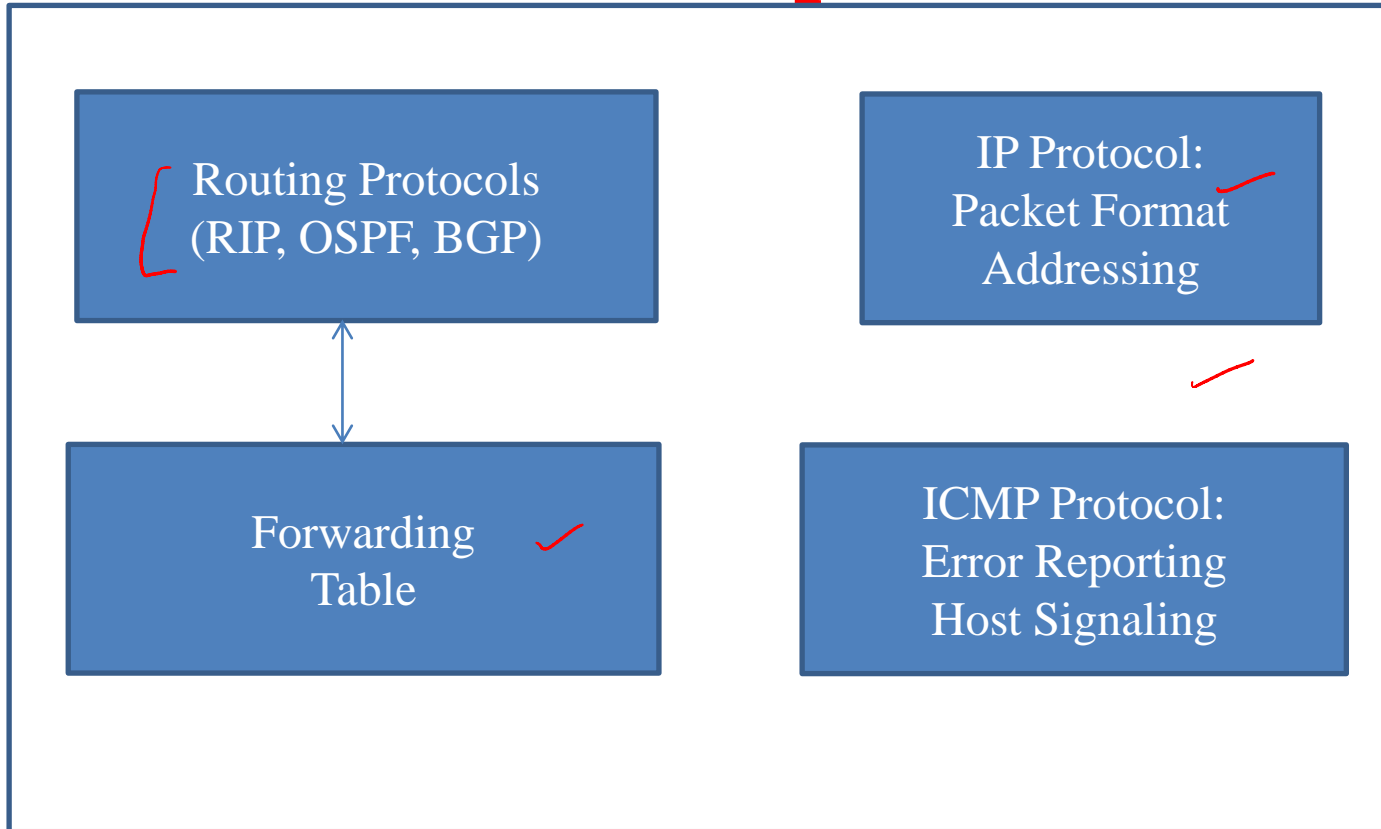



Routing -- Overview

Kameswari Chebrolu

Service Model Implementation

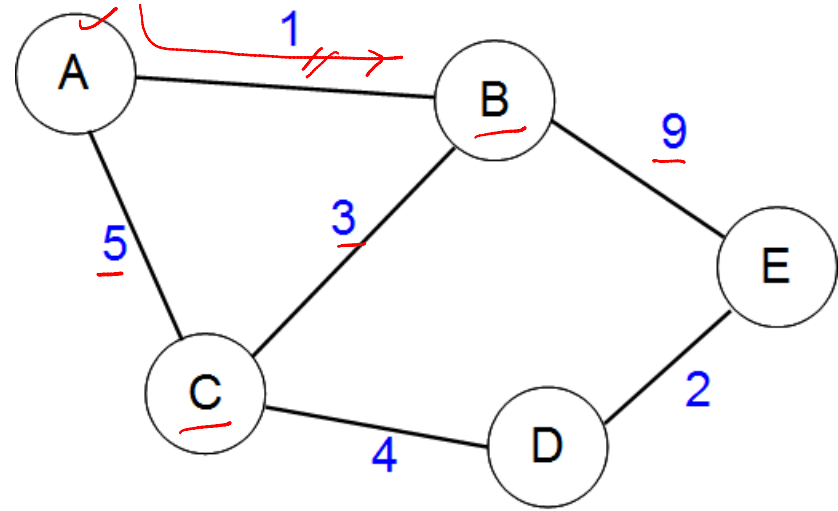


Background

- Role of network layer is ‘end-host delivery’
 - We looked at how packets are forwarded
- How are forwarding tables built? Via Routing Protocols
 - dest 
- Routing vs Forwarding
- Routing domain: All routers under same administrative control
 - E.g. University network, ISP network

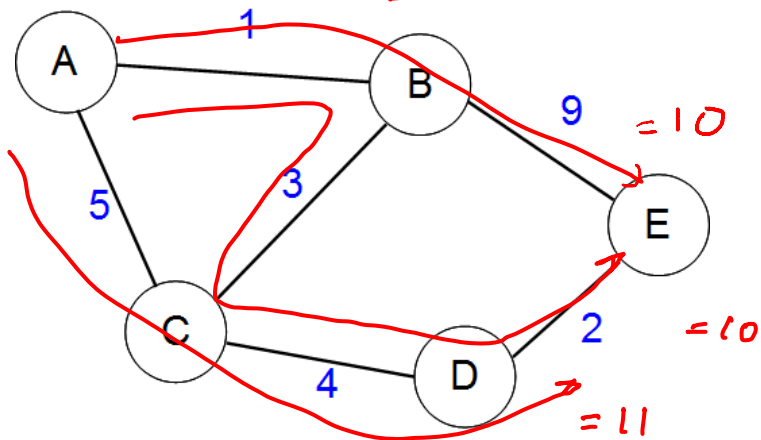
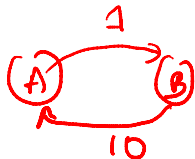
Theory

- Based on Graph theory
- Nodes: Hosts, **Routers**, Networks
vertices
- Edges: Correspond to physical links
 - Edges associated with a cost
 - No edge \rightarrow infinite cost
- Neighbor: Directly connected nodes



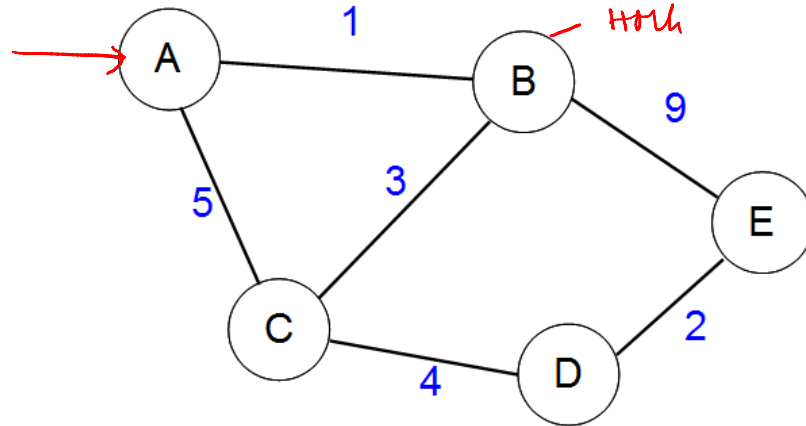
Goal of Routing

- Single Pair ‘shortest’ path problem: Find least cost path between two nodes
 - Path cost is sum of the costs of the individual edges
 - Assumption: Links are undirected



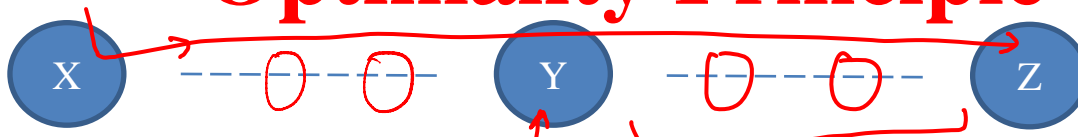
Goal of Routing

- Single-source 'shortest' path problem: Find least cost path from a source to all other nodes in the graph
- Refer to Dijkstra's algorithm



Optimality Principle

Least cost path



- Router Y is on the optimal path from X to Z \rightarrow Y to Z is also an optimal path

other least cost path

destination source




- Set of optimal routes to a destination from all sources form a 'sink tree' routed at the destination
 - Sink tree need not be unique
 - No loops \rightarrow each packet delivered within finite hops
 - For undirected links, a given source to all destinations also forms a tree
 - Routing algorithms helps find sink trees for all routers

Implementation Approach

- Static vs Dynamic
 - Static: Route computed in advance and downloaded in all routers
 - Dynamic: Handles changes in the topology
 - Nodes failure, addition of new nodes, variation in cost

Dynamic preferable over static

Implementation Approach

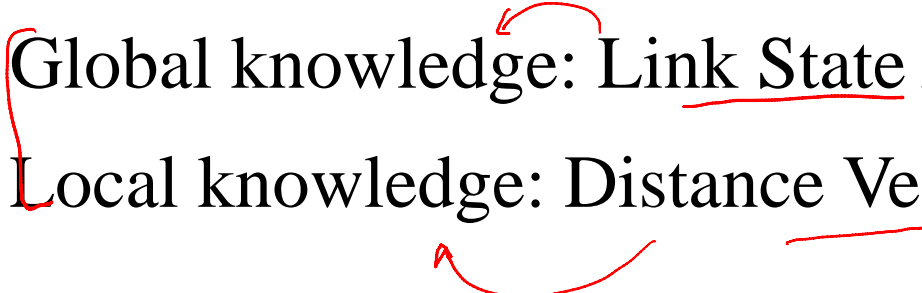
- Central vs distributed processing
- Central:  fault-tolerance, scalability issues
 - All nodes pass neighborhood information to a central node network topology
 - Central node calculates routes and distributes to all
- Distributed: Each node determines routes by itself complex

Distributed preferable to Central

Implementation Approach

- Global vs Local information
- Global: Node calculates routes based on full knowledge of entire topology
- Local: Node does not have global information, determine routes based on local message exchange

Popular Approaches

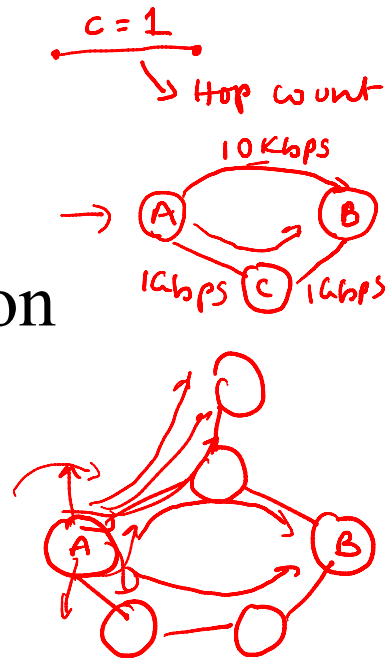
- Dynamic, distributed algorithms
 - Global knowledge: Link State Algorithm
 - Local knowledge: Distance Vector Algorithm
- 

Desirable Features

- Optimality: Least cost paths
- Correctness: Path actually exists
- Simplistic: Easy to implement
- Robust: Handle router/link failures
- Stable: Fast convergence to equilibrium after state change
- Minimal overhead: No. and frequency of message exchange
- Scalable: Handle large number of nodes

Cost Metric

- Cost = 1 \rightarrow Hop count
 - Doesn't distinguish between links based on bandwidth, delay, current load, losses etc
- Static: $1/\text{link_bandwidth}$
- Dynamic: Queue Length, Delay
 - Not stable (ping-pong effect)
- Reality: Links assigned 'static' cost by administrators (e.g. $\text{Constant}/\text{link_bandwidth}$)



Summary

- Routing based on graph theory
- Goal of routing is to find ‘optimal’ path between nodes
- Many approaches to routing
 - Popular: dynamic, distributed based on global/local information
- Up ahead: Popular routing algorithms