1. Whole network is one large broadcast domain. Broadcasts packets are seen everywhere.
   (Windows NT servers are worst offenders)

2. All switches must learn all MAC addresses. Hosts have large ARP tables.

3. Security: customer hosted servers and office machines can break your mail and web networks (by configuring a wrong IP address, ARP spoofing etc)

4. There's no such thing as a Layer 2 backbone, so any network problems are very hard to locate

5. A broadcast storm in one part of the network will affect the whole network

6. Top switch/hub is a single point of failure. Replace it and your whole network stops working for a while

7. Switches form a tree. There are no backup links.

8. All traffic aggregates at the central switch which could be a performance bottleneck.

9. When things go wrong, the central switch could also go down.

The best way is to have part of your network fail than your whole network fail

(1) Buy components which are inherently resilient
(2) Build your network so it can withstand failures
(3) Do both

What is wrong with this design?

Principles to follow:

Approaches to resilience:

- Keep different types of traffic - especially different levels of trust - on physically separate networks.
- If you have anything redundant (e.g. power supplies, fans, network links), make sure they are continually monitored.
- If you have anything redundant (e.g. power supplies, fans), keep different types of traffic - especially different levels of trust - on physically separate networks.
- If your network fails, it’s better to have part of your network fail than your whole network fail.
- Buy components which are inherently resilient.
- Build your network so it can withstand failures.
- Do both.

It’s better to have part of your network fail than your whole network fail.

Don’t build your network like this...