Routing

CSC 343·643

Wake Forest University
Department of Computer Science

Fall 2013

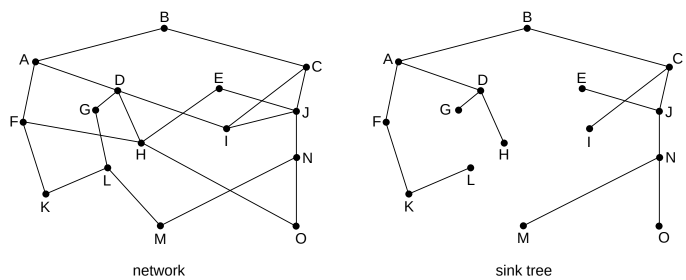
Routing Algorithms

- A primary function of the network layer is routing packets
- Routing algorithm is responsible for deciding which output line incoming packets should be transmitted
- If a datagram network
  - Routing decision is *new* for every packet
- If a VC network
  - Routing decision made for a VC request
  - Thereafter, packets follow established path
- Routing algorithm goals include: correctness, simplicity, robustness, stability, fairness, and optimality

What is Optimal

General statement about optimal routes

- Optimality principle - If router j is on the optimal path from router i to router k, then the optimal path from j to k falls on this route
- As a result the set of optimal routes from all destinations to a source is in the form of a *sink tree*



- The sink tree in not necessarily unique, but a tree never has...

Routing Algorithm Classifications

- Performance criterion
  - Number of hops, cost, delay, ...
- Decision time
  - Packet or session
- Decision place
  - Each node or central node
- Strategy
  - Nonadaptive (static routing) - Routes calculated off-line in advance
  - Adaptive (dynamic routing) - Routes calculated based on measurements or topology (*that change*)

Routing Methods

- Four different algorithms
  - Dijkstra's
  - Bellman-Ford
  - Flooding
  - Random
- For each method consider
  - Centralized or distributed
  - Ability to handle network dynamics
  - Amount of overhead required

Dijkstra's Algorithm

Summary: *Find the shortest paths from a given source node to all other nodes by developing the paths in order of increasing path length. Algorithm proceeds in stages, by the k stage shortest paths to the k nodes closest (least cost) to the source node have been determined*

- For the algorithm, let
  N = set of nodes in the network
  s = source node
  M = set of nodes incorporated by the algorithm
  $d_{i,j}$ = link cost from $i \rightarrow j$
  D_n = cost of least cost path $s \rightarrow n$ (direct paths have a value, otherwise ∞)

1. $M = \{s\}$
   - $D_n = d_{s,n} \quad \forall n \neq s$ (directly connected nodes)
2. Get next *cheapest* node and incorporate into set M

- Find neighboring node not in M that has least cost path from s and add to M (also incorporate edge)

$$\text{Find} \quad w \notin M \quad \text{such that} \quad D_w = \min_{\forall j \notin M} \{D_j\}$$

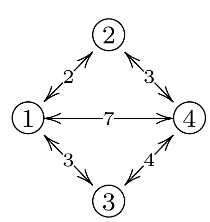
- Add w to M, $M = M \cup w$

3. Update least cost paths using w

$$D_n = \min\{D_n, D_w + d_{w,n}\} \quad \forall n \notin M$$

4. Go to step 2 and repeat until all nodes incorporated in M

Dijkstra Example



Iter.	M	D_2 Path	D_3 Path	D_4 Path
1				
2				
3				
4				

Bellman-Ford Algorithm

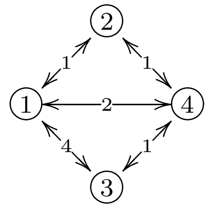
Summary: *Find the shortest paths from a given source node subject to the constraint that the paths contain at most one link; then find the shortest paths with a constraint of paths that contain at most two links, and so on …*

- For the algorithm, let
  s = source node
  $d_{i,j}$ = link cost from $i \rightarrow j$, $d_{i,i} = 0$, $d_{i,j} = \infty$ if not directly connected, $d_{i,j} > 0$ otherwise
  h = maximum number of hops allowed in any path
  D_n^h = cost of least cost path $s \rightarrow n$ using h hops

1. Initialization
   - $D_n^0 = \infty \quad \forall n \neq s$
   - $D_s^h = 0 \quad \forall h$
2. Update

   for each successive $h \geq 0$
   for each $n \neq s$
   $D_n^{h+1} = \min_{\forall j} \{D_j^h + d_{j,n}\}$

Bellman-Ford Example

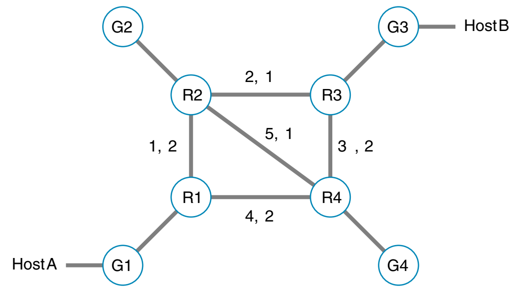


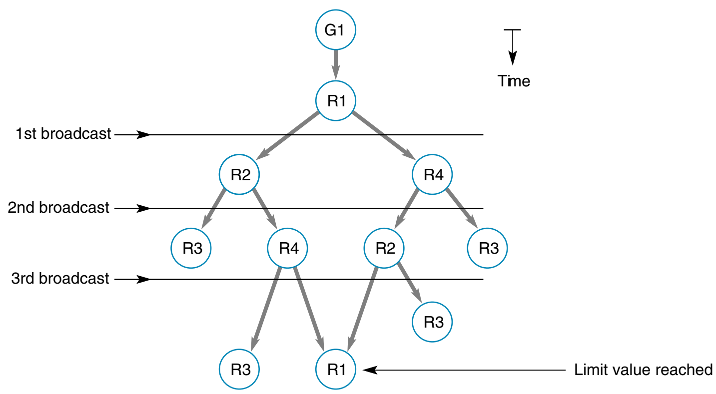
h	D_2^h	Path	D_3^h	Path	D_4^h	Path
1						
2						

Flooding

- Simple technique requires **no** network information
  - Source node sends to each neighbor
  - Neighbor transmits on every link accept arriving link
- Avoiding an *infinite* number of copies
  1. Nodes remember identity of packets, don't transmit duplicates
  2. Place *hop-counter* inside each packet
     - Node passes packet decrements counter by 1
     - If count is zero discard packet

     What should the hop count equal?
- Flooding properties
  - Robust, since all paths tried, at least one copies will arrive
  - One copy will arrive using the minimum hope route
  - All nodes are visited





Random Routing

- Random routing has the simplicity and robustness of flooding, but requires far less resources
- Algorithm details
  - Node **randomly** selects outgoing link for an arriving packet (exclude the link packet arrived on)
  - If links equally probable, then utilization of links is round-robin
- Performance modification for links with different rates
  - Assign a probability to each link based on its data rate

$$p_i = \frac{r_i}{\sum_j r_j}$$

  - Yields better utilization of links

What are the advantages and disadvantages?

Adaptive Routing

- Adaptive routing is used in packet switched networks
  - Routes will change based on network conditions
- **Network state** information needed by all nodes, as a result
  - Routing decisions are more complex
  - Must trade-off overhead and *freshness*
  - *Classic control problem*
- Control problem
  - Changing too quickly will cause oscillations
  - Changing too slowly never converging
- Two adaptive routing methods
  - **Distance vector** and **link state**

Distance Vector

- Each node (router) maintains a table (vector), indicates
  - Best known distance to a destination
  - Next link (hop) to take to get there
- Tables updated by exchanging info with neighbors
  - A form of *distributed Bellman-Ford*
- Each node i maintains two vectors

$$\text{delay vector} = D_i = \begin{vmatrix} d_{i,1} \\ d_{i,2} \\ \vdots \\ d_{i,n} \end{vmatrix} \quad \text{next node vector} = S_i = \begin{vmatrix} s_{i,1} \\ s_{i,2} \\ \vdots \\ s_{i,n} \end{vmatrix}$$

$d_{i,j}$ = current minimum delay from $i \rightarrow j$

$s_{i,j}$ = next node in the current route from $i \rightarrow j$

- Periodically nodes exchanges D_i with neighbors
  - Each node updates the vectors using the new information
- At node k, vectors are updated using the equations

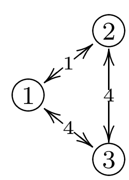
$$d_{k,j} = \min_{i \in A} \{l_{k,i} + d_{i,j}\}$$

$$s_{k,j} = i \quad \text{using } i \text{ from preceding equation}$$

$A =$ set of neighbors of k
$l_{k,i} =$ current delay from $k \to i$ (sent by node i)

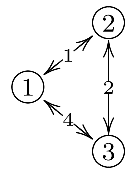
Distance Vector Example

- Given the following network, consider the vector for node 1



$$k = 1, \quad A = \{2, 3\}, \quad D_1 = \begin{vmatrix} d_{1,1} = 0 \\ d_{1,2} = 1 \\ d_{1,3} = 4 \end{vmatrix} \quad S_1 = \begin{vmatrix} s_{1,1} = - \\ s_{1,2} = 2 \\ s_{1,3} = 3 \end{vmatrix}$$

- Suppose the $2 \to 3$ link changes to 2



$$D_2 = \begin{vmatrix} d_{2,1} = 1 \\ d_{2,2} = 0 \\ d_{2,3} = 2 \end{vmatrix} \quad D_3 = \begin{vmatrix} d_{3,1} = 4 \\ d_{3,2} = 2 \\ d_{3,3} = 0 \end{vmatrix}$$

- Above two vectors are sent to node 1, which calculates routes

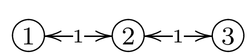
$$d_{1,j} = \min_{i \in \{2,3\}} \{l_{1,i} + d_{i,j}\} \quad j = \text{all other nodes}$$

$$j = 2 \quad d_{1,2} = \min\{l_{1,2} + d_{2,2}, l_{1,3} + d_{3,2}\} \Rightarrow s_{1,2} = 2$$

$$j = 3 \quad d_{1,3} = \min\{l_{1,2} + d_{2,3}, l_{1,3} + d_{3,3}\} \Rightarrow s_{1,3} = 2$$

Count to Infinity Problem

- Distance vector routing
  - Given n is the longest path in the network, with n exchanges all nodes will know about any *good news*
  - Bad news will take *infinite* exchanges
- Consider the following full-duplex network



  - Initially $1 \rightarrow 2$ is 1 unit and $3 \rightarrow 1$ is 2 units
- Assume node 1 goes down... *what happens on the exchanges?*
  1. Node 2 gets nothing from node 1 therefore cost is ∞
     - However, cost to node 1 via node 3 is 2, so node 2 uses path via node 3 to get to 1, cost is 3

2. Node 3 gets update cost from node 1, path to node 1 is 3, must update its cost to 4
3. Node 2 get updated cost from node 3, path to node 1 is 4, must update its cost to 5

   This repeats until cost is ∞

- Several changes to the DV routing have been proposed
  - **Split horizon** and **reverse poison** are two examples where minimum distance information is not sent to a neighbor if the neighbor is on the minimum path
  - Unfortunately these *fixes* do not always work

Link State

Has the following steps

1. Discover neighbors and learn addresses
2. Measure cost (delay) to neighbors
3. Create packet containing cost information
4. Send packet to **all** nodes (not just neighbors)
5. Compute shortest path to all other nodes (Dijkstra's Algorithm)

   Any problems with link state?

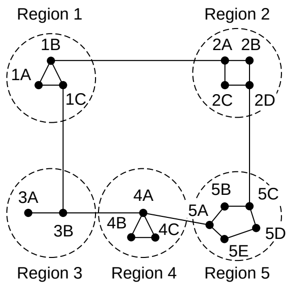
Routing Comparison

Characteristic	Algorithm: Distance Vector	Algorithm: Link State
Complexity	Messages sent to neighbors	Messages sent to all other nodes
Convergence	Time varies, count to infinity	May oscillate
Robustness	Routing tables depend on neighbor calculations (error propagates)	Calculations made on a per node basis

- *So which is used in IP?* ... **both**
  - Routing Information Protocol (RIP) is DV
  - Open Shortest Path First (OSPF) is LS
  - *However neither method is scalable...*

Hierarchical Routing

- As the network size increases, so do the routing tables
  - Infeasible for every router have an entry for every other router (called **source-based routing**)
  - The previous methods are not **scalable**
  - Hierarchical routing is used instead
- **Hierarchical routing**, divides the network into regions, called **Autonomous Systems** (AS)
  - Router can route to other routers inside its region
  - Router does not know how to route inside other regions
  - Traffic destined for another region forwarded by **gateway router**



Example network

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

non-hierarchical routing table

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

hierarchical table

- Example network is a two-level hierarchy
  - Source-based routing table has 17 entries
  - 2-level hierarchy has 7 entries

- Hierarchical tables
  - One entry for each router in same region (AS)
  - One entry for all routers in another region
    - All traffic from region 1 to region 2 traverse 1B-2A link

    Given a destination address in another region, how does a router know which region to route?
  - Addresses are *hierarchical*
    - In the example network, router addresses have the form *region-number* followed by *router-letter*
- Hierarchical routing disadvantages
  - May result in increased path lengths
  - For example, best route from 1A to 5C is via region 2, but hierarchical routing uses region 3

How Many Hierarchies

Consider a network with 720 routers

- No hierarchy, each router has 720 entries
- Partition into 24 regions each with 30 routers
  - This is a two-level hierarchy
  - Each routing table has _______ entries
- Using a three-level hierarchy, with 8 domains, each containing 9 regions of 10 routers
  - Each routing table
    - 10 entries for local routers
    - 8 entries for routing to other regions inside of own domain
    - 7 entries for other domains
  - Total of 25 entries

Static versus Dynamic IP Routing

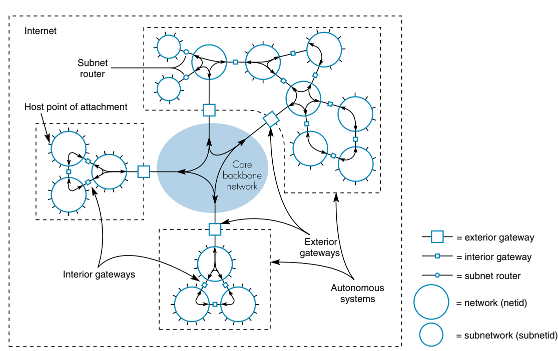
- Already know how IP packets are routed using routing tables
  - *How were the entries generated... statically or dynamically?*
- In static routing entries are manually adjusted
  - Acceptable for small networks
- For larger networks, dynamically change table entries
  - Routes should change based on network conditions
  - Allow routers to pass route information to one another
  - Use variations of Bellman-Ford and Dijkstra's
- N.B. This will **not** change the way IP datagrams are routed, just how/when the routing table contents change

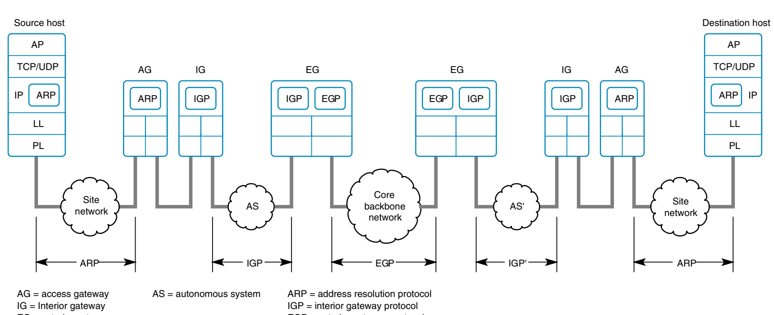
Internet and Autonomous Systems

- Internet is a collection of connected networks
  - Local, regional, national, and international ISPs
- Autonomous Systems (AS)
  - Collection of routers/hosts under an administration control
  - May consist of multiple networks

    Why is it important to divide into autonomous systems?

- Within this hierarchy classify routing algorithm as
  - **Intra-AS** route within one autonomous system
  - **Inter-AS** route among autonomous systems





Intra-AS Routing

- Used to configure and maintain routing tables within an AS
- Also called **Interior Gateway Protocols** (IGP)
- Historically three routing protocols have been used
  - Routing Information Protocol (RIP)
  - Open Shortest Path First (OSPF)
  - Enhanced Interior Gateway Routing Protcol (EIGRP)
    Cisco propriety

Routing Information Protocol

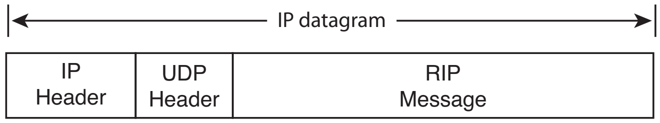
- RIP was one of the earliest intra-AS protocols
  - Still in use, popular since it was included BSD Unix
  - Two versions, original [RFC1058], and version 2 [RFC1723]
- Distance vector protocol
  - Neighboring routers exchange messages every 30 seconds
  - Message called **RIP response message** or **RIP advertisement**
  - In original version cost metric → hop-count

  Which shortest-path algorithm is it based-on?

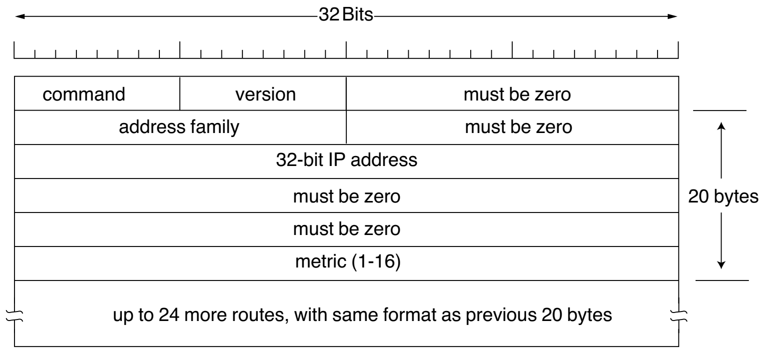
  What problem is associated with this algorithm?

RIP Messages

- Messages are carried in UDP datagrams (transport layer)



- RIP messages have the format



- RIP message fields
  - Command field, 1 = request or 2 = reply
  - Version field, 1 or 2
  - Next 20 bytes specify address family, an IP address, and an associated metric
- 25 routes can be advertised in a RIP message
  - Total size of RIP message is $20 \times 25 + 4 = 504$ bytes, less than 512 bytes
- *Why so much wasted space per entry?*

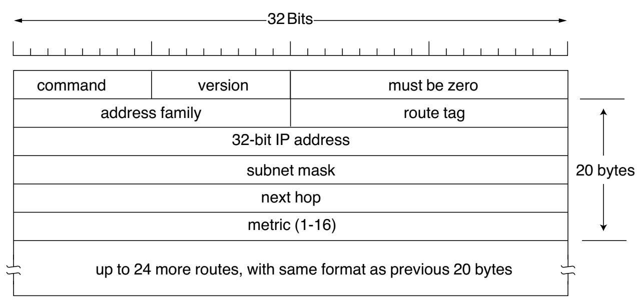
> *"The packet format is intended to allow RIP to carry routing information for several different protocols... None of the RIP implementations available to the author implement any other type of address... However, to allow for future development, implementations are required to skip entries that specify address families that are not supported by the implementation."* [RFC1058]

RIP Problems

- No knowledge of subnet addressing
- May take a long time to converge
  - Count to infinity problem
- Only uses hop count as metric
  - Other variables should be taken into account
  - Maximum of 15 limits size of network

RIP Version 2

- Protocol remains the same, but uses entire RIP message



- New message format
  - Routing domain - identifies the routing daemon
  - Route tag exists to support exterior gateway protcols
  - Mask applies to the corresponding IP address entry
- RIP-2 also provides a simple authentication scheme, to verify the sender of the routing table

Open Shortest Path First

- OSPF is another intra-AS routing protocol [RFC1247, 2328]
- Primary characteristics
  - Open protocol, specification is public domain
  - Link-state protocol, Dijkstra's algorithm used for SPF

    What is the disadvantage to using a link-state algorithm?
- OSPF can operate within a hierarchy
  - Largest entity is the AS
  - AS is divided into **area**
  - Routing algorithm will operate in each area

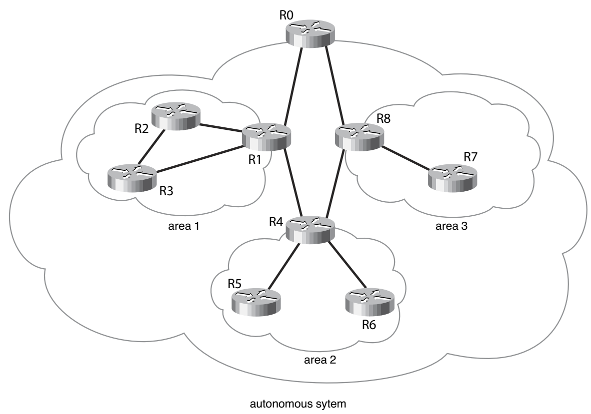
    What problem does this attempt to avoid?

OSPF Routing Hierarchy

- OSPF allows an AS to broken into **areas**
  - Areas do not overlap, but are not exhaustive

    Therefore we are reducing...
  - Outside an area, topology and details are not known
  - Area is *generalization* of a subnet
- Within an area each router must
  - Have the same link-state database and run the same SPA
  - Know the shortest path to any other router in its area
- Router classification
  - **Internal routers**, within an area
  - **Area border routers**, connect outside of area (within AS)
  - **AS boundary routers**, connect to other autonomous systems

- The diagram show all the different types of routers
  - Routers R2, R3, R5, R6, and R7 are internal routers
  - Routers ___________ are area border routers
  - Router R0 is the AS boundary router



- A **backbone** area also exists, consists of all area border routers, boundary routers, and any networks/routers not inside an area

OSPF Operation

- Determine neighbors by sending **hello** messages
  - Hello messages are multicast every 10 seconds
  - Failure to receive hello from neighbor for 40 seconds, assume link/neighbor failed
- Each router in an area distributes information about its local environment in datagrams called **link state advertisements**
  - Information about neighboring routers (links and costs)
  - LSA are distributed via **reliable flooding** (explicitly ACKed, sequenced, and time-stamped)
    - ∗ Sent if: new neighbor found, link failure, cost change, refresh every 30 minutes
  - Using this information, shortest tree path can be determined

OSPF Operation Review

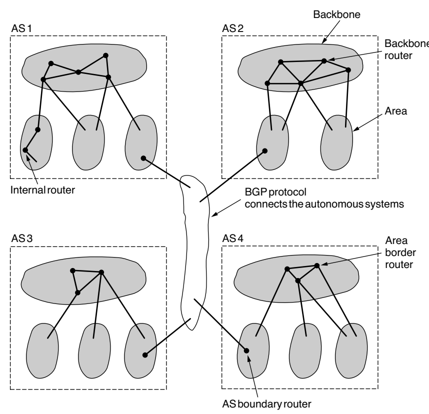
- Using reliable flooding, routers inform other routers in area
  - Each router can then construct a graph of the area and find the shortest paths
- Backbone area performs same procedure
  - Backbone also accepts information from area border routers
  - Can determine the best route from each backbone router to every other router
- Backbone information sent to area border routers, which advertise information within their area

OSPF Advanced Features

- Different routes based on traffic type of service
  - Allow different types of traffic to take different routes
    How can this be done?
- *Load balancing*
  - Using multiple routes may be better, than just one *best* route
- Security
  - Router exchanges are authenticated, prevents router spoofing

Exterior Gateway Routing Protocol

- Within a single AS, OSPF is generally used
- Routing between AS requires another routing protocol
  - Called an **exterior gateway protocol**
  - **Border Gateway Protocol** (BGP) [RFC1772]



- A different protocol is needed, since goals are different
  - Routing decisions are *policy-based*
  - Routing based on who controls AS rather than cost or delay
    - ∗ A telco accepts customer traffic, but not other telco traffic
- BGP routers classify traffic as
  - *Local traffic* originates or terminates in the AS
  - Otherwise is it *transit*
- BGP routers classify networks as
  - **Stub networks** have only one connection to the BGP graph (can not be used to transit traffic)
  - **Multiconnected networks** have multiple connections but refuse to transit traffic
  - **Transit networks** are willing to transmit third party traffic (backbone networks)

BGP Routing

- Path vector routing algorithm
  - BGP routers propagate path information (instead of cost)
  - Keeps track of the exact path used (not just the next hop)
  - The *count to infinity problem* is avoided since complete paths are exchanged
- BGP is the *de facto* standard for inter-AS routing Internet routing
  - Used to connect Network Access Points (NAPs)
  - Current BGP routing tables available at `http://bgp.potaroo.net/`

BGP Problems

- In 2008, YouTube was temporarily unavailable due to BGP entries
  - YouTube added the address block `208.65.152.0/22 (208.65.152.0 - 208.65.155.255)` in the Internet's routing tables
  - Pakistan Telecom advertised the `208.65.153.0/24 (208.65.153.0 - 208.65.153.255)` block
  - Traffic destined for YouTube when to Pakistan

    Why was this the case?

    Were the tables at the local hosts wrong?
- *A simple mistake at the top... or an attack...*

Unix Routing Dämons

- Routing daemon adds a *routing policy* to the system
  - Determines which routes to place in the routing table
  - For example, if multiple routes to a destination exists, the daemon selects the *best* route
- Unix systems run one of two routing daemons, `routed` or `gated`

	Interior Gateway Protocol		**Exterior Gateway Protocol**	
Daemon	**RIP**	**OSPF**	**EGP**	**BGP**
`routed`	V1			
`gated`, version 1	V1			V1
`gated`, version 2	V1,V2	V2		V2, V3

- Which one is being used on a Unix machine, enter the command `ps -ef | grep "ed"`