

## Interplay Between Routing and Forwarding

How do I figure out what are the right routes through the network? Especially in large networks?

## Graph Abstraction

Represent the network as a graph where the nodes are routers and the edges are links between routers. We need to also assign a cost to different links (e.g. the latency or the congestion, or some complicated quantity).

In many networks, the cost for every link is set to 1, so essentially you're just looking at number of hops.

This is useful because we can apply graph algorithms (e.g. shortest-path algorithms) to our networks.

You can also use graphs for P2P groups, where nodes are peers and edges are TCP connections – graphs are used widely in networking.

## Routing Algorithm Classification

Global vs. decentralised

- may not have info about all the links (routers will need to share info with each other about their nearest neighbours)
- link state algorithms (global)
- distance vector algorithms (local)

Also need to consider static vs. dynamic

- static algorithms have slowly-changing routes
- dynamic algorithms have routes changing more quickly
  - periodic updates in response to link cost changes

## A Link-State Routing Algorithm

Uses Dijkstra's algorithm.

Every individual router runs the algorithm, taking the list of links and the costs as input. Every router is assumed to know the identities of all other routers on the network.

How do routers know about other routers? Every router periodically sends out a link state broadcast, describing its interfaces and identifying itself.

[...]

## Distance Vector Algorithm

Uses the Bellman-Ford Equation (dynamic programming).

Unlike dijkstra, each router doesn't know the full path to every other, just the path cost (having been told it by a neighbouring router), and reports these destinations and costs to neighbouring routers.

Take node A. Every neighbouring node reports which nodes it can get to and at what cost. A can get to all nodes reported, with the minimum cost that was reported for each (including the cost to the neighbour).

Important to note that each node is trusting the other nodes.

Also, every node shares its whole table with every neighbour. From time to time, each node sends its distance vector estimate to its neighbours. On receiving a new estimate, a node updates its own estimate using the new info. These sends can be triggered by updates, or can happen at fixed intervals.

## Weird Things

### Link Cost Changes

Everything's fine when link costs go down, but not as much when they go up.

You can get a routing loop, where two routers think their shortest path is through each other.

You can get around it somewhat by not reporting routes back to routers involved. (If A's shortest path to X is through B, it won't report that to B.)