CS2003
IP routing

Recap: routing and forwarding

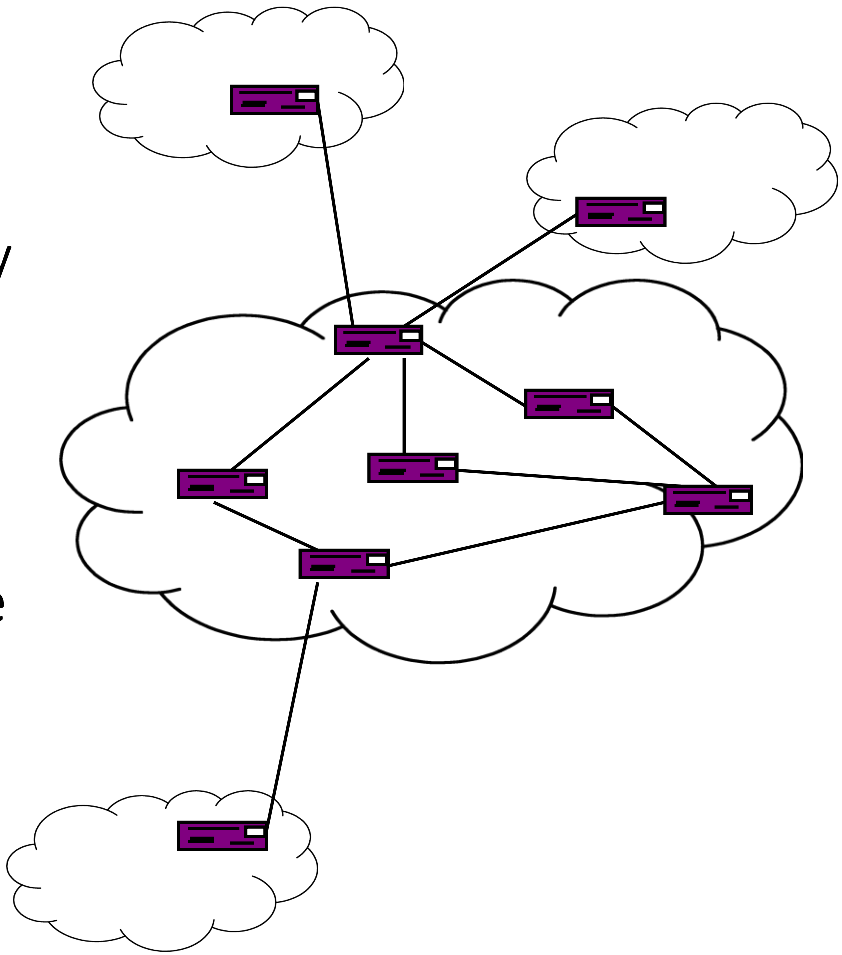
- **Forwarding:**
  - Receiving a packet, looking up destination address in a table, and sending the packet in the direction indicated by the table
- **Routing:**
  - The process by which forwarding tables are built
  - Discover paths through network, gather routing information, using a ***routing protocol***

Today: brief introduction to routing

- Example routing protocol (link state)
- Wider area routing (BGP)
- Examples of measuring routes
- If you want to know more, take CS3102 next year

Network topology – WAN

- Connectivity between sites:
  - “interconnection units”
- Network “clouds”:
  - need cloud-to-cloud connectivity
- IP addresses used:
  - to identify interfaces
  - globally unique
- Should an IP address/interface mapping be permanent?
  - do devices move location?



Routing information

- Fixed topology:
  - what is **supposed to be** connected to what?
- Dynamic topology:
  - what is **actually** connected to what?
- Historic information
- Metrics – "distance" to destination:
  - hop count
  - link throughputs
  - link delays, link jitters
  - link error rates
  - link financial costs (in £££)
  - etc.

Routing policies and constraints

- Administrative:
  - commercial agreements to carry traffic from certain sources
  - priority routes for some traffic
- Security:
  - avoid "untrusted" networks
- Quality of Service (QoS):
  - route certain traffic types via "suitable" links
- Real cost (financial):
  - only use high-cost links when all else fails

Routing protocols on the Internet

- Distributed routing:
  - no centralised operation and **control** of protocol
  - routing policy may be **managed** centrally
- Two main elements:
  - messages: **routing updates**
  - algorithm: **find routes**
- Combination of messages and algorithms provides behaviour of protocol.

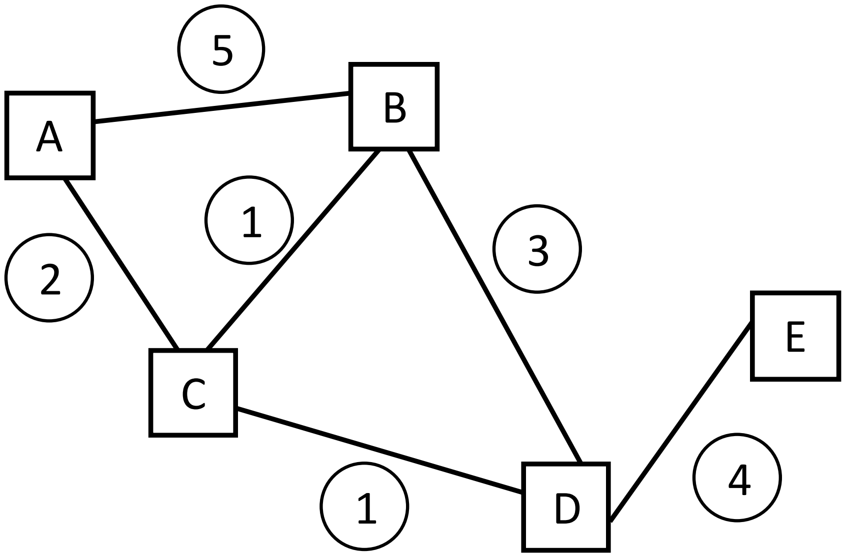
The original ARPANET routing algorithm

University of St Andrews

- **Distance-vector (DV)** – inherent problems:
  - Bellman-Ford
- ✓ Use queue length as metric:
  - ✓ diverts traffic away from congestion
- ✗ High capacity links not specially favoured
- ✗ Queue lengths are not stable
- ✗ Oscillations:
  - **route flapping**
- **So, replace it!**

Link-state algorithms

- Each node:
  - assesses **"cost"** of local links
  - distributes information to **all** nodes
  - receives information **from** all nodes
  - finds lowest cost path to all other nodes
- Dijkstra's SP algorithm:
  - **shortest-path (SP) tree** to all other nodes



Dijkstra’s algorithm

Dijkstra’s algorithm

- net topology, link costs known to all nodes
  - accomplished via “link state broadcast”
  - all nodes have same info
- computes least cost paths from one node (‘source”) to all other nodes
  - creates forwarding table for that node
- iterative: after k iterations, know least cost path to k dest.’s

notation:

- c(x,y): link cost from node x to y; $= \infty$ if not direct neighbours
- D(v): current value of cost of path from source to dest. v
- p(v): predecessor node along path from source to v
- N': set of nodes whose least cost path definitively known

Dijkstra's algorithm

```
Initialization:
 N' = {u}
 for all nodes v
   if v adjacent to u
       then D(v) = c(u,v)
   else D(v) = ∞

Loop
  find w not in N' such that D(w) is a minimum
  add w to N'
  update D(v) for all v adjacent to w and not in N' :
      D(v) = min( D(v), D(w) + c(w,v) )
  /* new cost to v is either old cost to v or known
   shortest path cost to w plus cost from w to v */
until all nodes in N'
```

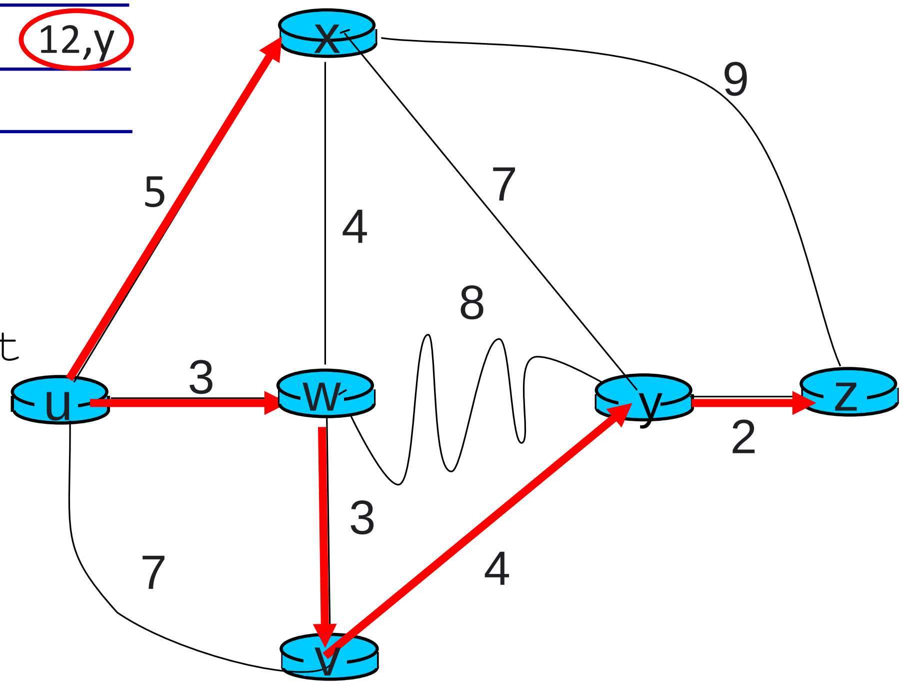
Dijkstra's algorithm: example

Step	N'	D(v) p(v)	D(w) p(w)	D(x) p(x)	D(y) p(y)	D(z) p(z)
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxvyz					

- construct shortest path tree by tracing predecessor nodes
- ties can exist (can be broken arbitrarily)

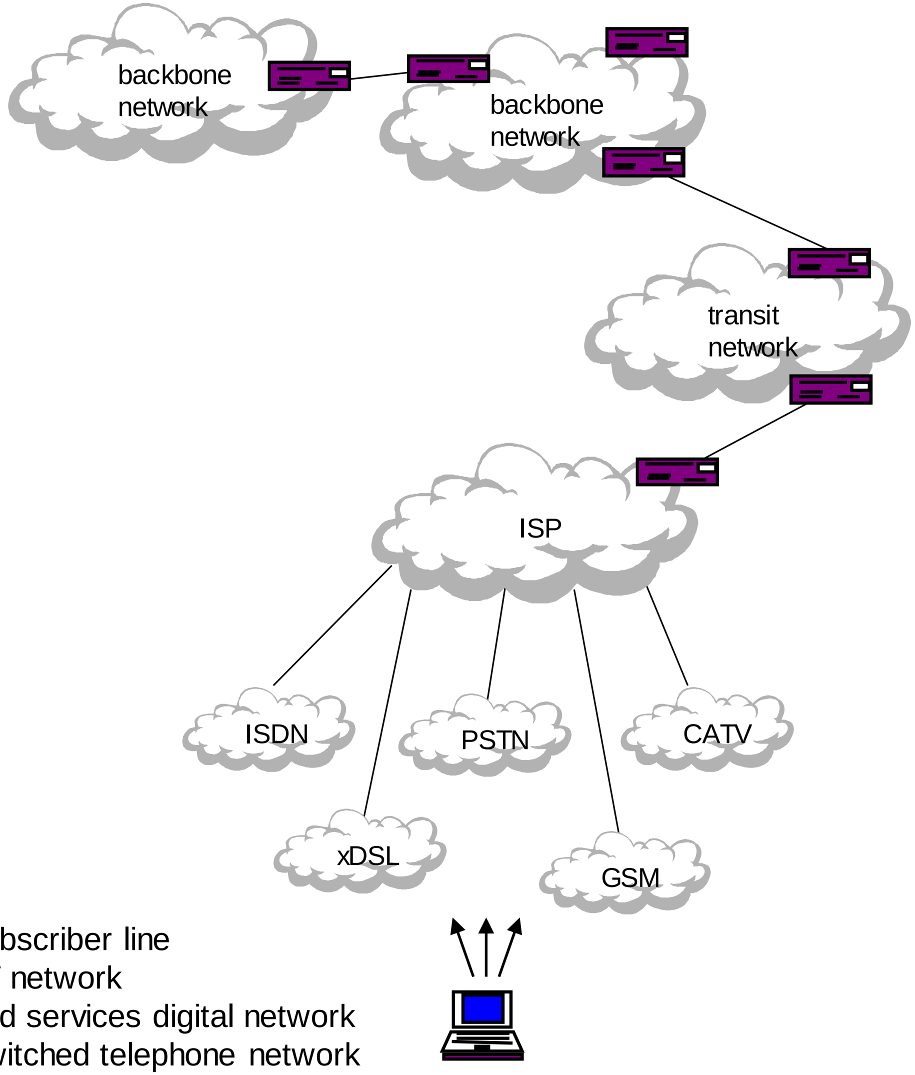
Routing table for u:

destination	next hop	cost
v	w	6
w	w	3
x	x	5
y	w	10
z	w	12



Internet

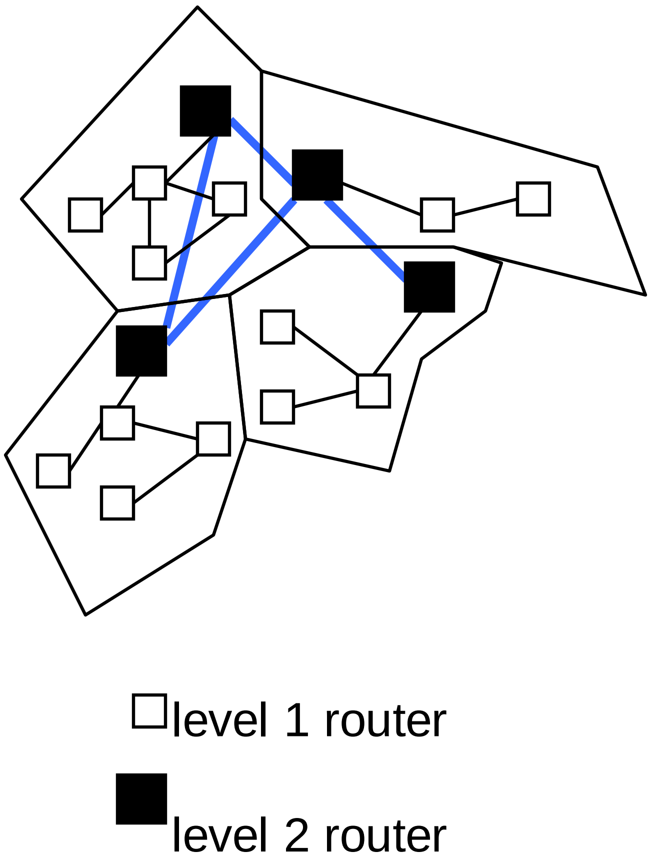
- Connectivity between network clouds:
  - internal workings hidden
  - IP provides convergence
- Backbone providers:
  - commercial network operators
- Transit networks:
  - may be IP-aware
  - may provide basic connectivity only (possibly not IP-aware)



xDSL	digital subscriber line
CATV	cable TV network
ISDN	integrated services digital network
PSTN	public switched telephone network
GSM	Global System for Mobile communication

Hierarchical routing

- Area:
  - network cloud, e.g. administrative domain
  - single routing protocol within area
- Connectivity between areas:
  - hierarchy of routers
- Routing at levels:
  - allows routing information to be aggregated



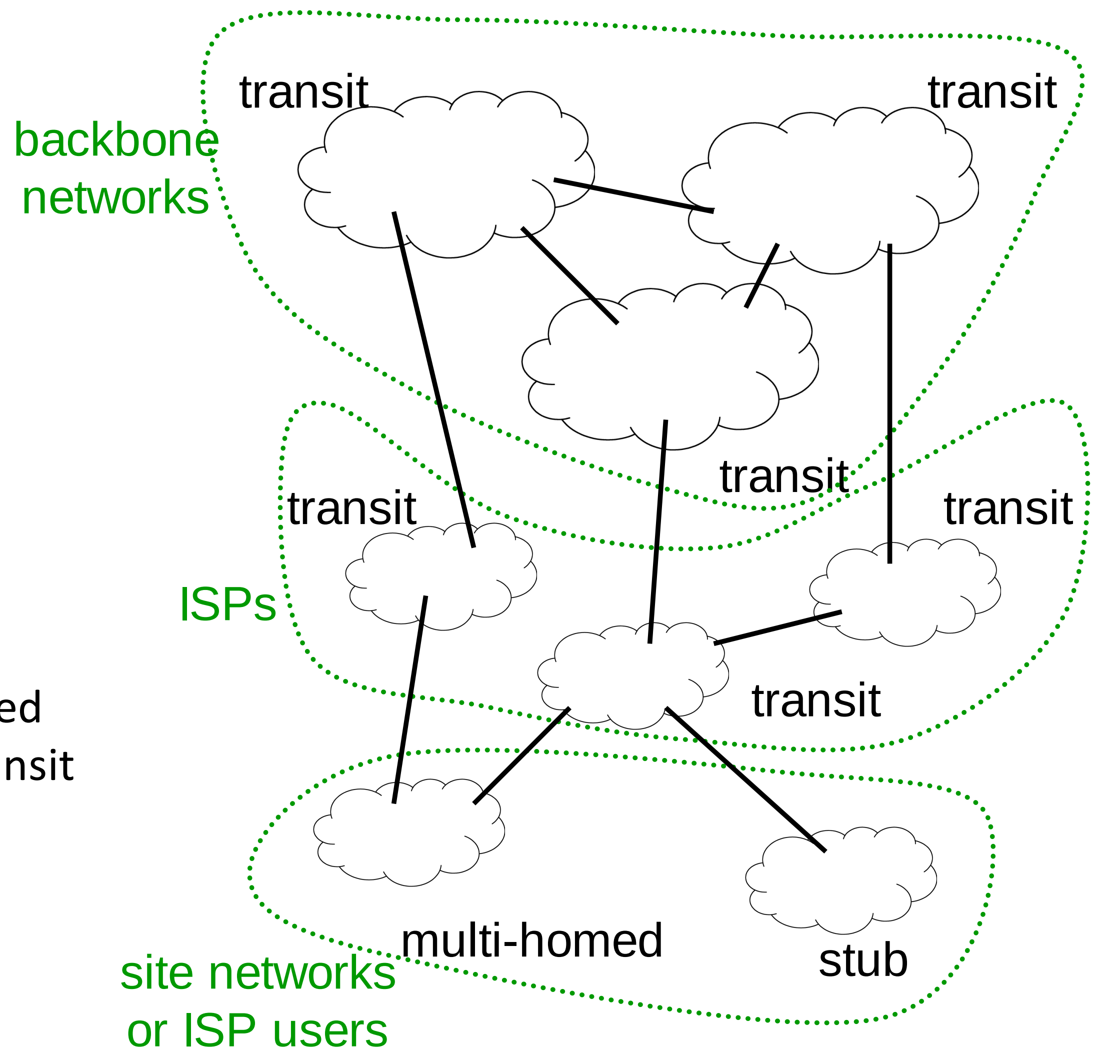
Autonomous Systems

- Internet connectivity is partitioned along **administrative boundaries**:
  - **Autonomous System (AS)** boundaries
  - AS identified by AS numbers: https://www.iana.org/assignments/as-numbers/as-numbers.xhtml

    16-bit or 32-bit numbers, uniquely identifying an AS
- Within an AS (intra-AS):
  - typically a single routing protocol, e.g. OSPF (a link-state protocol), various proprietary protocols also.
- Between ASs:
  - need a consisting/common routing protocol
  - Border Gateway Protocol (BGP)
  - **policy-based routing**

Internet – large-scale structure

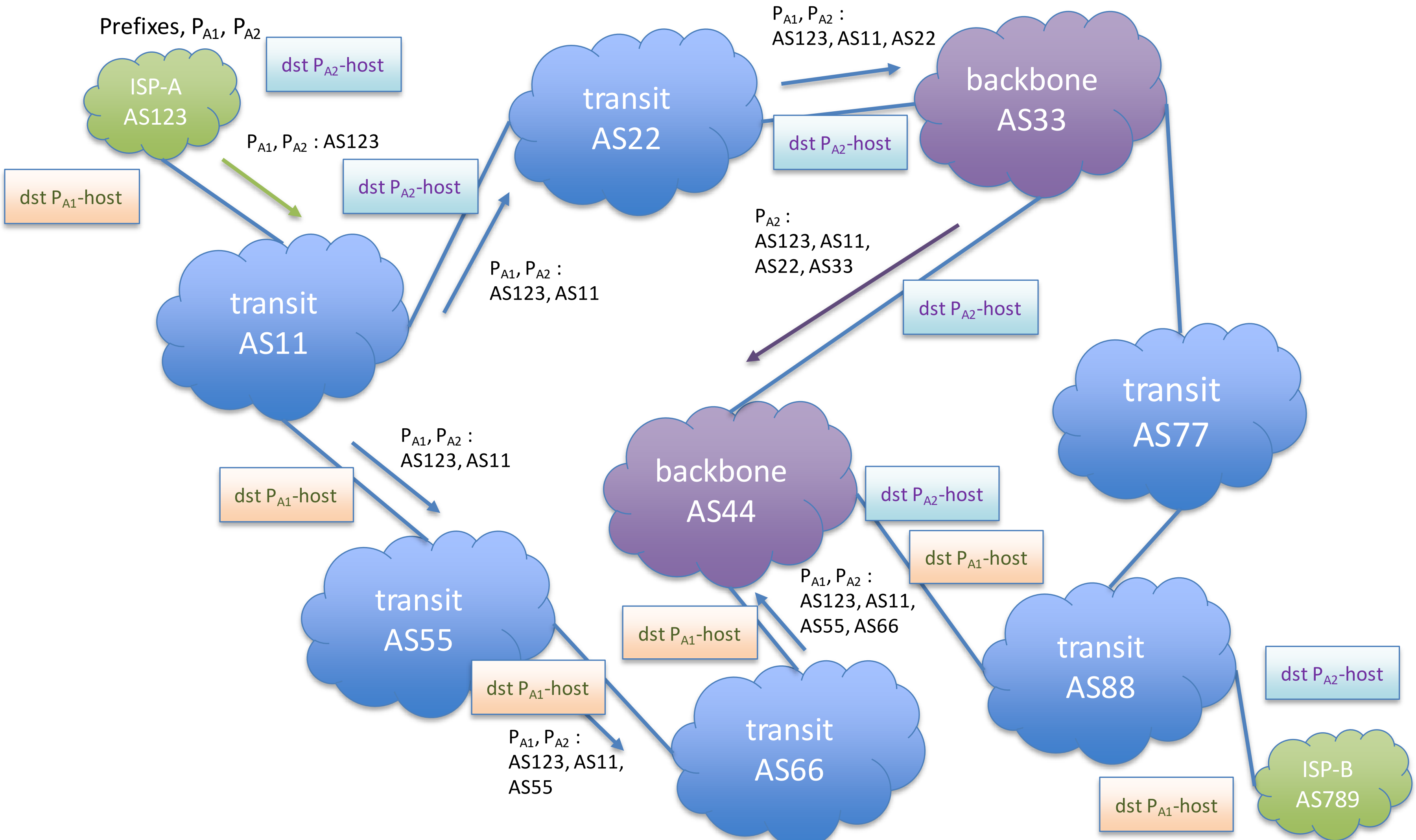
- Traffic types:
  - local: intra-AS
  - transit: inter-AS
- AS:
  - stub AS: e.g. site network
  - multi-homed AS: e.g. ISP
  - transit AS: e.g. backbone provider
- Internet:
  - collection of interconnected stub, multi-homed and transit ASs



BGP – inter-AS routing

- **Border Gateway Protocol (BGP)**:
  - inter-AS protocol
  - between "border routers" ("border gateways")
  - reachability information – no routing metric
  - **path vector** – full AS path
  - policy-based routing possible
- Updates are less frequent than for an intra-AS protocol:
  - large-scale structure of the Internet is relatively stable
- Information exchanged with "neighbours":
  - neighbours configured manually by network administrators
  - TCP used for information transfer

BGP path vector protocol (simplified!) (1)



BGP path vector protocol (simplified!) (2)

- In reality, ISPs are ASs also.
- BGP (RFC4271) messages (updates) have:
  - **path attributes**
  - authentication information
- BGP is mainly policy based:
  - commercial (service level) agreements (SLAs)
  - time of day
  - “political” requirements
  - multi-homing (multiple connectivity)

Some network exploration tools

- ping:
  `$ ping www.caida.org`
  `$ ping –n www.caida.org`
  check if a remote host is up (to network level)
  (sometimes disabled in network or end-host)
- traceroute:
  `$ traceroute www.caida.org`
  `$ traceroute -n www.caida.org`
  find the network path to the remote host
  (sometimes disabled in network end-host)
- Various `whois` services:
  IP address registration records, e.g.:
  https://www.nominet.uk/whois/
  (lookup "ac.uk" – it is registered to JANET https://ja.net/)

Measurement of links (1)

$$T_x = b / r$$

T_x	**transmission delay**
	(time taken to put bits on to the wire)
b	number of bits
r	data rate (bits per second, b/s)

Measurement of links (2)

$$T_p = d / s_s$$

T_p	**propagation delay** of the signal (time taken for a signal to traverse a link)
d	distance of link (metres)
s_s	speed of signal (metres per second, m / s)

Measurement of links (3)

$$T_d = T_x + T_p$$

T_d one-way delay on a link (or path)
(time taken for a signal to traverse a link)

Measurement of paths

- Approximation of delay for end-to-end **path** (rather than an individual **link**).
- Use ***ping*** to measure one-way delay of the ***whole end-to-end path***.
  - assumes path is symmetric
  - assumes all packets treated equally
  - uses ICMP (Internet Control Message Protocol) to send low-level messages (sits at network layer)
- Use ***traceroute*** to see the ***individual links*** of the whole end-to-end path.
  - times from ping and traceroute may not be equal: measurement errors, path / traffic effects, and system-level effects.

Summary

- Example routing protocols:
  - link-state routing, e.g. OSPF:
    - Dijkstra's shortest path algorithm
- Hierarchical routing
- Autonomous Systems:
  - policy-based routing: BGP
- Measurement
  - ping, traceroute
- Reading: Peterson & Davie Ch 3.4, Kurose & Ross Ch 5.1-5.4