1. Basic setup
   - Directly connected
   - Not directly connected
   - Unix and Linux routing commands
   - Route selection

2. Mathematical representation

3. Routing in the Internet

4. Different routing mechanisms
Outline

1. Basic setup
   - Directly connected
   - Not directly connected
   - Unix and Linux routing commands
   - Route selection

2. Mathematical representation

3. Routing in the Internet

4. Different routing mechanisms
Outline

1 Basic setup
   - Directly connected
   - Not directly connected
   - Unix and Linux routing commands
   - Route selection

2 Mathematical representation

3 Routing in the Internet

4 Different routing mechanisms
Directly Connected IP Networks

- Configuration under Unix with `ifconfig`
- Packets can be delivered immediately by the data link (layer 2 software)
- No explicit route commands needed under Linux, because `ifconfig` sets a route automatically

**ifconfig command**

```
ifconfig <interface> <ip_address>
netmask <netmask> broadcast <broadcast_address>
```
Data link layer “routing”

- Only outgoing interface and layer 2 address is needed
- Packet is (selectively) flooded by layer 2
- Bridges and switches use a lookup table to map layer 2 addresses to outgoing interfaces
- Spanning tree protocol takes care of loops
Outline

1. Basic setup
   - Directly connected
   - Not directly connected
   - Unix and Linux routing commands
   - Route selection

2. Mathematical representation

3. Routing in the Internet

4. Different routing mechanisms
Global routing

- Not directly connected networks need a gateway (next hop)
- Recursive lookup
  - In practice this is only one level deep
  - Hence the gateway needs to be directly connected
  - In theory a deeper recursion level would be possible and useful
**Routing Table**

- **netstat -r** shows the routing table

### Route flags

<table>
<thead>
<tr>
<th>Flag</th>
<th>Set</th>
<th>Unset</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>route needs gateway</td>
<td>route is directly connected</td>
</tr>
<tr>
<td>H</td>
<td>route to host</td>
<td>route to network</td>
</tr>
<tr>
<td>S</td>
<td>route added statically (mostly by admin)</td>
<td>route added dynamically (by a protocol)</td>
</tr>
</tbody>
</table>
Outline

1. Basic setup
   - Directly connected
   - Not directly connected
   - Unix and Linux routing commands
   - Route selection

2. Mathematical representation

3. Routing in the Internet

4. Different routing mechanisms
route command

- Syntax varies between systems
  - Adding a (static) default route (Solaris)
    - `route add default <gateway>`
  - Adding a (static) default route (Linux)
    - `route add default gw <gateway>`
  - Adding a (static) host route (Linux)
    - `route add -host <host> gw <gateway>`
  - Adding a (static) network route (Linux)
    - `route add -net <network> netmask <netmask> gw <gateway>`
arp command

- Used to interact with the arp table
- Read the arp table
  - `arp` (without arguments)
- Delete an entry from the arp table
  - `arp -d <address>`
- Add an entry to the arp table
  - `arp -s <address> <hardware_address>`
ip command

- Introduced in the Linux iproute2 package
- General interface to kernel addressing and routing
- Replaces ifconfig, route and arp (almost) completely
- Has support for routing policies and multiple routing tables
## ip subcommands

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>link</td>
<td>layer 2 interface settings (MAC)</td>
</tr>
<tr>
<td>address</td>
<td>layer 3 interface settings (IP)</td>
</tr>
<tr>
<td>neighbor</td>
<td>arp cache data</td>
</tr>
<tr>
<td>route</td>
<td>routing table data</td>
</tr>
<tr>
<td>rule</td>
<td>routing table selection</td>
</tr>
</tbody>
</table>
Outline

1. Basic setup
   - Directly connected
   - Not directly connected
   - Unix and Linux routing commands
   - Route selection

2. Mathematical representation

3. Routing in the Internet

4. Different routing mechanisms
Route selection

- **Longest prefix match**
  - Host routes are preferred over network routes
- **Default route has shortest prefix**
  - Route of last resort
  - Cannot be used in the Internet core
Outline

1. Basic setup
   - Directly connected
   - Not directly connected
   - Unix and Linux routing commands
   - Route selection

2. Mathematical representation

3. Routing in the Internet

4. Different routing mechanisms
Mathematical representation

- Graphs (undirected, labeled)
  - Nodes are routers and hosts
  - Edges are (point to point) connections
  - Labels represent “cost” of the route
  - Undirected edges imply a restriction to
    - full duplex links
    - with the same cost in both directions
Broadcast networks
Node reduction

- LAN with $O(N)$ routers or hosts
- Full mesh of $O(N^2)$ edges
- Reduced with only 1 extra node to $O(N)$ edges
- Works for LAN’s, but also for NBMA networks
  - In many situations
  - But not always (take care)
Edge reduction does not always work

- Using a virtual network node is not always equivalent to a full mesh of point to point links
- Example of the introduction of loops
Edge reduction does not always work

- Using a virtual network node is not always equivalent to a full mesh of point to point links
- Example of the introduction of loops
Outline

1 Basic setup
   - Directly connected
   - Not directly connected
   - Unix and Linux routing commands
   - Route selection

2 Mathematical representation

3 Routing in the Internet

4 Different routing mechanisms
Internet Routing (1)

- Based on top level structure defined by Autonomous Systems (AS)
- Each AS has administrative control over a collection of
  - Routers (and hosts)
  - Networks

Definition (AS — Autonomous System (from RFC 1930))

An AS (Autonomous System) is a connected group of one or more IP prefixes run by one or more network operators which has a single and clearly defined routing policy.
Edge routers inside an AS can be directly connected to edge routers in another AS

- Used for inter-AS routing
- Using an exterior routing protocol (EGP)
  - Example: BGP4
  - No other protocols in use (except variants of BGP)
Internet Routing (3)

- Routers within the boundary of a single AS communicate with each other to provide
  - Intra-AS routing
  - Using an interior routing protocol (IGP)
    - Examples: RIP, OSPF, IS-IS
Outline

1. Basic setup
   - Directly connected
   - Not directly connected
   - Unix and Linux routing commands
   - Route selection

2. Mathematical representation

3. Routing in the Internet

4. Different routing mechanisms
Routing Protocol Classification

- **Static**
  - A “human” protocol
  - Explicit commands like “route add” or “ip route add”
  - Side effect of a script (“semi-automatic/semi-human”)

- **Dynamic (automatically by a routing protocol)**
  - Distance Vector (for instance RIP)
  - Path Vector (for instance BGP)
  - Link State (for instance OSPF)
Distance Vector Routing

- Distance vector algorithm
  - Bellman-Ford (1957)
  - Distributed shortest path

- Original ARPANET routing algorithm

- Decentralised

- Asynchronous
Path Vector Routing

- Like distance vector routing, but
- Instead of the distance the complete path (on AS level) to the destination is given
- The algorithm is still
  - decentralised
  - asynchronous
- The algorithm might depend on explicit or implicit policies
Link State Routing

- Link state algorithm
  - Dijkstra (1959)
  - Single source shortest path

- Complete knowledge is distributed to all nodes in an area

- Knowledge about the local network topology is flooded to all participants in an area

- Every node executes the shortest path algorithm and draws the same conclusions