Routing

COMP312

Richard Nelson
`richardn@cs.waikato.ac.nz`

`http://www.cs.waikato.ac.nz`

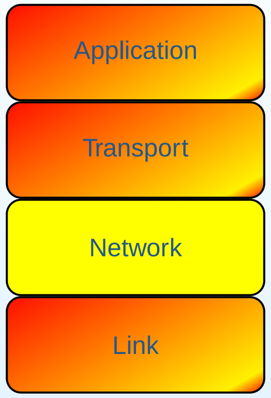
Department of Computer Science

University of Waikato

Lecture Outline

- Routing Concepts.
- Simple Routing Schemes.
- Dijkstra’s Algorithm
- Bellman-Ford Algorithm.

Internet Protocol Model



Working Here

Routing Concepts

- Routing
- Graphs
- Optimisation

Review: Forwarding

Forwarding is the process a node performs when it recieves a packet on one link and transmits it on another.

- In order to choose the correct next hop, the forwarding process consults a *routing table*.
- Routing is the process of building the routing table so that the next hop is chosen correctly.
- A node that forwards packets on the internet is commonly called a router.

Routing

Routing needs to produce paths through a network from sources to destinations. Normally, this is from any node to any other.

- Requires knowledge of the topology of the network.
- Each router makes independent forwarding decisions so their view of the network must be consistant.
- Must account for Topology changes through link and node failures, additions and upgrades.
- Also traffic demands need to be balanced with capacity availabilty.

Optimal Routes

Choosing the “best” route is important, but there can be many definitions of best.

- Lowest delay and/or jitter
- Least number of hops
- Highest bandwidth and/or available capacity

Often routing is done on hop count as minimising this reduces the total network utilisation. More sophisticated routing protocols allow the use of arbitrary metrics and leave the setting of per link metrics up to network operators so they can choose which parameters are important in their network.

Centralised vs Decentralised

Centralised.

- Network information is gathered at a network control centre which processes it, chooses paths and downloads the routing information to switch nodes.
- Relies on the network for collection of information and route dissemination.
- Can choose very efficient routes.
- May be slow or unreliable if network is overloaded.

Decentralised

- Network information is passed amoung switches which then make their own routing decision.
- Routing may be less than optimal during periods when new information is being distributed.
- No central point of failure.
- Individual switches find out about changes in their local environment very quickly.

Offline vs Online

Offline.

- Routes are pre-computed and downloaded to the network nodes.
- Backup routes are also required for failures.
- New routes are calculated at a time of the network managers choosing.
- Generally associated with Centralised routing.

Online

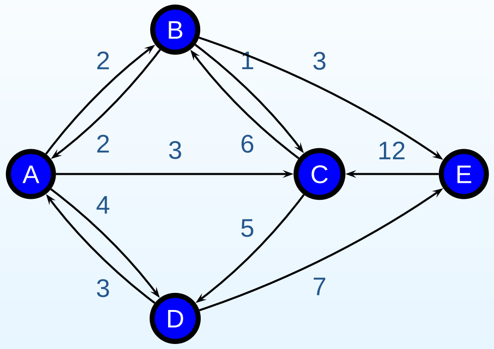
- Route information is recalculated in response to any network changes.
- Has more flexible response to failures, but response is lengthened by recalculation time.
- Generally associated with decentralised routing schemes.

Graphs

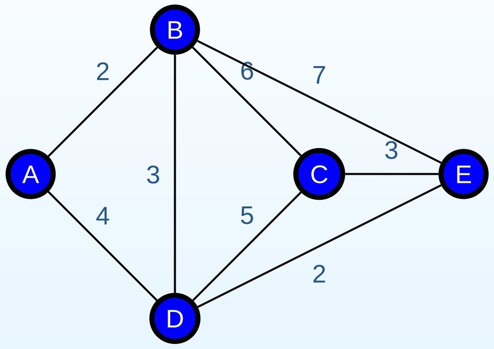
Route finding is part of Graph Theory.

- Graphs consist of *Vertices* (nodes) and *Edges* (links).
- Edges may be directed and may be weighted.
- A bidirectional link is represented by two directed edges.
- The weight or cost of a link may represent
  - The bandwidth
  - The delay.
  - The financial cost
  - etc
- Most networks use bidirectional links with symmetrical costs.

Directed Graph



Simplified Network Graph



Optimisation

Generally routing is designed to optimise use of the network in some way.

- Delay.
- Link Utilisation
- Cost
- Path Length

Generally the optimisation aim will affect the choice of link metric.

Circuit Switched Networks

- Utilisation on circuit switched trunks can be 100% without affecting individual circuit quality.
- The routing tries to provide many different paths between each source and destination switch so that if one trunk is full a different path can be used.
- This is called *alternate* routing.

Circuit Switched Networks

- Generally a separate signalling network is available.
- Circuit holding times are often long compared to end to end transmission times.
- Circuit connection patterns are often stable and predictable.
- Circuit switched networks generally use centralised routing functions.
- Paths may be changed at different times of the day to match traffic demands.

Packet Switched Networks

- Want to minimise delay and maximise available bandwidth.
- Delay depends on the traffic patterns.
- Traffic demands may change extremely quickly.
- Routing on traffic demands may destabilise the network.
- Normally aim to minimise utilisation by using bandwidth based metric.

Shortest Path Routing

- Define link metrics
- Find paths such that they are as short as possible with respect to the chosen metrics.
- Minimising hop count minimises the traffic carried by the network.
- Algorithms to calculate shortest paths are important.

Routing Concepts - Summary

- Routing is the path finding function of the network
- Routing is based on graph theory.
- Route optimisation uses shortest path calculation based on metrics related to the parameter to be optimised.
- Circuit switched networks often route based on traffic demands.
- Packet networks generally do not due to rapid traffic fluctuations.

Return to ToC

Simple Routing Schemes

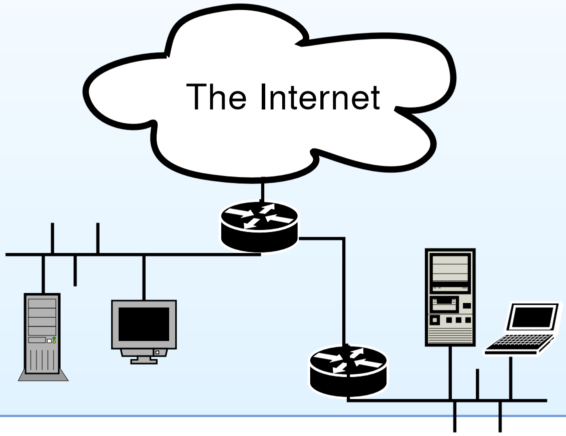
- Fixed Routing
- Flooding and Random Routing
- Hierarchical Routing

Fixed Routing

- Routes are manually configured into switches.
- Routes are worked out by the network manager.
- Very stable, predictable routing.
- No capacity to deal with topology changes without manual updates.

Static Routing

- Fixed routes entered into IP routers are called *static* routes.
- Very commonly used where there are no alternate links.
- Static routing is easy to manage for very simple topologies.



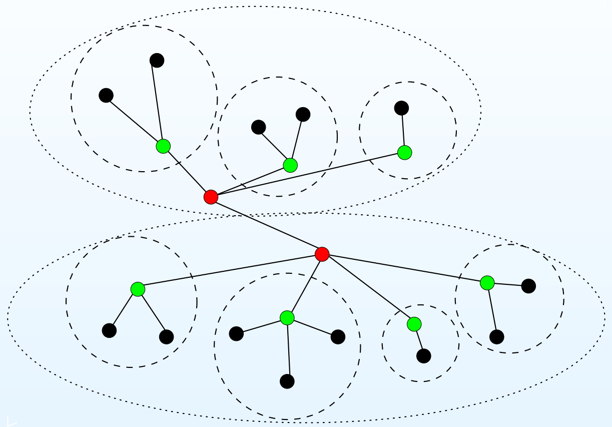
Flooding and Random Routing

- Simple techniques that require no network information.
- Very reliable
- Applicable to highly dynamic situations where up-to date network information is unavailable (e.g. ad-hoc networks, battlefield situations)
- Flooding always finds the shortest route (it tests every route) but generates huge extra traffic
- Random routing generates less extra traffic than flooding but routes are longer.
- Both techniques can be improved somewhat by not trying routes that have already been used.

Hierarchical Routing

- Switches/routers are divided into groups.
- Each switch knows about the topology within its group.
- A group-leader switch is used to connect to group-leader switches in other groups.
- The group-leader switches know the topology of all the group-leader switches in the super group.
- There may be more than two levels in the hierarchy with super-group-leaders being interconnected at the super-super-group level and super-super-group leaders being interconnected etc...

Topological Hierarchy



Logical Hierarchy

Hierarchical Routing

- Minimises the routing information each node requires.
- Causes small increases in path lengths.
- Has been used extensively in telephone networks.
- Some IP routing protocols have limited hierarchy.

Simple Routing Schemes - Summary

- Fixed routing relies on manual route calculation. It is commonly used in simple network topologies.
- Flooding and Random routing are inefficient but require no knowledge of the network topology. They may be useful in specialised situations.
- Hierarchical routing can limit the size of routing tables.

Return to ToC

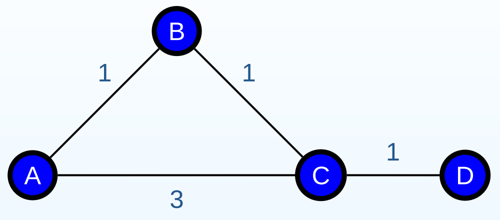
Dijkstra’s Algorithm

- Also known as the shortest path algorithm.
- Relies on having a full knowledge of all nodes and links in the network.
- Calculates shortest path from one node to all others in the network.
- If multiple paths are equally short they can all be found.
- The algorithm terminates once all the shortest paths are found and must be re-run if new node/link information is received.

Dijkstra's Algorithm (simple definition)

1. Start with a set consisting of the single source node.
2. Calculate cost of paths to all node directly attached to the set.
3. Find lowest cost path to a node not already in the set
4. Add that node to the set
5. If there are any nodes left not in the set go to step 2.

Dijkstra's Algorithm



i	Set	B	C	D

Dijkstra's Algorithm - Summary

- Requires full knowledge of all links and nodes.
- Calculates shortest paths for one source.
- Terminates once paths found and must be re-run when the topology changes.

Return to ToC

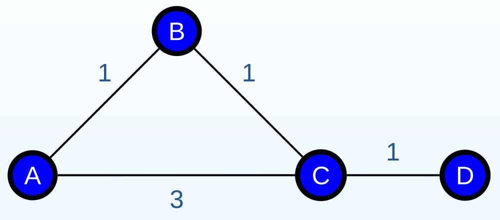
The Bellman-Ford Algorithm

- This is a distributed algorithm that calculates the shortest paths to each node.
- Each node maintains a list of known distances to other nodes with first hop information.
- Full path information is not exchanged.
- The algorithm follows the stages:
  - Finding all one hop shortest paths.
  - Finding all two hop shortest paths.
  - Finding all three hop shortest paths.
  - etc
- The algorithm continuously updates as new information arrives.

The Bellman-Ford Algorithm

1. Each node maintains a list of known destinations with the distance to each destination, and the first hop.
2. Each node starts by adding itself to the list.
3. Nodes exchange their set of known destinations and the distances with neighbours.
4. Each node checks their neighbours list updating distances with the link cost to that neighbour.
   - New destinations are added to the list with that neighbour as first hop
   - Previously known destinations are checked to see if the neighbour offers a shorter path.
5. Goto step 3.

Bellman-Ford Algorithm

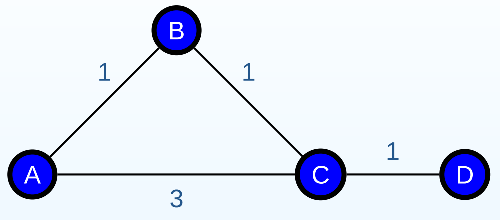


i	A	B	C	D

Bellman-Ford Algorithm - Topology changes

- Standard Bellman-Ford calculates shortest route.
- If a new, better route becomes available, the information will spread through neighbour exchanges and nodes will start using it.
- If a route becomes invalid, the information is not advertised.
- To solve this invalidation timers are used.
- Routes become invalid if no advertisement is received for some time.

Counting to Infinity



i	A	B	C	D

Solving Bellman-Ford Problems

- Infinity:
  - Infinity is normally set low (e.g. 16)
  - When a destination's distance reaches infinity limit it is set to unreachable.
  - Limits network size.
- Split Horizon: Don't advertise destinations back to nodes from which the distance information was received.
- Poison reverse: When a neighbour node becomes unreachable, advertise its distance as infinity.

Bellman-Ford Algorithm - Summary

- Distributed computation of best next hops for all nodes.
- Routes calculated for one hop paths then two hop then . . . .
- Computationally more efficient than Dijkstra due.
- Has practical problems, particularly in dealing with failures.

Return to ToC