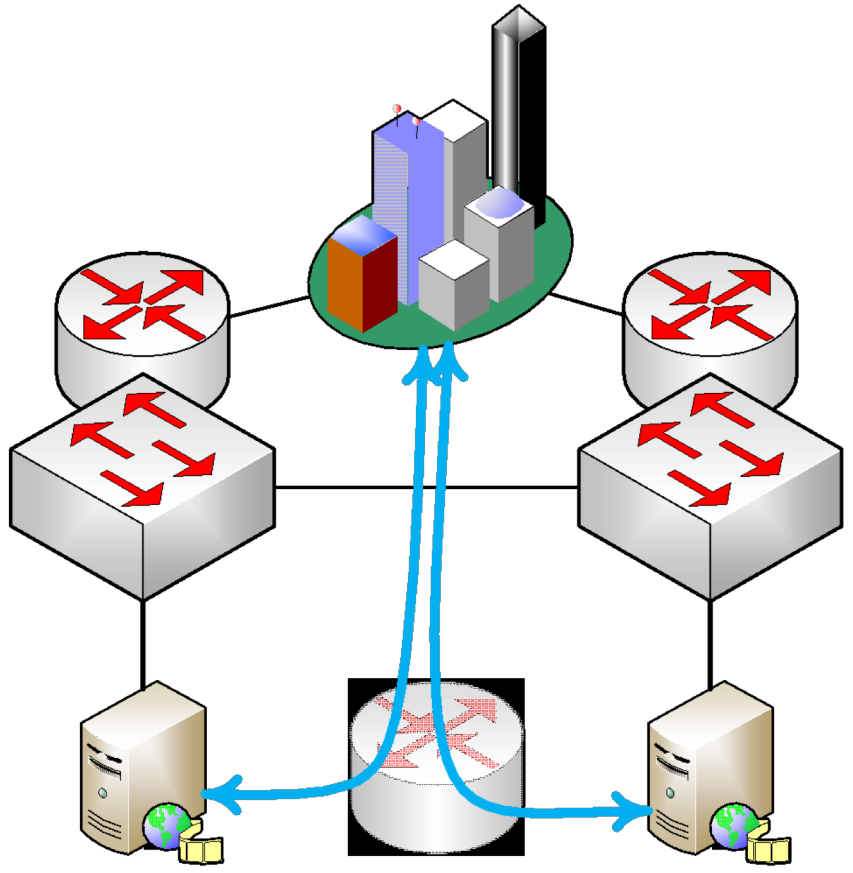
ND Spoofing for Fun and Profit
Distributing server farm traffic efficiently

Lutz Donnerhacke

IKS Service GmbH

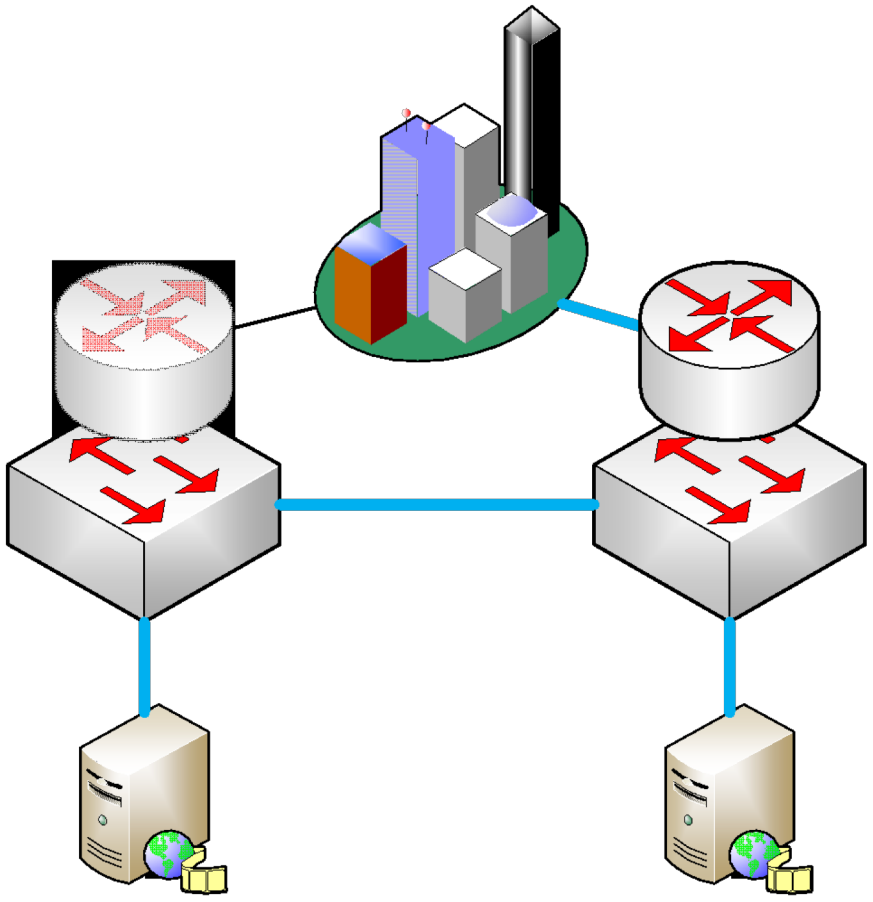
The Problem

- High bandwidth servers
- Distributed clients
  - Distribute locations
  - Intermediate bandwidth limited
- Third party appliances
  - Internal communications?
  - Single default gateway
  - No technical contacts
- Design violation
  - Should buy two clusters



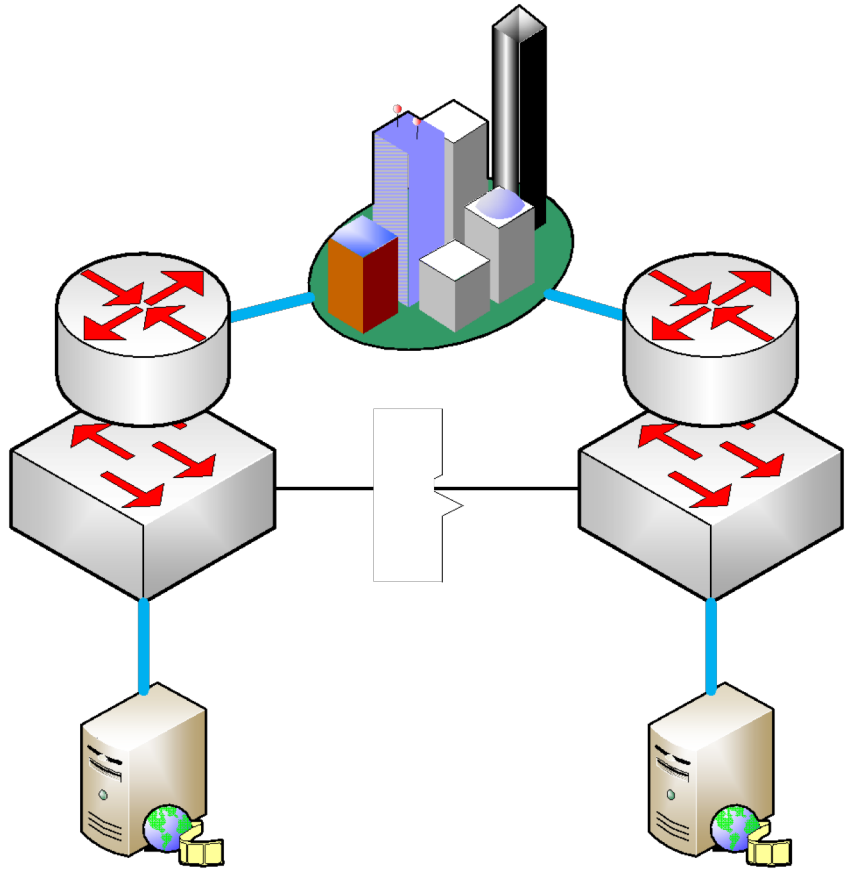
First Hop Redundancy

- Single active router
  - HSRP, etc.
  - Failover
- Traffic flow
  - Deterministic
  - Not optimal
  - Intermediate bandwidth required



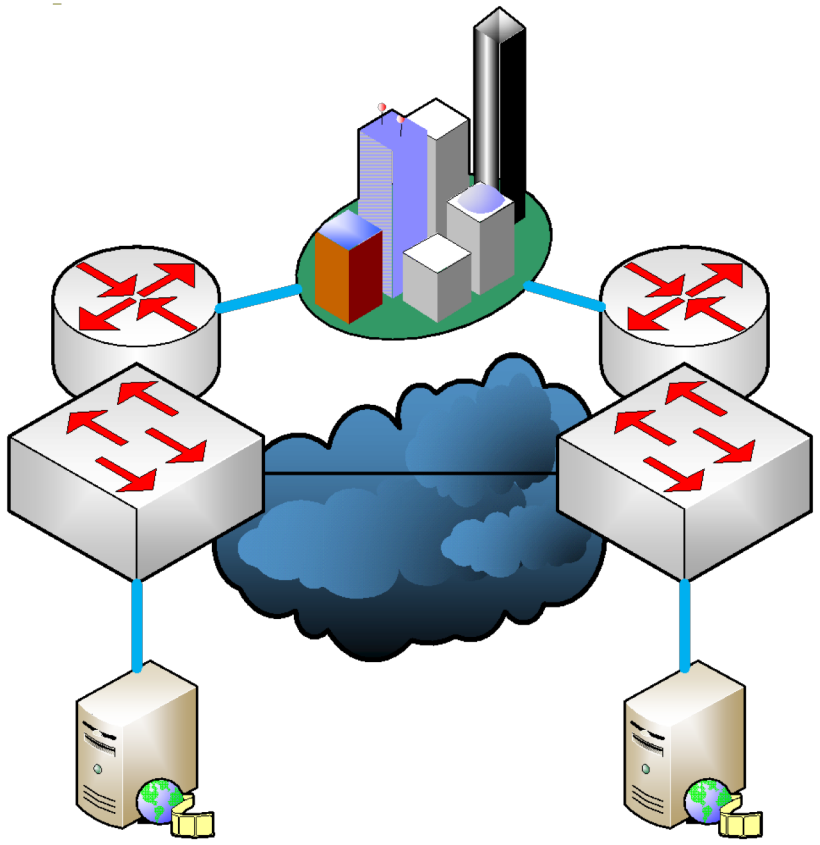
Disturb First Hop Redundancy

- Prevent FHR communication
  - Both nodes active
  - Complicated, error prone
- Low latency = local
  - First come, first serve
  - Slow and unstable redundancy
- Do not disturb the cluster
  - May harm internal communication
- Hard to operate
  - Always a fail state



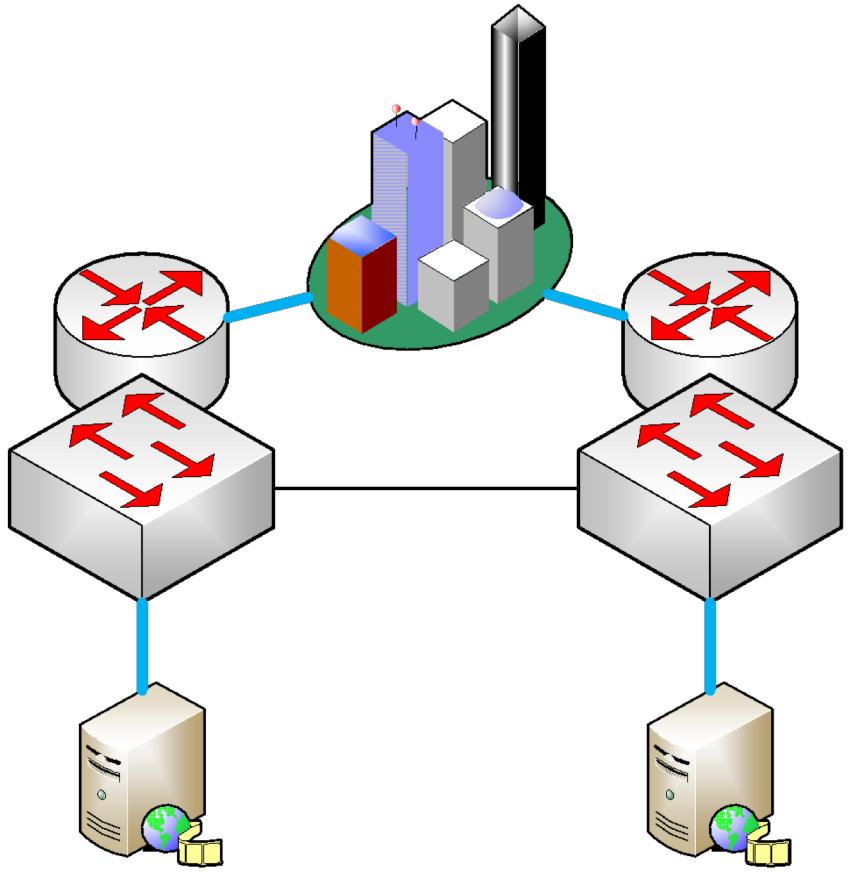
SDN for the rescue

- Inject the router twice
  - MAC into BGP
  - Least cost route
- Pro
  - Stable
  - Redundant
- Con (for us)
  - Redesign of core network
  - Expensive



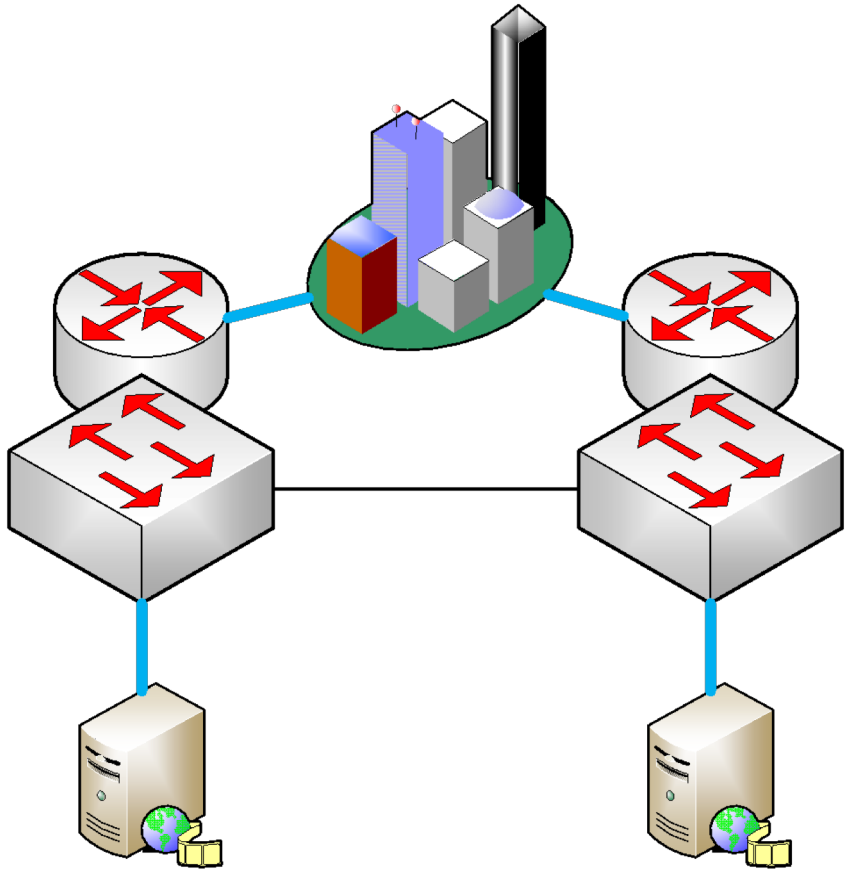
Back to the blackboard

- Different gateways
  - Each server has an other router
  - HSRP still possible
- Locality depend configuration
  - Communicate with vendor
  - Change application
  - Change rollout
- Unlikely



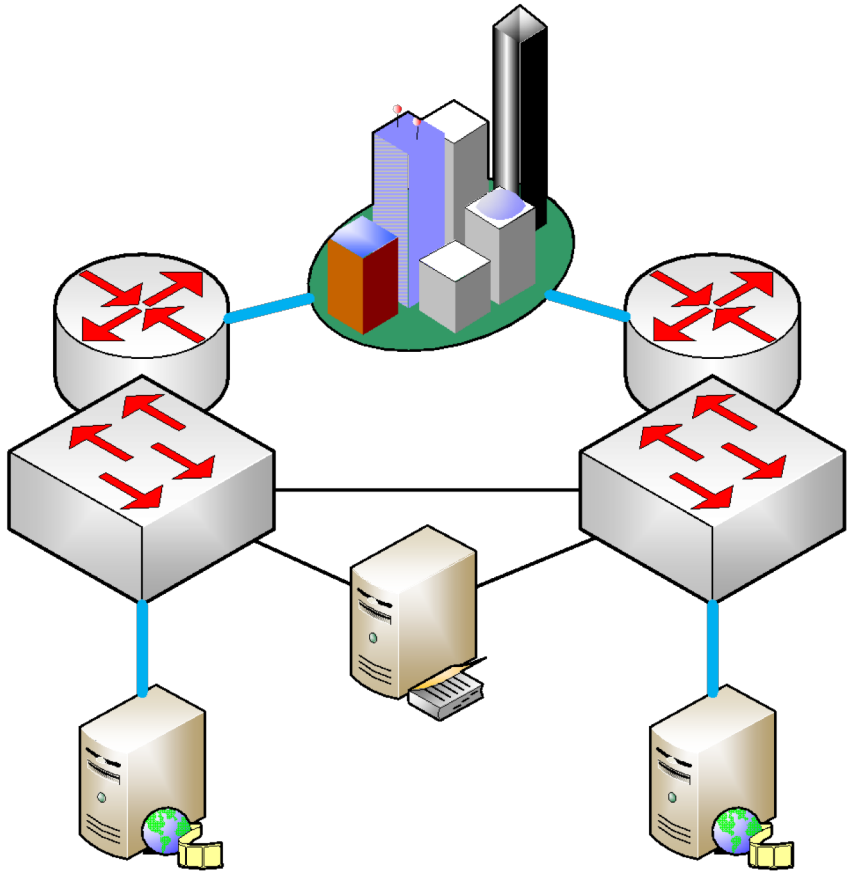
Can we fool the servers?

- Trivial idea
  - Same IP, different MAC
  - First come, first server
- Fails in practice
  - Duplicate IP detection
  - Missing ND responses
  - Core in danger



ND for the rescue

- Router
  - IPs from different networks
  - Down: Host routes to interface
- ND-Server
  - Fake ND responses
  - Rule based: who, whom, what
  - Can respond with HSRP-MACs
- Server
  - Automatically learn optimal MAC



Background

- xDSL networks
  - Carrier blocks
  - Customers need
- PARPD
  - Rule based ARP/ND responder
- Sources
  - https://lutz.donnerhacke.de/Blog/Proxy-ARP-daemon
  - https://bugs.freebsd.org/bugzilla/show_bug.cgi?id=223594