S1E8 - Routing Protocols

Nicholas Morrison nick@nanocat.net

What is Routing?

- moving layer 3 packets from one network to another
- .. by choosing an outgoing interface
- .. by referencing the routing table (`show ip route`)

CIDR: Classless Inter-Domain Routing

- CIDR: Classless Inter-Domain Routing
- Notation xxx.xxx.xxx.xxx/yy - `192.168.100.100/24`
- /yy = the number of bits that represent the “network”
  - called “mask” (or “netmask” or “network mask” or “subnet mask”)
- remaining bits = the number of bits that represent the “host”

CIDR examples: /24

192.168.20.20/24

Decimal	192	168	20	20
Binary	1100 0000	1010 1000	0001 0100	0001 0100
Bit #	1.....8	9.....16	17.....24	25.....32

- the **first 24 bits** define the **network**
- 192.168.20.0 (zero out all the other bits) == **network address**
- last 8 bits (192.168.20.**20**) == host part
- 8 bits == max 254 host addresses
- we can say:

  - the host with the address 192.168.20.20/24 is in the 192.168.20.0/24 network
  - 192.168.20.0/24 contains up to 254 hosts

CIDR examples: /16

192.168.20.20/16

Decimal	192	168	20	20
Binary	1100 0000	1010 1000	0001 0100	0001 0100
Bit #	1.....8	9.....16	17.....24	25.....32

- the **first 16 bits** define the **network**
- 192.168.0.0 (zero out all the other bits) == **network address**
- last 16 bits (192.168.**20.20**) == host part
- we can say:
  - the host with the address 192.168.20.20/16 is in the 192.168.20.0/16 network
  - 192.168.0.0/16 contains up to 65,534 hosts

CIDR examples: /8

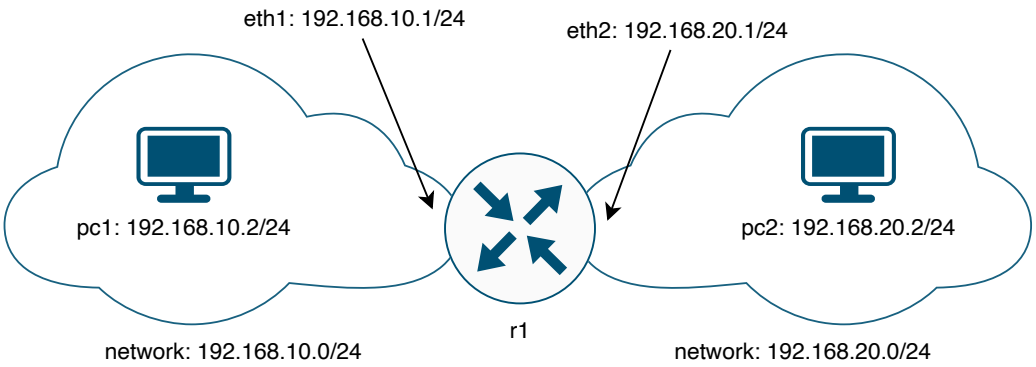
192.168.20.20/8

Decimal	192	168	20	20
Binary	1100 0000	1010 1000	0001 0100	0001 0100
Bit #	1.....8	9.....16	17.....24	25.....32

- the **first 8 bits** define the **network**
- 192.0.0.0 (zero out all the other bits) == **network address**
- last 24 bits (192.**168.20.20**) == host part
- we can say:
  - the host with the address 192.168.20.20/8 is in the 192.0.0.0/8 network
  - 192.0.0.0/8 contains up to 16,777,214 hosts

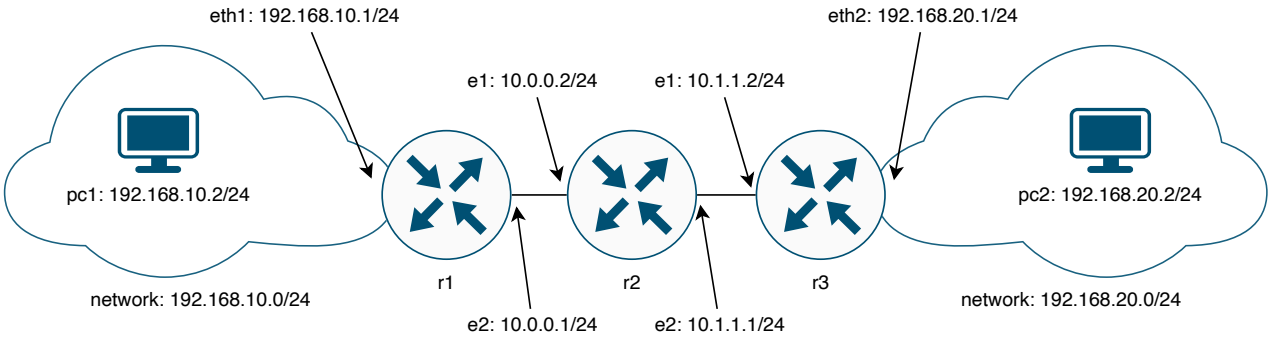
... what is routing?

- moving packets between different Layer 3 networks!



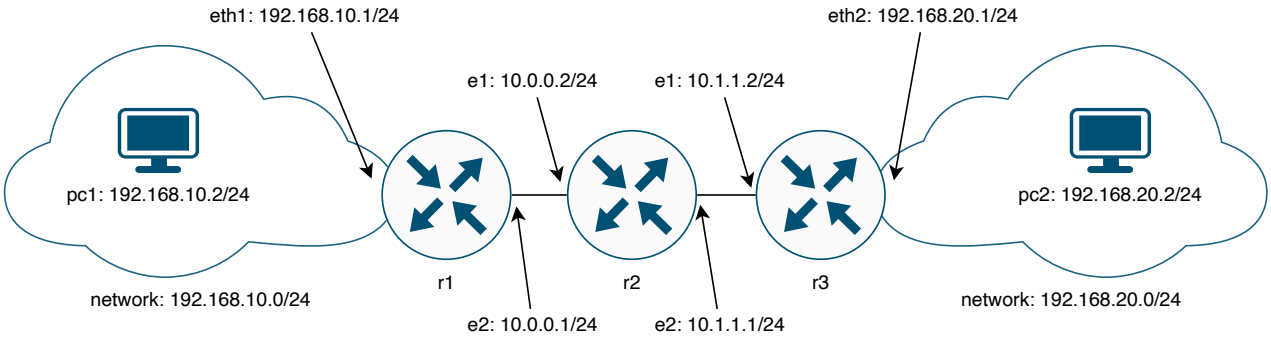
- pc1 wants to send a packet to pc2
- eth1: 192.168.10.1/24
- eth2: 192.168.20.1/24
- enough information to route!

More routers



- r1 knows 192.168.10.0/24
- r1 knows 10.0.0.0/24
- r1 does not know where 192.168.20.0/24 is! ??? OwO ???

Static routes



We can help the routers with static routes to get to 192.168.20.0/24:

```
r1: ip route 192.168.20.0/24 10.0.0.2
r2: ip route 192.168.20.0/24 10.1.1.2
```

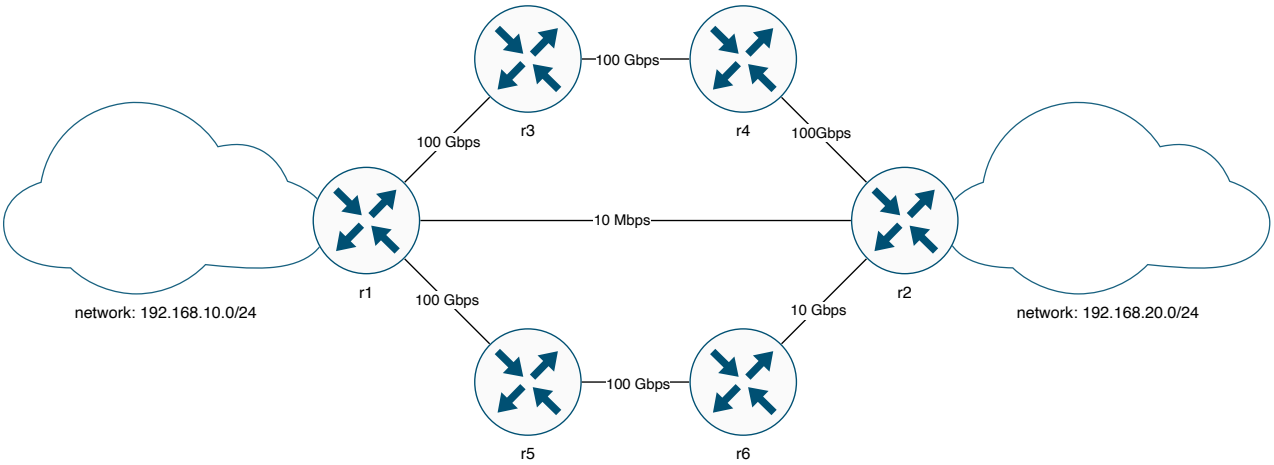
.. and for the way back to 192.168.10.0/24:

```
r3: ip route 192.168.10.0/24 10.1.1.1
r2: ip route 192.168.10.0/24 10.0.0.1
```

The problem with static routes

- they don't scale
- they are prone to error
- it's a mostly manual process
- totally ok for very small networks
- very difficult for any other network

The solution: Routing Protocols!



- share reachability information
- provides loop-free paths
- every router runs the same protocol
- every router sees the network from its own perspective
- answers the question: which interface to send this packet through?

So many protocols

- RIP, OSPF, EIGRP, ISIS, BGP...
- do essentially the same thing
- with vastly different levels of:
  - scalability
  - security
  - complexity
  - awesomeness

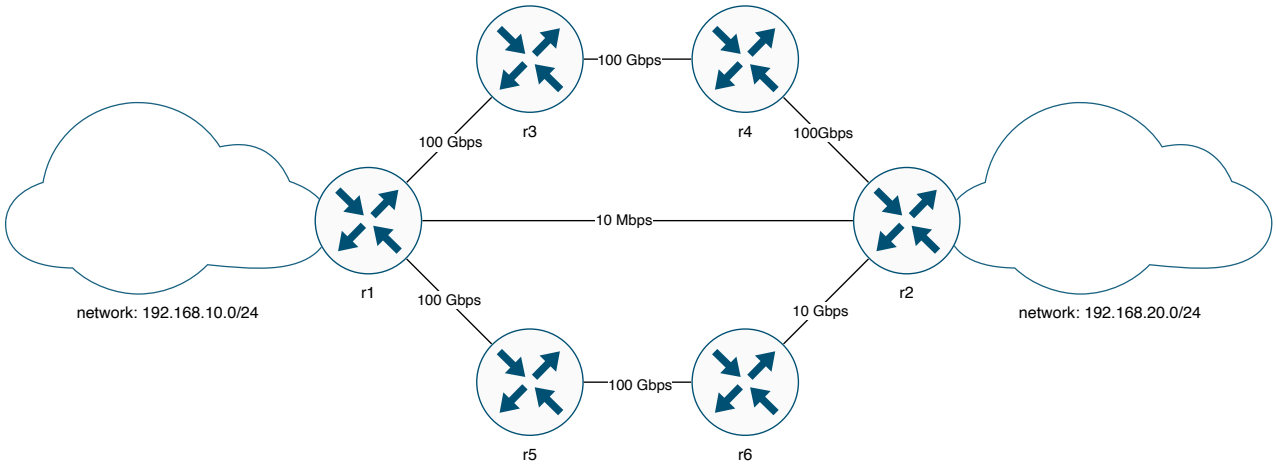
AS (Autonomous System)

- a network controlled by a single organisational entity
- eg a university, a business, a government
- maybe your organisation needs an assigned “AS Number”
- AS numbers are globally unique
- assigned by globally-cooperating RIRs (Regional Internet Registry)

RIP (the Routing Information Protocol)

- **distance vector**
- used within a single AS
- simple code, simple configuration, low CPU usage
- good for small networks

RIP path selection



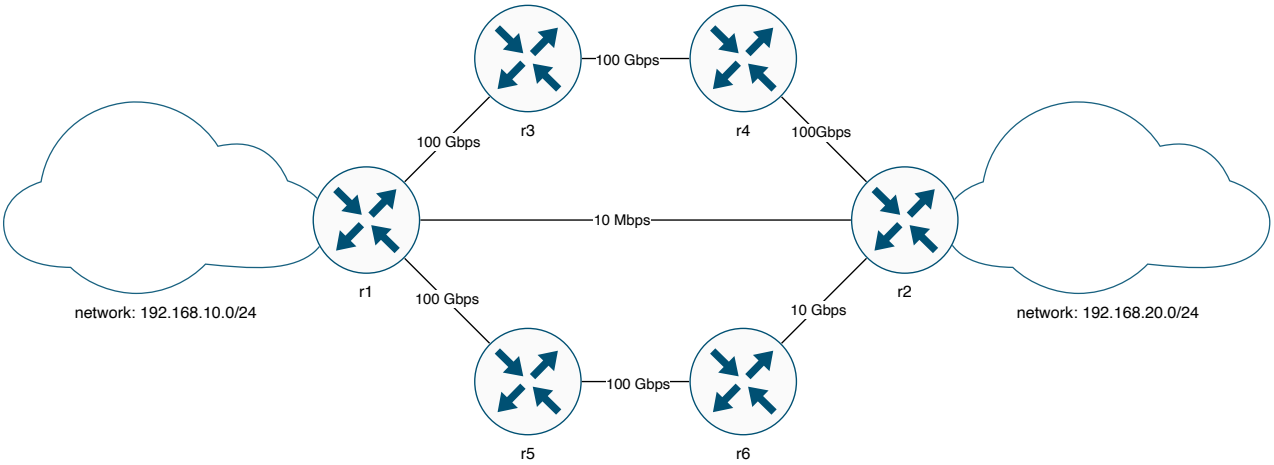
- decides the best path based purely on hop-count (distance)

```
r1 r2       = 2 hops   <- RIP chooses this path
r1 r3 r4 r2 = 4 hops
r1 r5 r6 r2 = 4 hops
```

OSPF (Open Shortest Path First), ISIS (Intermediate System to Intermediate System)

- **link state vector**
- used within a single AS
- much less simple code
- much more scalable than RIP

OSPF, ISIS path selection



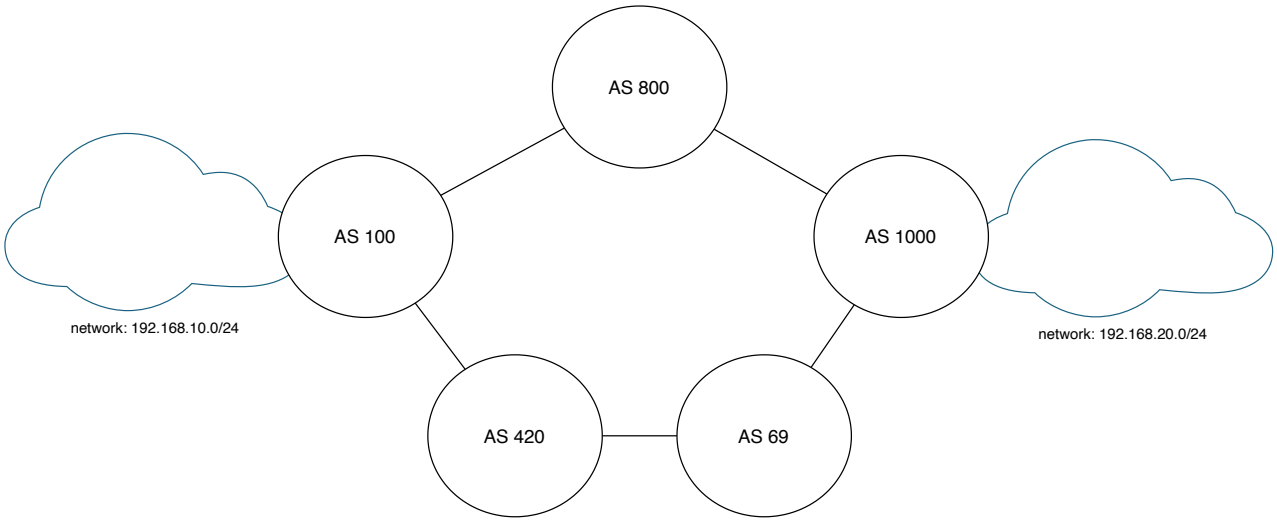
- routers assign a “cost” to their interfaces, for example:
  - 100 Gbps cost 1
  - 10 Gbps costs 10
  - 1 Gbps costs 100
  - 100 Mbps costs 1000
- path is chosen by lowest cost (link state)

```
r1 r2       = 1000
r1 r3 r4 r2 = 3      <- OSPF chooses this path
r1 r5 r6 r2 = 12
```

BGP (the Border Gateway Protocol)

- **path vector**
- used *globally* between multiple ASs
- requires global collaboration to function
- complex code, very memory-intensive
- can store reachability information for the entire Internet

BGP path selection



- path is chosen by AS path length
  - with a long list of tie-breakers

```
100 800 1000    = 3   <- BGP chooses this path
100 420 69 1000 = 4
```

- my personal favourite protocol

Which one to choose?

- tiny networks: RIP or OSPF
- small-large networks: OSPF
- global networks: OSPF and BGP
- the internet: BGP

We are going to learn all of these protocols.

Questions?