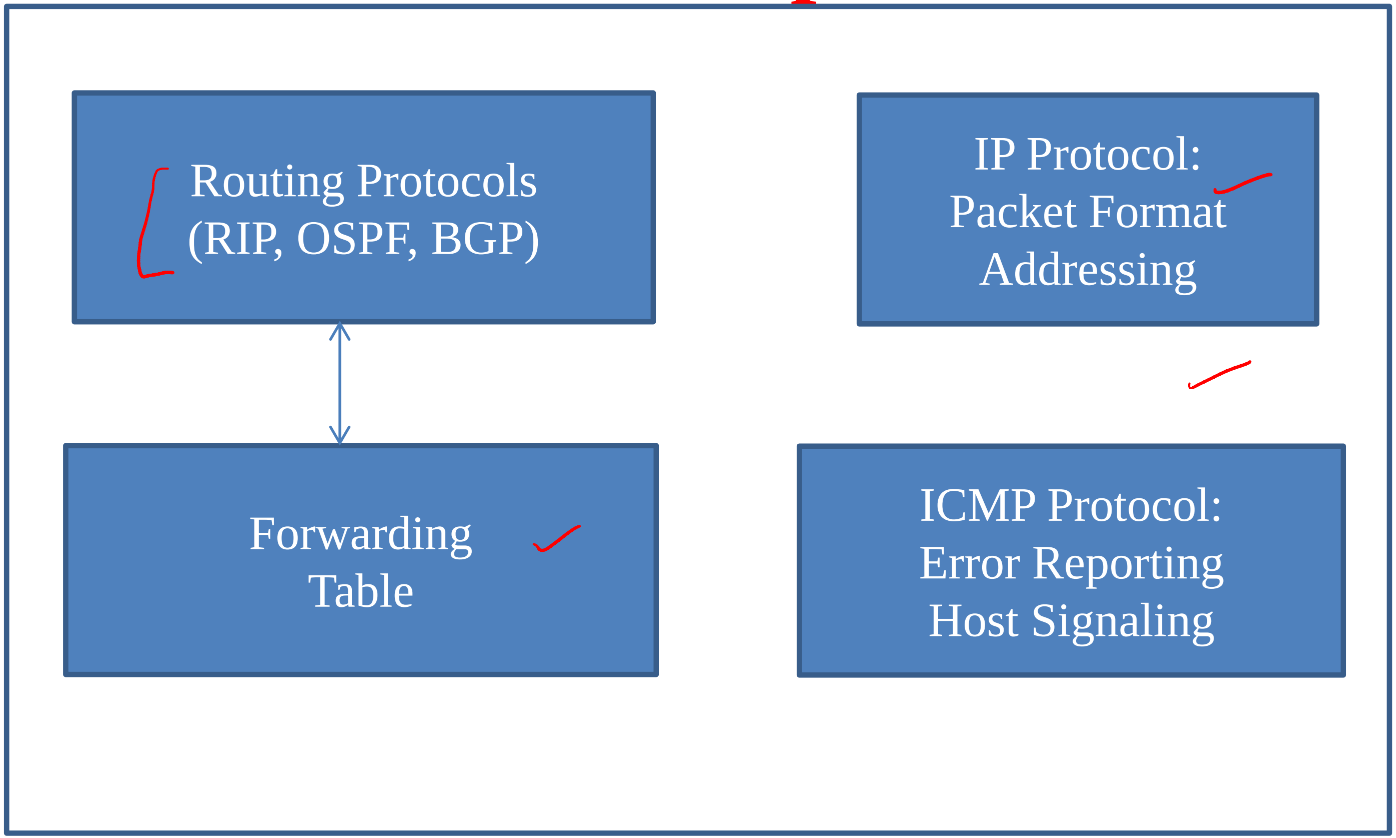
Routing -- Overview

Kameswari Chebrolu

Service Model Implementation



Background

- Role of network layer is ‘end-host delivery’

  src dst

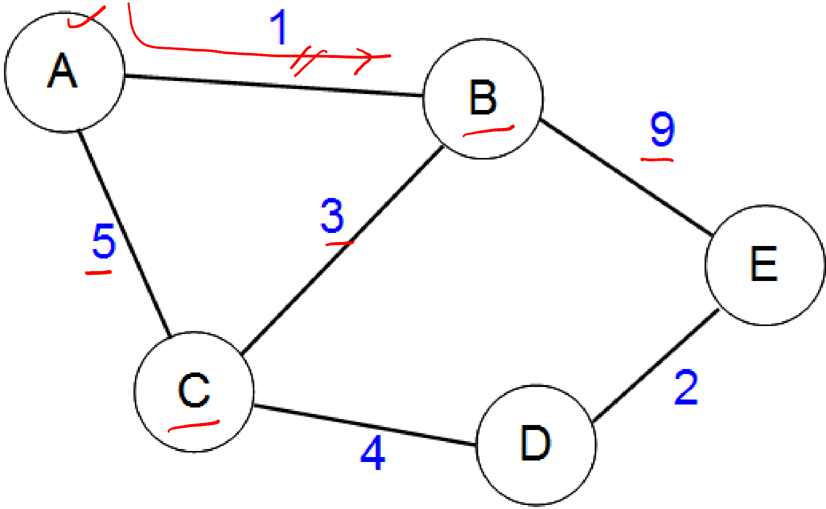
  - We looked at how packets are forwarded
- How are forwarding tables built? Via Routing Protocols

  dest

- Routing vs Forwarding
- Routing domain: All routers under same administrative control
  - E.g. University network, ISP network

Theory

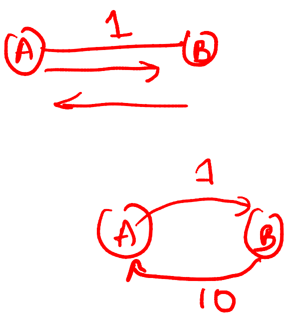
- Based on Graph theory
- Nodes: Hosts, **Routers**, Networks vertices
- Edges: Correspond to physical links
  - Edges associated with a cost
  - No edge → infinite cost
- Neighbor: Directly connected nodes

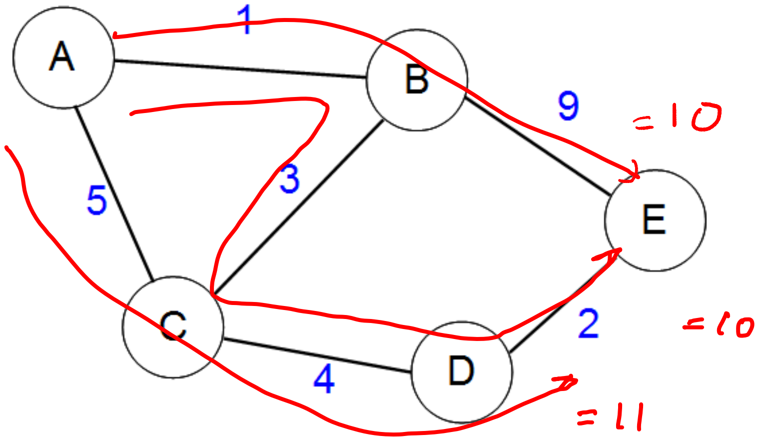


Goal of Routing

- 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

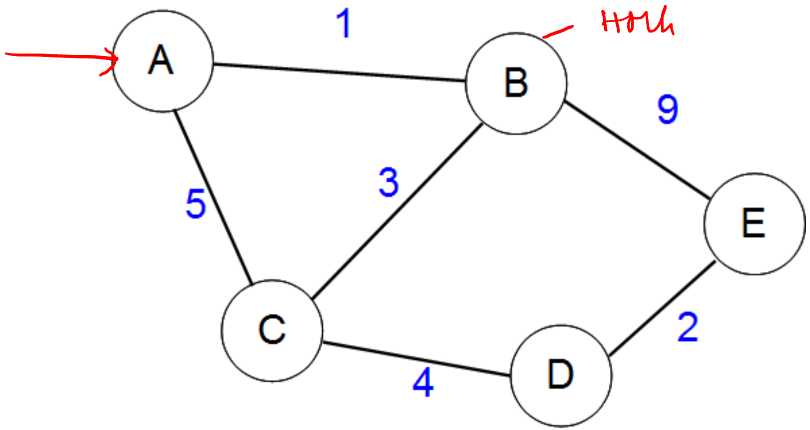
Nodes A, E





Goal of Routing

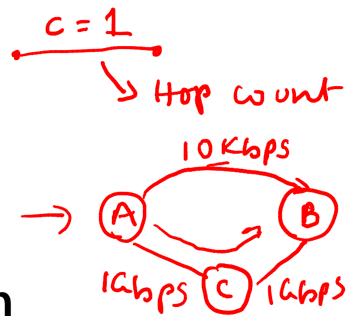
- Single-source 'shortest' path problem: Find least cost path from a source to all other nodes in the graph
- Refer to Dijkstra's algorithm



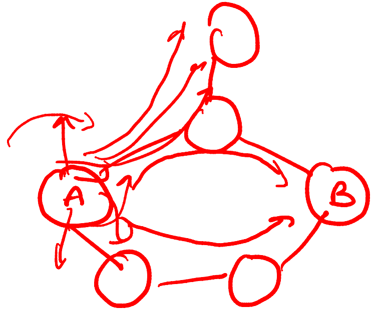
Desirable Features

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