Internet Engineering

241-461

Robert Elz

kre@munnari.OZ.AU
kre@coe.psu.ac.th

http://fivedots.coe.psu.ac.th/~kre

Network Routing

Network Routing

- Kurose & Ross: Computer Networking
  - Chapter 4: 4.2, 4.3 & 4.5

James F. Kurose & Keith W. Ross
Computer Networking
A Top-Down Approach Featuring the Internet

Contents

Network Routing

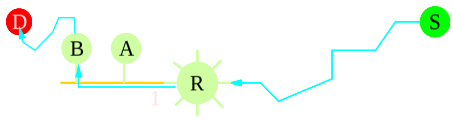
- Have data to send
  - A packet
- Want to get data to destination
  - Know its address
  - Address is in packet
- How do we get packet to destination?
  - Obviously we send over network!
- How does network know

where packet should go?

FORWARDING & ROUTING

Forwarding

- Packet arrives at a router
  - Must be transmitted towards destination
  - Send it where ?
- Router has Forwarding Table
  - List of Destinations
    - Which interface to use
    - Where to go next



to	interface	next-router
D	1	B
X	3	Q

- Hop-by-Hop Routing

Routing

- Forwarding Table is needed
  - Routing builds it.
- Routing styles
  - Global
  - Distributed
  - Static
  - Dynamic
  - Chaotic
- Hot Potato Routing
  - No forwarding table
  - Send packet anywhere
    - Pick random destination
  - Hope packet eventually arrives
  - Not very useful!

Contents

Global Routing

- Centralised Calculation
  - Paths from all sources
  - To all destinations
- Distributed to routers
  - As Forwarding tables
  - First segment of path
    - From this router
    - To each destination
- Can be
  - Static
  - Dynamic

Static Routing

- Paths are calculated once
  - Often by a person (by hand)
- Then fixed
  - Do not change over time
  - Or only when recalculated
    - Infrequently
  - Do not react to network changes
    - broken links
    - crashed routers
    - congestion
- Acceptable where
  - No alternative paths exist
  - No alternative path wanted
  - Network outage is OK

Dynamic Routing

- Paths re-calculated as required
  - Whenever network changes
  - Whenever something important changes
  - Network link down
    - or up
  - Router down
    - or up
  - Usually not congestion
    - Changes too rapidly
- Updated forwarding tables installed
  - Soon after each re-calculation

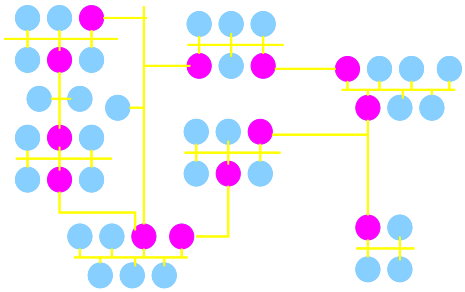
Distributed Routing

- Each router calculates
  - Builds its own forwarding tables
- Can be
  - static
    - installed by operator
  - dynamic
    - reacting to network conditions
- IMPORTANT
  - All routers must calculate consistent paths
  - Why?

Contents

Routing Calculations

- What must be done?
- Let's look at a network.



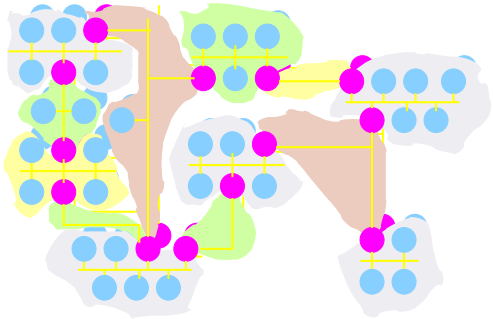
- Many nodes
  - And links connecting them
- The nodes that connect two links
  - Routers (usually)

Routing Calculations (2)

- Want to find paths through network
  - From any node to any other
- First, simplify things a little
  - Note that all nodes on a net have similar addresses
  - Just like houses in the same street
- To find path to a node
  - Find path to its network
  - Then node is there somewhere
    - Link layer can find it.

Routing Calculations (3)

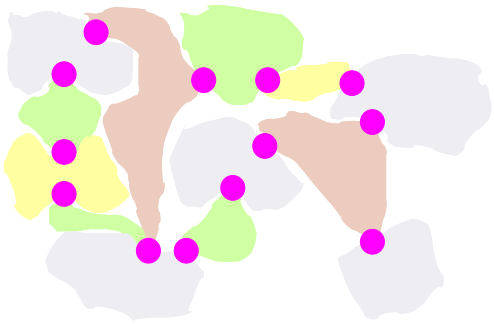
- Take our network



- And find the networks
  - Shown as coloured backgrounds
    - different colours mean nothing
- Note that routers
  - are connected to multiple networks

Routing Calculations (4)

- End nodes are irrelevant
  - We only need networks



- Networks often called clouds
  - When viewed this way
    - because of appearance
    - and they hide what is inside them
  - We need connections between networks
    - Bring back the routers

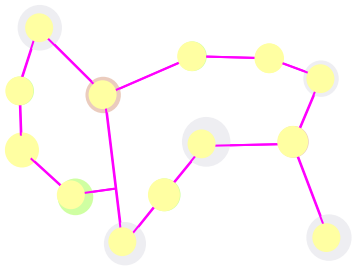
Alternative Views

- Shape of clouds not important so...

- Can be represented differently
  - Same network
    - Different drawing
- And differently again

Taken Further

- The same net:

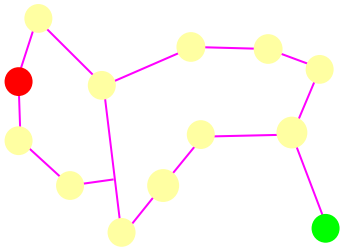


- End result
  - Nodes represent Networks
  - Arcs represent Routers
- It makes no difference!
  - Sometimes it all gets mixed up

Contents

The Routing Problem

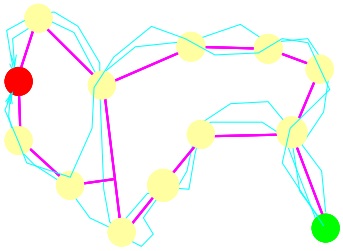
- To find a path
  - from one node
  - to another node



- Doesn't matter which nodes
  - All paths are needed eventually
  - We just pick one to start
    - From one node to another

The Routing Problem (2)

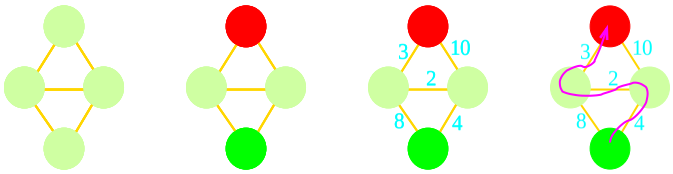
- There are often many paths
  - possible paths
- If only one
  - not an interesting problem



- Which path should we pick?
  - And how do we find it?

Graph Traversal

- Mathematics Problem
  - Find path through a graph
  - Satisfying some constraints



- We have a graph
  - Nodes and Arcs
  - What they represent does not matter
- Graph has start node and end node
  - We want to find a path
  - The constraint?
- Arcs have costs
  - Aim: minimise path cost