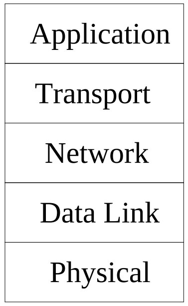
Layered Architecture



Network Layer

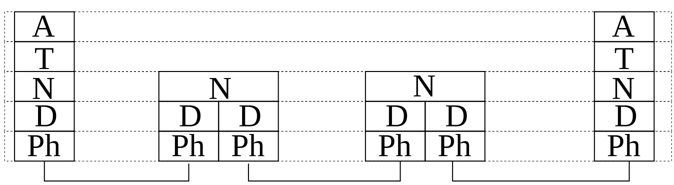
Service:

- Delivers data packets (segments) from the transport layer of the origin host to the transport layer of the destination host.

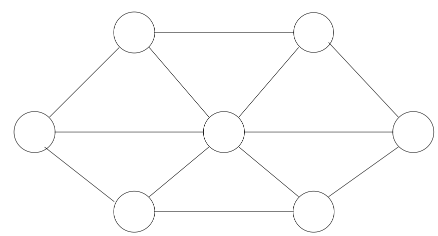
Functionality

- Path Determination (Routing)
- Switching
- Addressing

Layered Architecture



Network Layer



- Path Determination (Routing)
- Switching

Network Layer

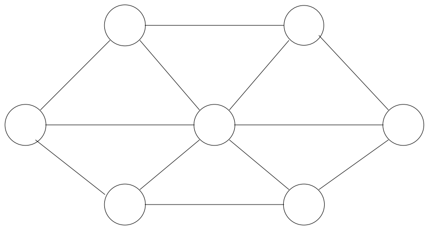
Complexity

- Coordination between peer processes of all nodes
- Error handling (link/node failure)
- Adaptation to changes in the traffic load

Outline

- Introduction
  - Routing
- Shortest Path Routing
  - Bellman-Ford algorithm
  - Dijkstra's algorithm
- Internet Nework Layer

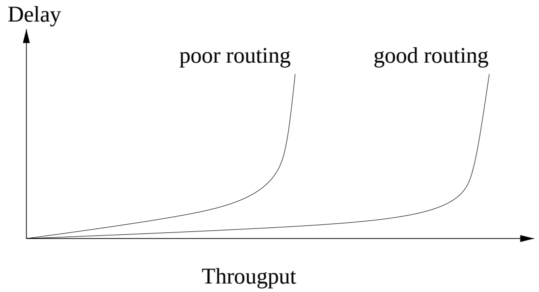
Routing Algorithms



Routing Algorithms

Performance Measure

- Relation Throughput - Average Delay



Routing Algorithms

Routing Decision:

- Datagram service: for each packet
- Virtual Circuit service: for each session

What is the advantage of virtual circuit service?

Virtual Circuit Service

Connection Oriented

- Virtual Circuit setup
- Data Transfer
- Virtual Circuit teardown

Virtual Circuit Service

Call Admission Control

- Rate Allocation (and Policing)
- QoS guarantees

Datagram Service

Connectionless

- no Call Admission Control
- no QoS guarantees ("best effort service")

$->$ dynamic transmission rate adaptation (TCP)

Routing Algorithms: Classification

Centralized - Distributed

- Centralized: routing decision are made at a central node
- Distributed: computation of routes is shared among the network nodes

Static - Dynamic

- Static: routes are fixed regardless of traffic conditions
- Dynamic: routes are changed in response to changes in traffic conditions (congestion)

Routing Algorithms: Flooding

Mechanism

- Each node forwards packets to all its neighbors

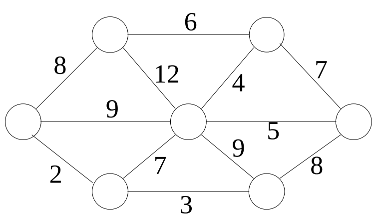
Rules

- do not send packet to node from where it was obtained
- send the same packet to a neighbor at most once

Why use Flooding?

- broadcast (topology) information
- when network topology changes frequently

Routing Algorithms: Shortest Path Routing



Cost

-
-
-

Routing Algorithms: Shortest Path Routing

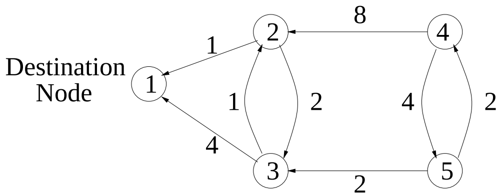
Link costs will change with time and so may shortest path. Therefore, we need a way to efficiently compute the shortest paths.

Two main approaches

- Bellman-Ford algorithm (Distance Vector Routing)
- Dijkstra's algorithm (Link State Routing)

Bellman-Ford Algorithm

Find shortest path from all nodes to a destination node.



$$D_i = \min_j \{d_{ij} + D_j\}$$

D_i: shortest distance from node i to the destination node 1

d_{ij}: cost of link from node i to node j

Bellman-Ford algorithm

- Node 1 is the destination node
- $d_{ij} = \infty$, when no link from i to j
- $D_1^k = 0$, $k = 0, 1, 2, ...$

Initialization

- $D_i^0 = \infty$, for all $i \neq 1$

Update

- $D_i^{k+1} = \min_j \{d_{ij} + D_j^k\}$, $k = 0, 1, 2, ...$

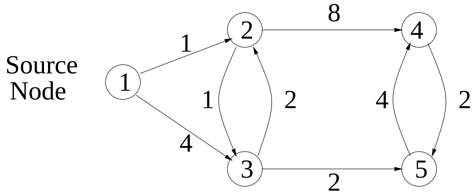
Distance Vector Routing

- Distributed algorithm. Each nodes has local information.
- Each node maintains a table (distance vector) of its (best known) shortest distance to each destination node, as well as which link to get there.
- Each node knows the link costs to all its neighbors
- Nodes exchange their distance vector with its neighbors
- Nodes update their distance vector based on these exchanges

Bellman-Ford algorithm is used to update distance vector.

Dijkstra's algorithm

Find shortest form a source node to all other nodes.



Idea

- Find the closest node
- Find the second closest node
- etc.

Dijkstra's algorithm

- Node 1 is the source node
- $d_{ij} = \infty$, when no link from i to j
- D_i: estimate of distance from the source node 1 to node i
- P: set of "permanently labeled" nodes

Initialization

- $P = \{1\}$
- $D_1 = 0$
- $D_i = d_{i1}$, for all $i \neq 1$

Update

- Step 1: (Find closest node) Find $i \notin P$ such that

  $$D_i = \min_{j \notin P} D_j$$

  Set $P = P \cup \{i\}$. If P contains all nodes, then stop.

- Step 2: (Update distance estimates) For all $j \notin P$ set

  $$D_j = \min\{D_j, D_i + d_{ij}\}$$

  Go to Step 1.

Link State Routing

- Distributed algorithm. Each nodes needs global information.
- Each node discovers its neighbors and the corresponding link costs.
- Each node broadcasts this information to all other nodes.
- Using this global information, each node computes the shortest path to all other nodes.

Dijkstra's algorithm is used to compute the shortest path.

Routing Algorithms

Distance Vector Routing

- Distributed algorithm
- Each node uses local information
- Suffers from count-to-infinity problem
- Can oscillate

Link State Routing

- Distributed algorithm
- Each node needs global information
- Can oscillate