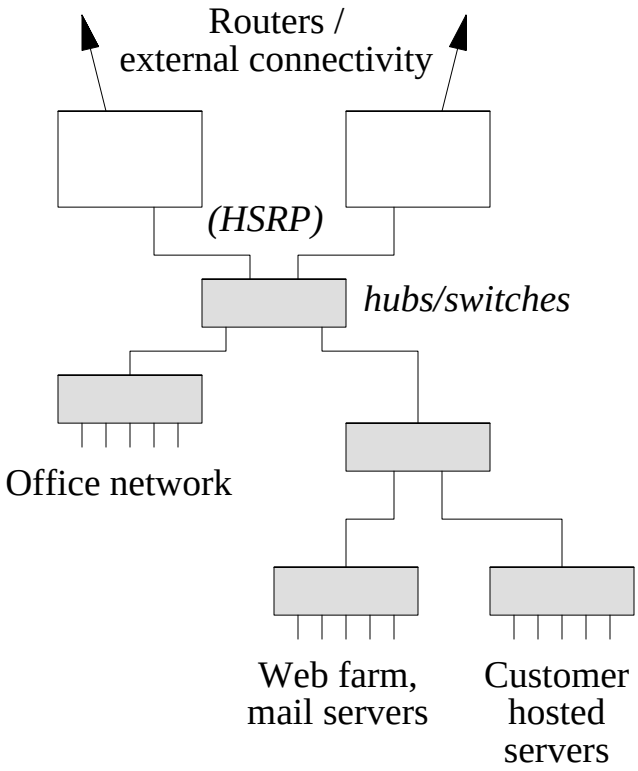
Don't build your network like this...



What is wrong with this design?

Principles to follow

- ☛ It's better to have part of your network fail than your whole network fail
- ☛ Keep different types of traffic - especially different levels of trust - on PHYSICALLY SEPARATE NETWORKS (not just separate subnets on secondary addresses on the same cable) - separated at layer 3
- ☛ If you have anything redundant (e.g. power supplies, fans, network links), make sure they are continually monitored

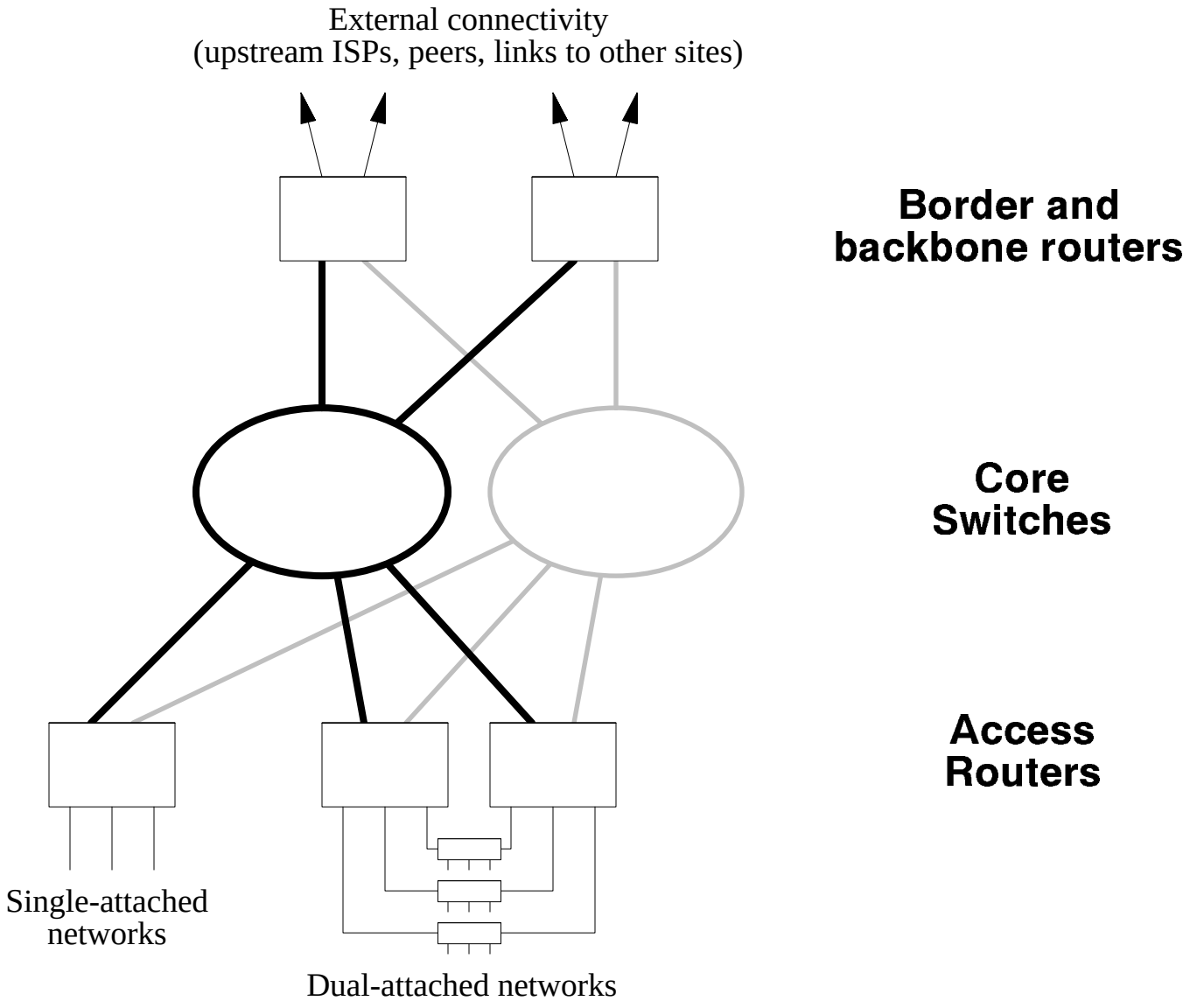
Approaches to resilience:

(1) Buy components which are inherently resilient

(2) Build your network so it can withstand failures

(3) Do both

A ready-made core network design



Applicability:

Network within a single site. If you have multiple sites, replicate this design at each site

Features of this design

Scalability

- ✓ Links (switch ports) can be 10M, 100M, gigabit, or any combination
- ✓ Whenever you run out of router ports, just plug another router (or pair) into the core
- ✓ Core switches give you very high aggregate bandwidth: e.g. 5 routers, each with two 100M interfaces = 1000M total bandwidth

Resilience

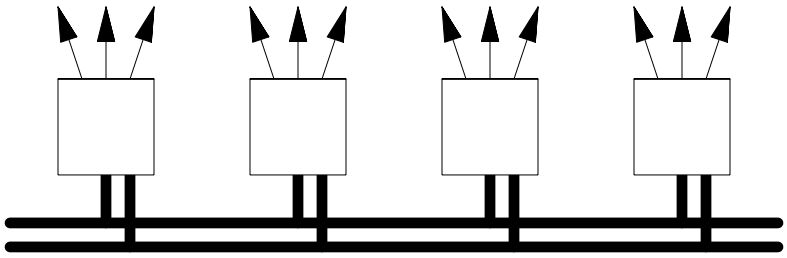
- ✓ All routers are dual-attached; can withstand a failure of any single interface, cable, or core switch
- ✓ OSPF provides end-to-end test of each link
- ✓ Traffic is always active down *both* paths, so you have confidence that they are working

Ease of maintenance

- ✓ Can power down and exchange core switches one at a time, with minimal disruption to network
- ✓ Flexibility to mix and match routers; outages limited to part of network

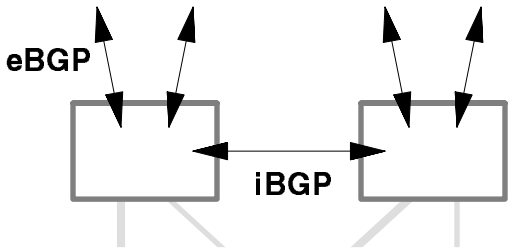
Observation: all packets go through no more than 2 routers to reach their destination.

P.S. There is no magic in this design. An alternative way of thinking about it is that you have a bunch of routers which exchange data via a private network (two of them, for resilience)



Backbone and border routers

External connectivity
(upstream ISPs, peers, links to other sites)



Terminology

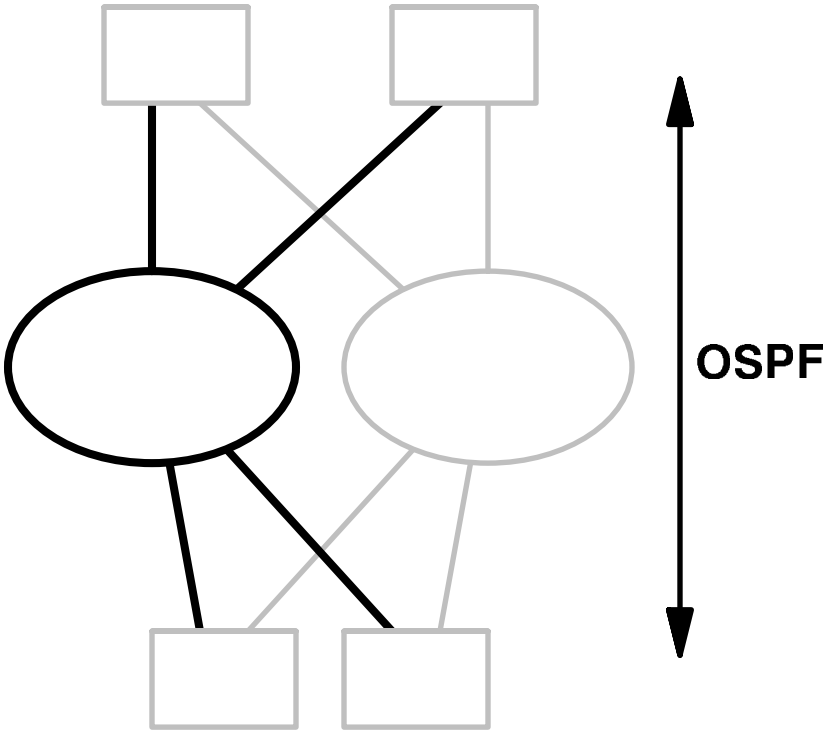
Border routers connect to your upstream providers or peers. *Backbone routers* connect to WAN links between your sites. As far as this design is concerned, they are the same.

BGP

- Each router is dual-homed: one ethernet interface into each core.
- If you only have a single border router that's fine, there will be no BGP, just a static default to your upstream provider. It's a single point of failure of course.
- If you have multiple border/backbone routers, they will iBGP peer with each other.
- Your iBGP peering sessions should be between **loopback interfaces** on your routers. This is so that if connectivity into the core switches changes, it does not affect any iBGP sessions.

```
neighbor x.x.x.x remote-as yyyy
neighbor x.x.x.x update-source Loopback0
```

Core network



General

- Thanks to equal-cost multipath, a router with 2x10M connections has 20M of bandwidth available; a router with 2x100M connections has 200M available

OSPF

- Border routers only should originate **defaultroute** into the OSPF cloud. Never attempt to redistribute BGP into OSPF!
- Originate defaultroute with a different metric from each border router

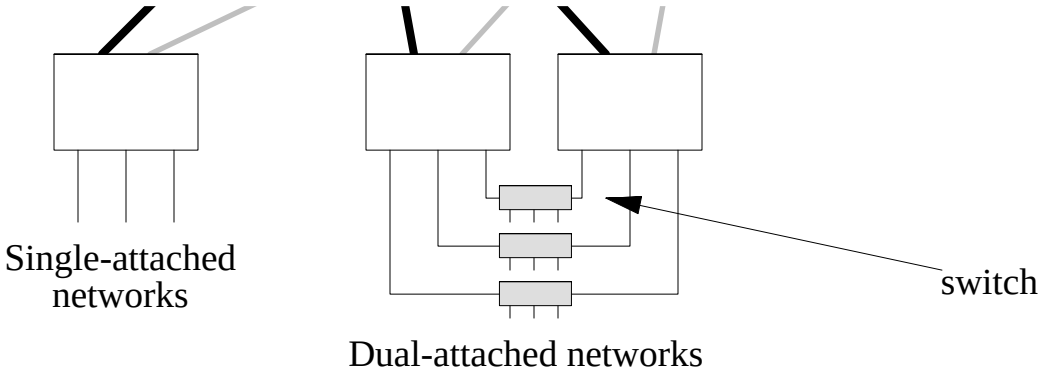
(If you follow these recommendations, the router with the lowest defaultroute metric will be the "preferred" router for outgoing traffic from the access routers. Routing of outgoing traffic may not be optimal - it may hit the 'wrong' router first and be bounced to the right one - but this is typically not a problem)

- Use MD5 authentication. Disable OSPF except on necessary interfaces.
- For ease of maintenance: use hellointerval 2 secs, routerdeadinterval 8 secs
- Choose appropriate costs, so that 10M is more expensive than 100M, and 100M more expensive than gigabit
- Try to keep routes within one site within larger netblocks. This will allow you to use multi-area OSPF and aggregation later. (Often not possible with customer netblocks)
- Notice that the same routers are visible as neighbors by multiple paths. Some older versions of IOS, e.g. 11.3(8), had problems with this; if you lost one connection, they would think that the neighbor was not reachable via the other. Use a recent 12.0 release.

Choice of switches: note, must be switches, not hubs!

- The key feature to look for is non-blocking. performance The scalability depends on being able to pump more and more traffic through the switch without it losing packets.
- Appropriate number and speed of ports for the types of routers you want to connect
- For extra reliability, choose switches with redundant power supplies and fans if you can afford them
- SNMP management (allocate one IP address on each core to the switch itself)

Access routers



Single-attached networks

- Single router connects into both cores. Where resilience is less important, or cannot be provided anyway (e.g. serial ports to leased lined customers)

Dual-attached networks

- Two routers, using HSRP/VRRP or equivalent
- Use for essential services such as mail and dial-in servers
- Note that the two routers will still have to connect to a common switch, which is a single point of failure. So it's still worth putting servers on separate IP subnets where you can: especially radius servers, DNS servers.

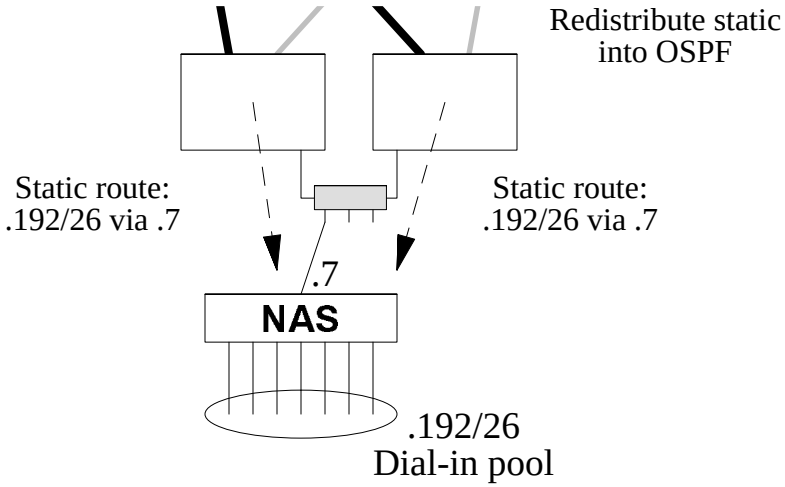
Connecting dial-in servers (NAS)

- In theory, you *could* connect them directly into the core switches, IF they have two ethernet ports, AND they have an OSPF implementation that you trust, AND you are sure they will not flap routes (e.g. announce a separate /32 for each dial-in user!)
- Safer bet to connect them downstream of access router(s), with defaultroute/HSRP. Use static routes on the access routers to point to their dial-in IP pool. These routers will then "redistribute static" to let the other routers know about it. Gives you more control and less reliance on possibly unreliable OSPF implementations.

Choice of router

- Enough ports, including 2 ethernet ports to link into core. A 5-port ethernet router will only give you 3 ports usable for networks, since the other 2 are for linking into the core.
- Enough performance to handle the full load of traffic to/from the core.
- Robust OSPF implementation with equal-cost multipath support.
- On large Ciscos: remember to enable Cisco Express Forwarding (ip cef [distributed]).
- Redundant PSUs and fans are nice, especially for single-attached networks

Connecting a NAS



- NAS does not need to participate in OSPF
- Single netblock for dial-in pool announced to rest of network
- *!! You will break static IP accounts unless all static IP users hit the same NAS !!*

Beware Static IP dial-ins

Do you provide a static IP dial-in service?

- ▲ IT DOESN'T SCALE
- ▲ The last thing you want is flapping /32 routes throughout your network whenever customers dial in or hang up. They will kill your routers.
- ▲ It might work now, but you will have severe problems in the future
- ▲ Supporting it will become very expensive

Suggested solution

1. Make sure that all static IP addresses are within the same netblock(s)
2. Make sure your existing static IP users all hit ONE NAS. You may have to set up a separate telephone number (hunt group) and get them to use this number.
3. Static route this netblock to this NAS
4. DON'T SELL ANY MORE STATIC IP ACCOUNTS

Do your customers *really* need static IP anyway?

- ☛ In many cases, what they really want is SMTP mail delivery. There are other ways of providing this.
- ☛ Set up a mailserver which can 'kick' out mail via SMTP to their dynamic IP address. (Tools you can use include 'fetchmail' and 'serialmail'). It can be triggered by dialling in (in your radius server), or by ETRN. Unfortunately this requires some scripting or coding.
- ☛ Usually this is for NT Exchange servers, and there is now a module available which can pull mail using POP3 instead.
- ☛ If there is a big financial disincentive to using static IP, customers will be more creative in finding alternative solutions.