt*
  - To failures
  - To new equipment
  - To the need to balance load
  - …
- That is where routing protocols come in…

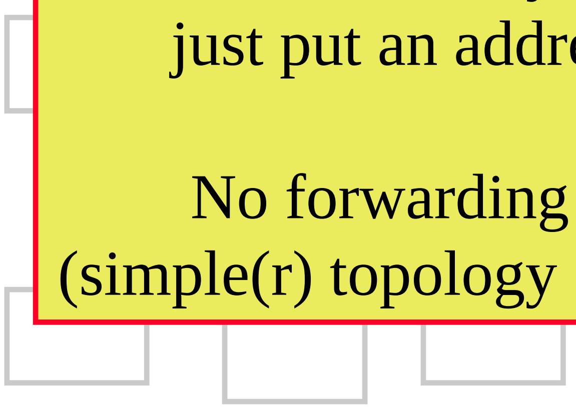
Routing in Simple Topologies

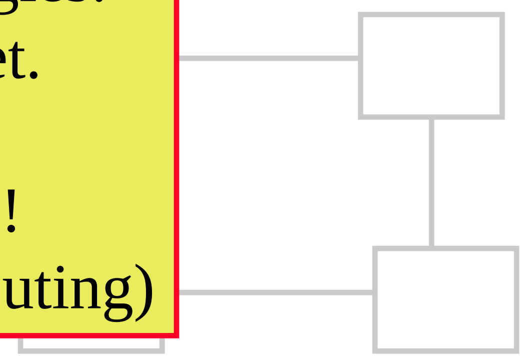
Bus: Drop pkt on the wire…

: port# = dest-addr

Indirection is easy in these topologies:
just put an address in the packet.

No forwarding tables required!
(simple(r) topology => simple(r) routing)



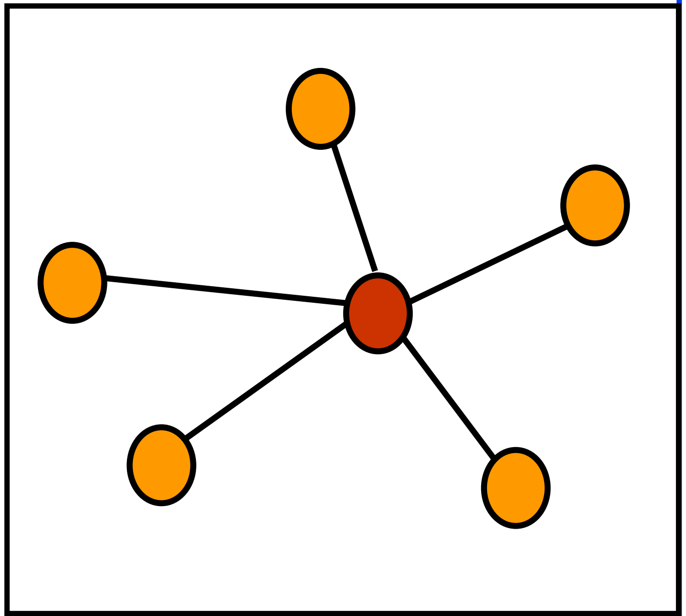


Ring: send packet consistently in (anti-)clockwise direction

Star: stubs point to hub; hub behaves like full mesh

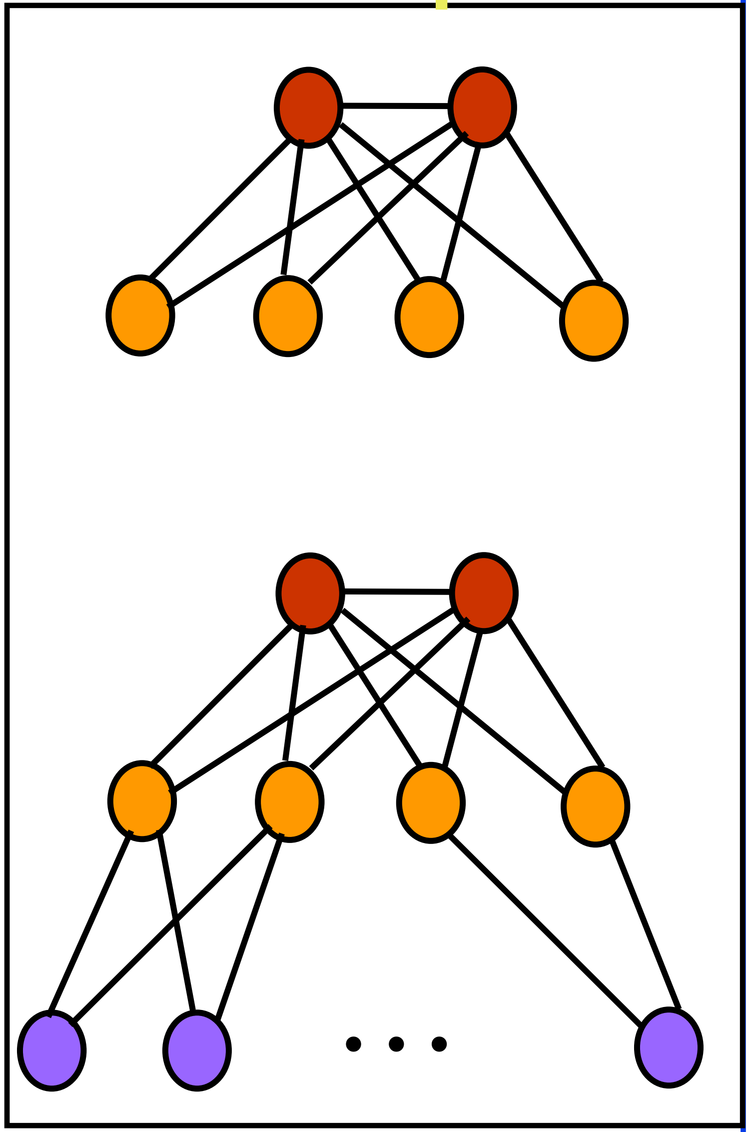
Hub-and-Spoke Topology

- Single hub node
  - Common in enterprise networks
  - Main location and satellite sites
  - Simple design and trivial routing
- Problems
  - Single point of failure
  - Bandwidth limitations
  - High delay between sites
  - Costs to backhaul to hub



Simple Alternatives to Hub-and-Spoke

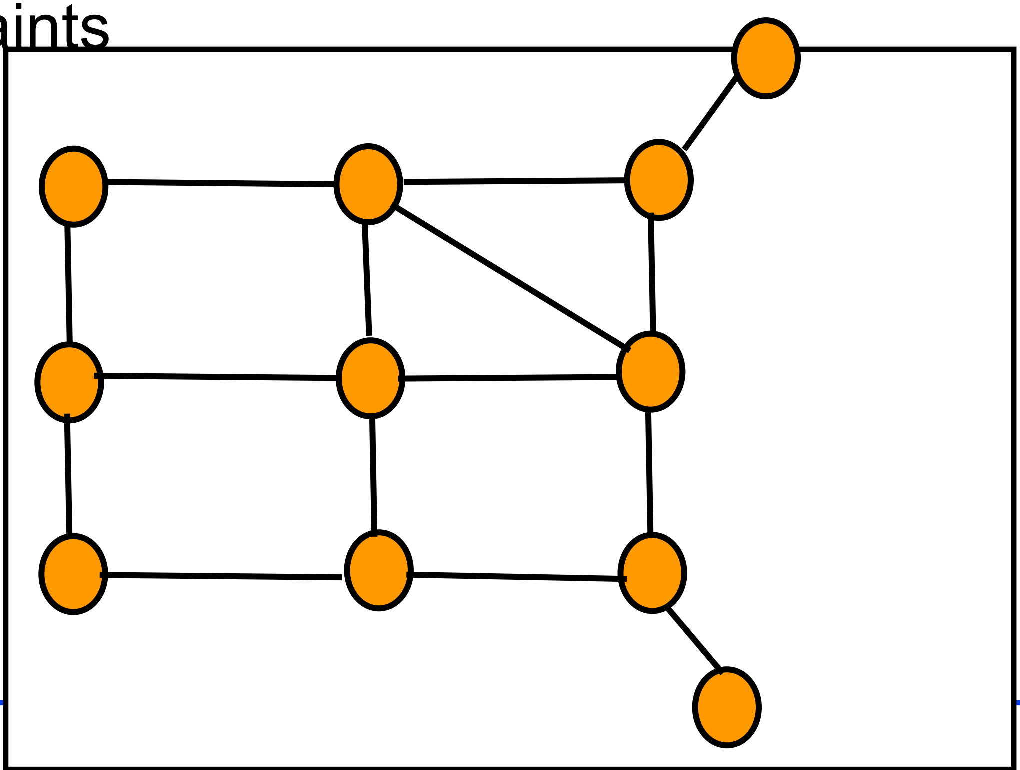
- Dual hub-and-spoke
  - Higher reliability
  - Higher cost
  - Good building block



- Levels of hierarchy
  - Reduce backhaul cost
  - Aggregate the bandwidth
  - Shorter site-to-site delay

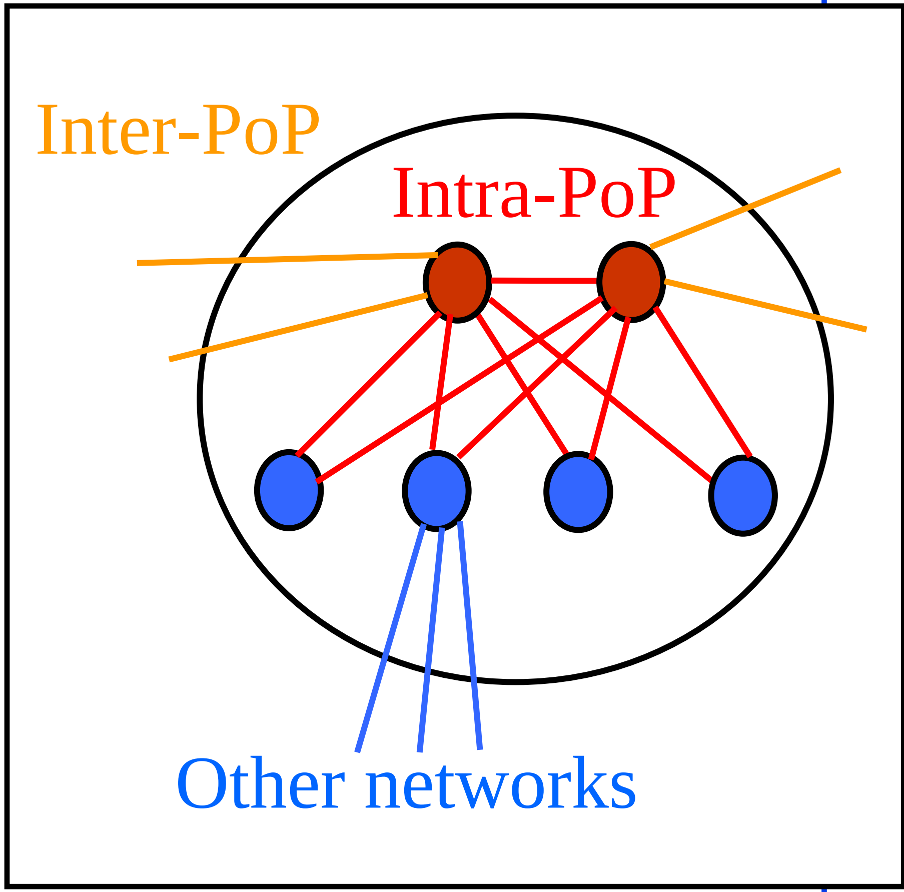
Need for Complex Topology in Backbone Networks

- Backbone networks
  - Multiple Points-of-Presence (PoPs)
  - Lots of communication between PoPs
  - Need to accommodate diverse traffic demands
  - Need to limit propagation delay
  - Cost constraints



Points-of-Presence (PoPs): Topologies

- Inter-PoP links
  - Long distances
  - High bandwidth
- Intra-PoP links
  - Short cables between racks or floors
  - Aggregated bandwidth
- Links to other networks
  - Wide range of media and bandwidth



Where are we?

- Routing vs Forwarding
- Forwarding table vs Forwarding in simple topologies

- **Routers vs Bridges: review**

- Routing Problem
- Telephony vs Internet Routing
- Source-based vs Fully distributed Routing
- Distance vector vs Link state routing
- Addressing and Routing: Scalability

Recall… Layer 1 & 2

- **Layer 1:**
  - *Hubs* do not have “forwarding tables” – they simply broadcast signals at Layer 1. No filtering.
- **Layer 2:**
  - Forwarding tables not required for simple topologies: *simple forwarding rules suffice*
  - The next-hop could be *functionally* related to destination address (i.e. it can be computed without a table explicitly listing the mapping).
    - This places too many *restrictions* on topology and the assignment of addresses vis-à-vis ports at intermediate nodes.
  - Forwarding tables could be statically (*manually*) configured once or from time-to-time.
    - Does not accommodate *dynamism* in topology

Recall… Layer 2

- Even reasonable sized LANs cannot tolerate above restrictions
- *Bridges* therefore have "*L2 forwarding tables*," and use dynamic learning algorithms to build it locally.
  - Even this allows LANs to scale, by limiting broadcasts and collisions to collision domains, and using bridges to interconnect collision domains.
  - The learning algorithm is *purely local*, *opportunistic* and expects no addressing structure.
  - Hence, bridges often may not have a forwarding entry for a destination address (I.e. *incomplete*)
  - In this case they resort to *flooding* – which may lead to duplicates of packets seen on the wire.
  - Bridges coordinate "globally" to build a spanning tree so that flooding doesn't go out of control.

Layer 3 Routing

- *Routers* have "*L3 forwarding tables*," and use a distributed protocol to *coordinate* with other routers to *learn and condense a global view* of the network in a *consistent and complete* manner.

- Routers NEVER broadcast or flood if they don't have a route – they "pass the buck" to another router.
  - The good filtering in routers (I.e. restricting broadcast and flooding activity to be within broadcast domains) allows them to *interconnect broadcast domains*,

- Routers communicate with other routers, typically neighbors to collect an abstracted view of the network.
  - In the form of distance vector or link state.
  - Routers use algorithms like Dijkstra, Bellman-Ford to compute paths with such abstracted views.

Summary so far

- If *topology is simple and static, routing is simple* and may not even require a forwarding table

- If *topology is dynamic, but filtering requirements are weak* (I.e. network need not scale), then a local heuristic setup of forwarding table (bridging approach) suffices.

- Further, if a) filtering requirements are strict,
  b) optimal/efficient routing is desired, and
  c) we want small forwarding tables and bounded control traffic, then …

some kind of global communication, and smart distributed algorithms are needed to condense global state in a consistent, but yet complete way …

Advanced routing: What Are The Issues?

- Routers are efficient in the collection of the abstracted view (control-plane filtering)
- Routers accommodate a variety of topologies, and sub-networks in an efficient manner
- Routers are organized in hierarchies to achieve scalability; and into autonomous systems to achieve complex policy-control over routing.
- Routers then condense paths into next hops, either:
  - depending upon other routers in a path to compute next-hops in a consistent manner (fully distributed), or
  - using a signaling protocol to enforce consistency.
- Advanced routing algorithms support "QoS routing" and "traffic engineering" goals like multi-path routing, source-based or distributed traffic splitting, fast re-route, path protection etc.

Where are we?

- Routing vs Forwarding
- Forwarding table vs Forwarding in simple topologies
- Routers vs Bridges: review

- **Routing Problem**

- Telephony vs Internet Routing
- Source-based vs Fully distributed Routing
- Distance vector vs Link state routing
- Addressing and Routing: Scalability

Routing problem

- Collect, process, and condense global state into local forwarding information
- Global state
  - inherently large
  - dynamic
  - hard to collect
- *Hard issues:*
  - *consistency, completeness, scalability*
  - Impact of resource needs of sessions

Consistency

- Defn: A series of independent local forwarding decisions must lead to connectivity between any desired (source, destination) pair in the network.
- If the states are inconsistent, the network is said not to have "*converged*" to steady state (I.e. is in a transient state)
  - Inconsistency leads to *loops*, wandering packets etc
  - In general a part of the routing information may be consistent while the rest may be inconsistent.
  - Large networks => inconsistency is a scalability issue.
- Consistency can be achieved in two ways:
  - Fully distributed approach: a consistency criterion or invariant across the states of adjacent nodes
  - Signaled approach: the signaling protocol sets up local forwarding information along the path.

Completeness

- *Defn:* The network as a whole and every node has sufficient information to be able to compute all paths.
  - In general, with more complete information available locally, routing algorithms tend to converge faster, because the chances of inconsistency reduce.
  - But this means that more distributed state must be collected at each node and processed.
  - The demand for more completeness also limits the scalability of the algorithm.
- Since both consistency and completeness pose scalability problems, large networks have to be structured hierarchically and abstract entire networks as a single node.

Design Choices …

- *Centralized vs. distributed* routing
  - Centralized is simpler, but prone to failure and congestion
  - Centralized preferred in traffic engineering scenarios where complex optimization problems need to be solved and where routes chosen are long-lived

- *Source-based (explicit) vs. hop-by-hop (fully distributed)*
  - Will the source-based route be *signaled* to fix the path and to minimize packet header information?
    - Eg: ATM, Frame-relay etc
  - Or will the route be condensed and placed in each header? Eg: IP routing option
  - Intermediate: *loose source route*

Design choices…

- Static vs Dynamic Routing:
  - a) ‘route’ command **[Static]**
  - b) ICMP redirect message.**[Static]**
  - c) routing daemon.Eg: ‘routed’ **[Dynamic, connectionless]**
  - d) A signaling protocol **[Dynamic, virtual-circuit]**

Static vs Dynamic

Statically

Administrator manually configures forwarding table entries

+ More control
+ Not restricted to destination-based forwarding
- Doesn't scale
- Slow to adapt to network failures

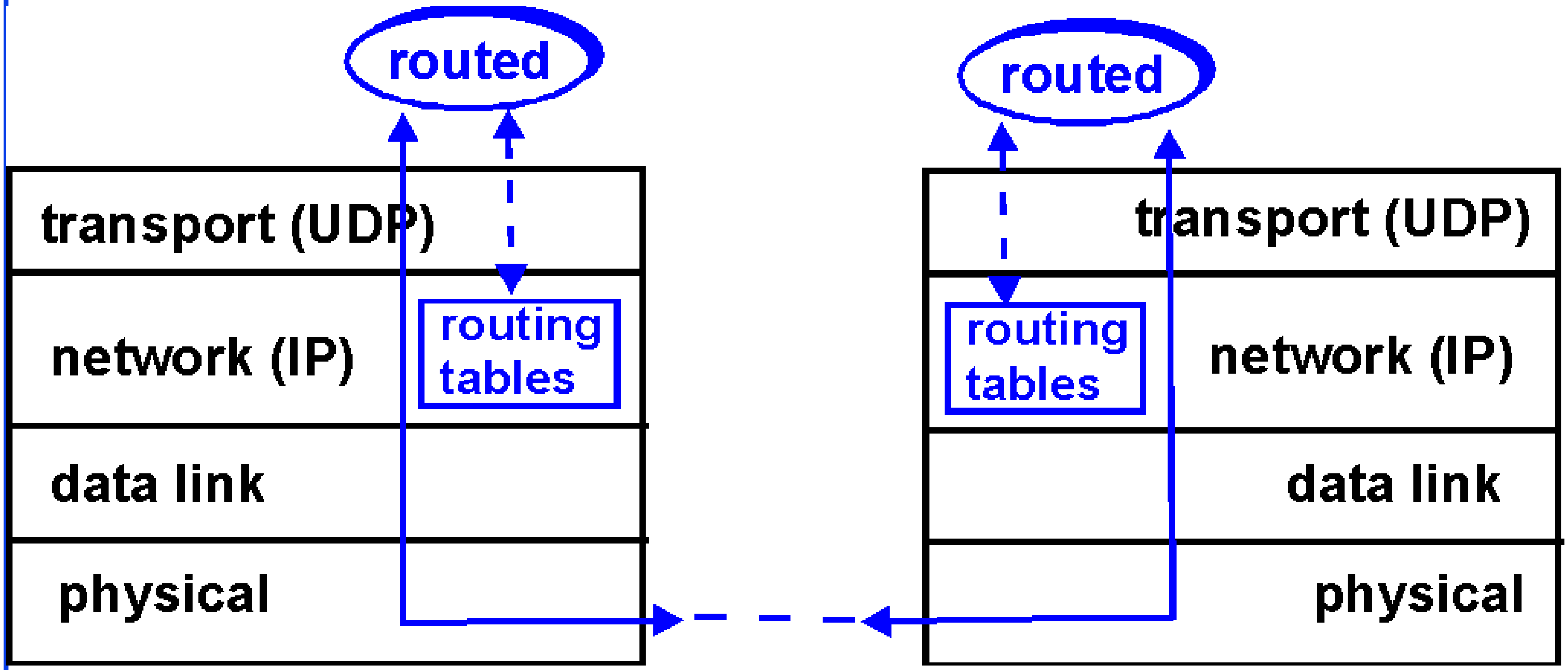
Dynamically

Routers exchange network reachability information using <u>ROUTING PROTOCOLS</u>. Routers use this to compute best routes

+ Can rapidly adapt to changes in network topology
+ Can be made to scale well
- Complex distributed algorithms
- Consume CPU, Bandwidth, Memory
- Debugging can be difficult
- Current protocols are destination-based

Practice : a mix of these.
Static routing mostly at the "edge"

Example Dynamic Routing Model



Telephony Routing algorithm

- If endpoints are within same CO, directly connect
- If call is between COs in same LEC, use one-hop path between COs
- Otherwise send call to one of the cores
- Only major decision is at toll switch
  - one-hop or two-hop path to the destination toll switch.

- Essence of telephony routing problem:

  which *two-hop* path to use if *one-hop* path is full

  (almost a static routing problem…)

Features of telephone routing

- Resource reservation aspects:
  - Resource reservation is coupled with path reservation
    - Connections need resources (same 64kbps)
    - Signaling to reserve resources and the path
  - Stable load
    - Network built for voice only.
    - Can predict pairwise load throughout the day
    - Can choose optimal routes in advance
- Technology and economic aspects:
  - Extremely reliable switches
    - Why? End-systems (phones) dumb because computation was non-existent in early 1900s.
    - Downtime is less than a few minutes per year => topology does not change dynamically

Features of telephone routing (contd)

- Source can learn topology and compute route
- Can assume that a chosen route is available as the signaling proceeds through the network
- Component reliability drove system reliability and hence acceptance of service by customers

- Simplified topology:
  - Very highly connected network
  - Hierarchy + full mesh at each level: simple routing
  - High cost to achieve this degree of connectivity
- Organizational aspects:
  - Single organization controls entire core
  - Afford the scale economics to build expensive network
  - Collect global statistics and implement global changes

=> Source-based, signaled, simple alternate-path routing

Internet Routing Drivers

- Technology and economic aspects:
  - Internet built out of cheap, unreliable components as an overlay on top of leased telephone infrastructure for WAN transport.
    - Cheaper components => fail more often => topology changes often => needs dynamic routing
  - Components (including end-systems) had computation capabilities.
    - Distributed algorithms can be implemented
  - Cheap overlaid inter-networks => several entities could afford to leverage their existing (heterogeneous) LANs and leased lines to build inter-networks.
    - Led to multiple administrative “clouds” which needed to inter-connect for global communication.

Internet Routing Model

- 2 key features:
  - Dynamic routing
  - Intra- and Inter-AS routing, AS = focus of admin control
- Internet organized as *"autonomous systems" (AS).*
  - AS is internally connected
- Interior Gateway Protocols (IGPs) within AS.
  - Eg: RIP, OSPF, IS-IS
- Exterior Gateway Protocols (EGPs) for AS to AS routing.
  - Eg: BGP-4

Two-Tiered Internet Routing Architecture

- Goal: distributed management of resources
  - Internetworking of multiple networks
  - Networks under separate administrative control
- Solution: two-tiered routing architecture
  - Intradomain: inside a region of control
    - Okay for routers to share topology information
    - Routers configured to achieve a common goal
  - Interdomain: between regions of control
    - Not okay to share complete information
    - Networks may have different/conflicting goals
- Led to the use of different protocols…

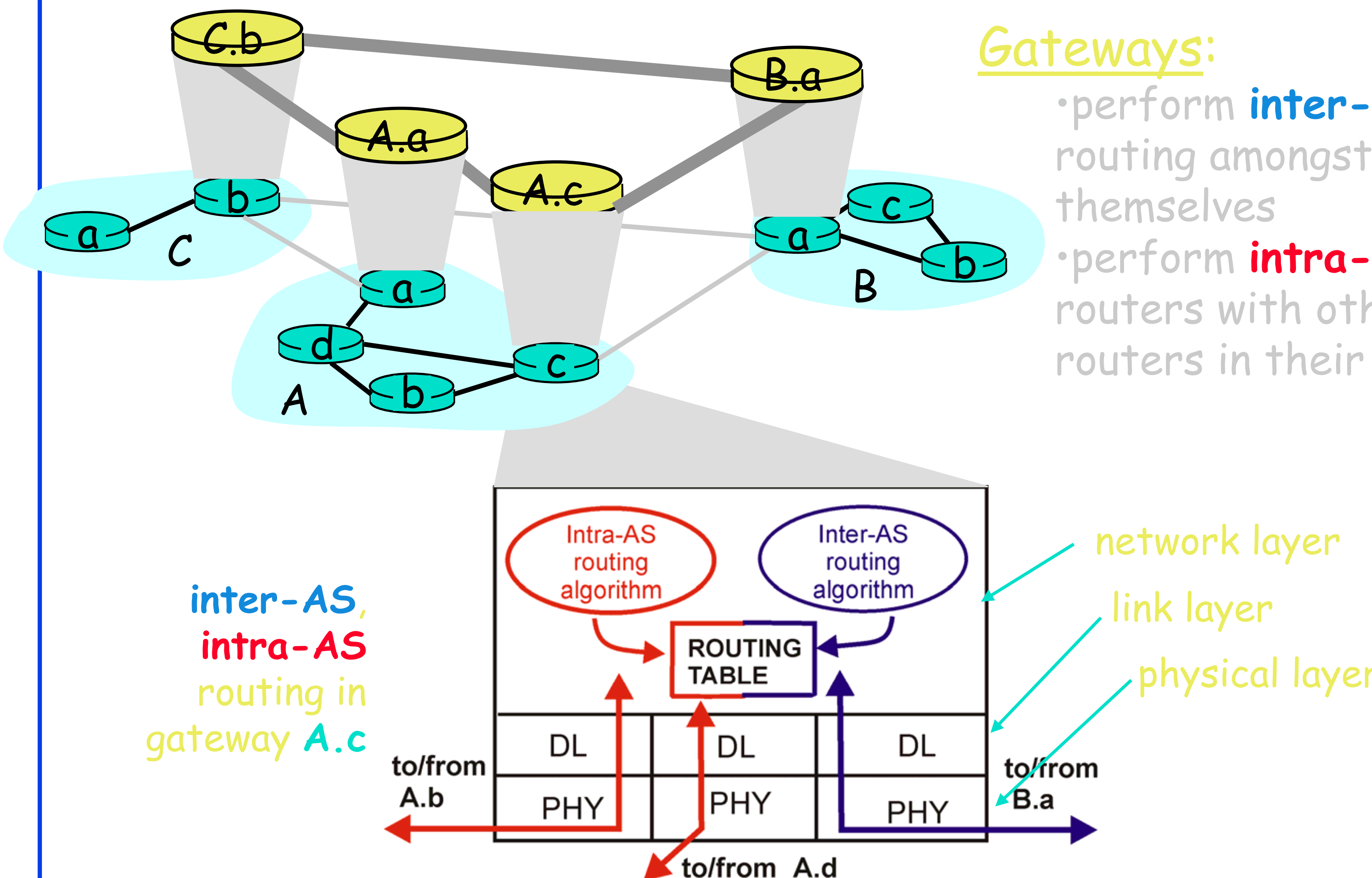
Requirements for Intra-AS Routing

- Should scale for the size of an AS.
  - Low end: 10s of routers (small enterprise)
  - High end: 1000s of routers (large ISP)
- Different requirements on routing convergence after topology changes
  - Low end: can tolerate some connectivity disruptions
  - High end: fast convergence essential to business (making money on transport)
- Operational/Admin/Management (OAM) Complexity
  - Low end: simple, self-configuring
  - High end: Self-configuring, but operator hooks for control
- Traffic engineering capabilities: high end only

Requirements for Inter-AS Routing

- Should scale for the size of the global Internet.
  - Focus on *reachability*, not optimality
  - Use *address aggregation* techniques to minimize core routing table sizes and associated control traffic
  - At the same time, it should allow *flexibility in topological structure* (eg: don't restrict to trees etc)
- Allow policy-based routing between autonomous systems
  - Policy refers to *arbitrary preference among a menu of available options* (based upon options' *attributes*)
  - In the case of routing, options include advertised AS-level routes to address prefixes
  - Fully distributed routing (as opposed to a signaled approach) is the only possibility.
  - Extensible to meet the demands for newer policies.

Intra-AS and Inter-AS routing

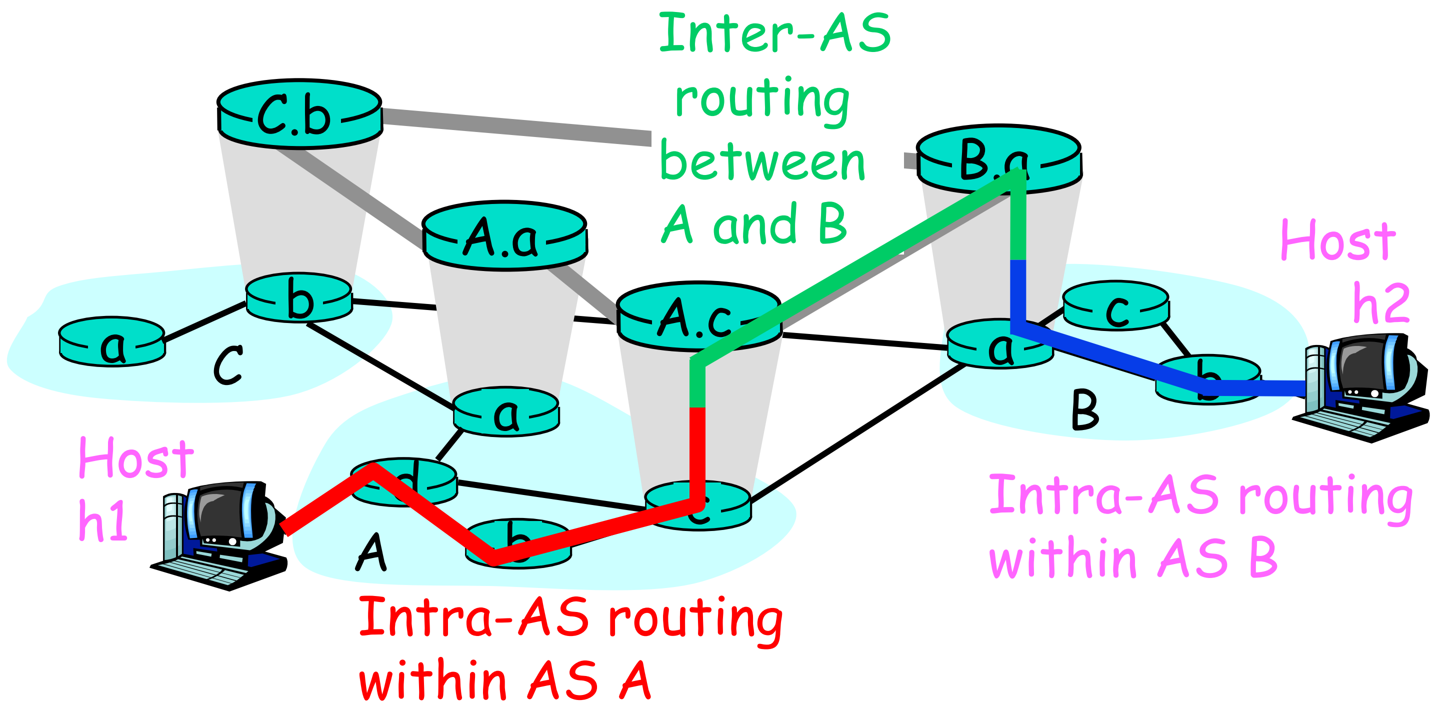


Gateways:

- perform inter-AS routing amongst themselves
- perform intra-AS routers with other routers in their AS

inter-AS, intra-AS routing in gateway A.c

Intra-AS and Inter-AS routing: Example



Summary: Internet Routing Architecture

- Divided into Autonomous Systems
  - Distinct regions of administrative control
  - Routers/links managed by a single “institution”
  - Service provider, company, university, …
- Hierarchy of Autonomous Systems
  - Large, tier-1 provider with a nationwide backbone
  - Medium-sized regional provider with smaller backbone
  - Small network run by a single company or university
- Interaction between Autonomous Systems
  - Internal topology is not shared between ASes
  - … but, neighboring ASes interact to coordinate routing