# The Scheduler meets the network

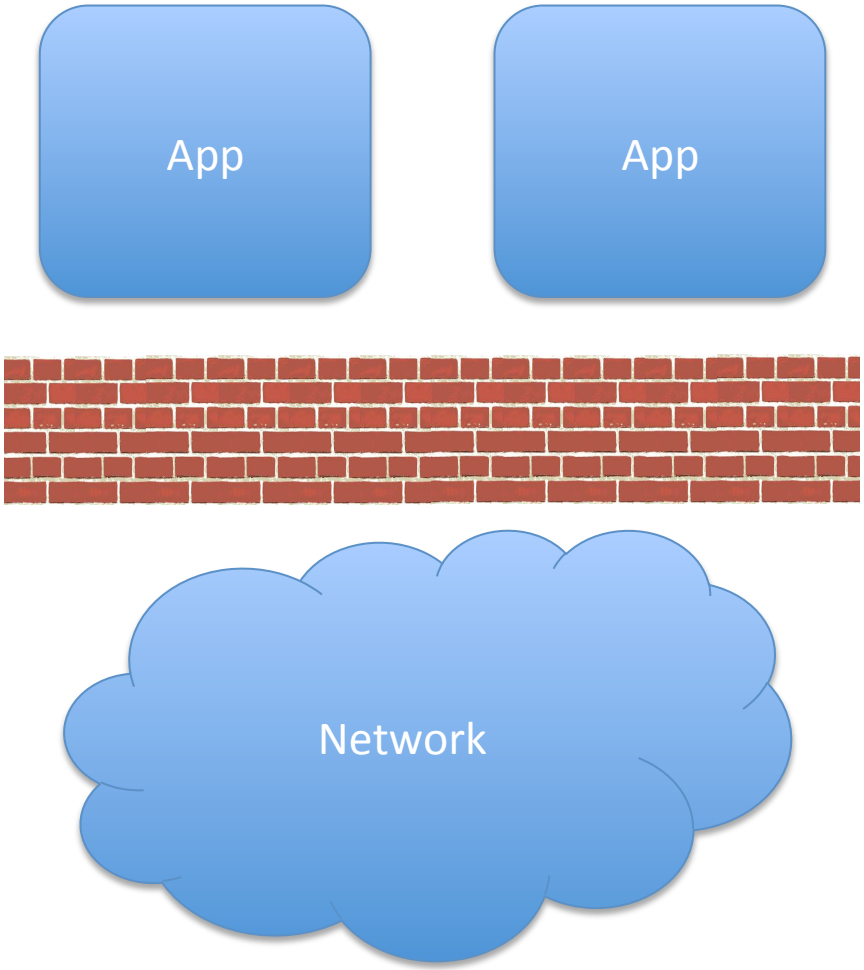
Arunabha Ghosh

Moz

# Outline

- Motivation
- Brief history of network evolution
- The network and Mesos
- Open Questions
- Conclusions

# The big divide



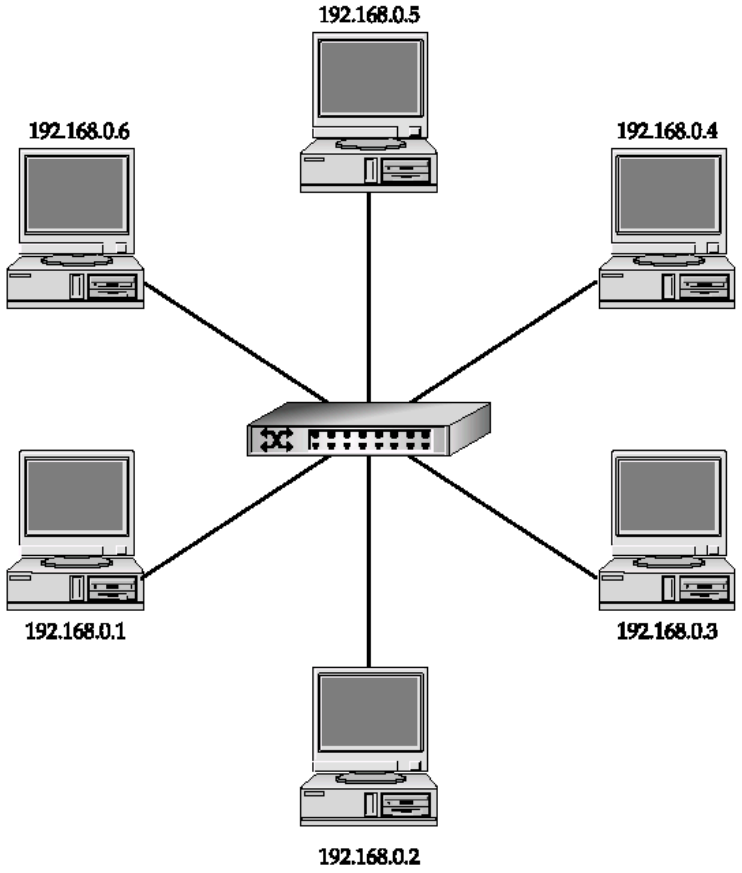
# Common Goal

- End goal for both is communication.
- Apps and networks have different, complementary views on communications
  - Apps know intent really well, but can't change the network (much)
  - The network knows the implementation of connectivity, but can't divine app intent easily.

# DISCLAIMER

# A (very) brief (and incorrect) tour of network evolution

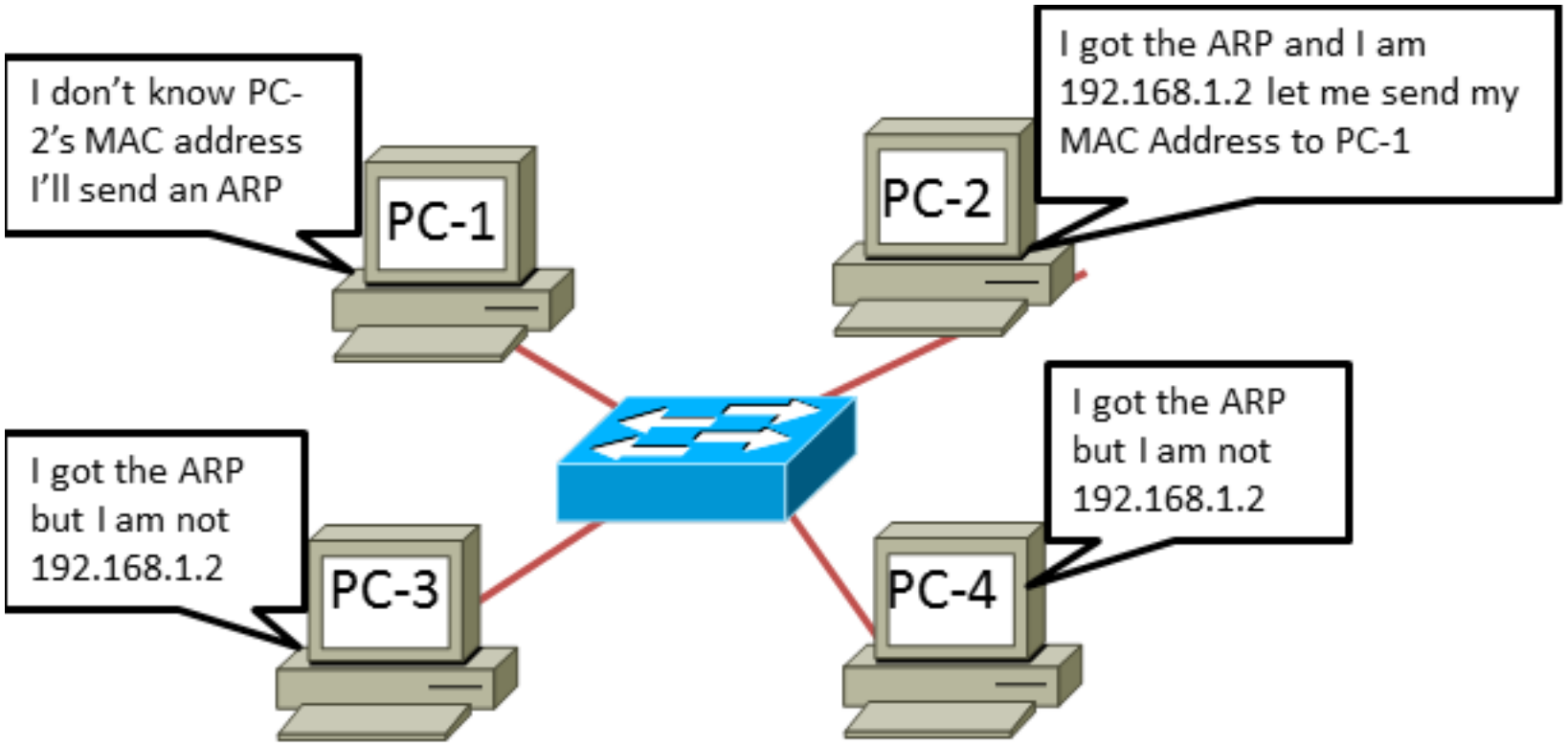
# Early LAN



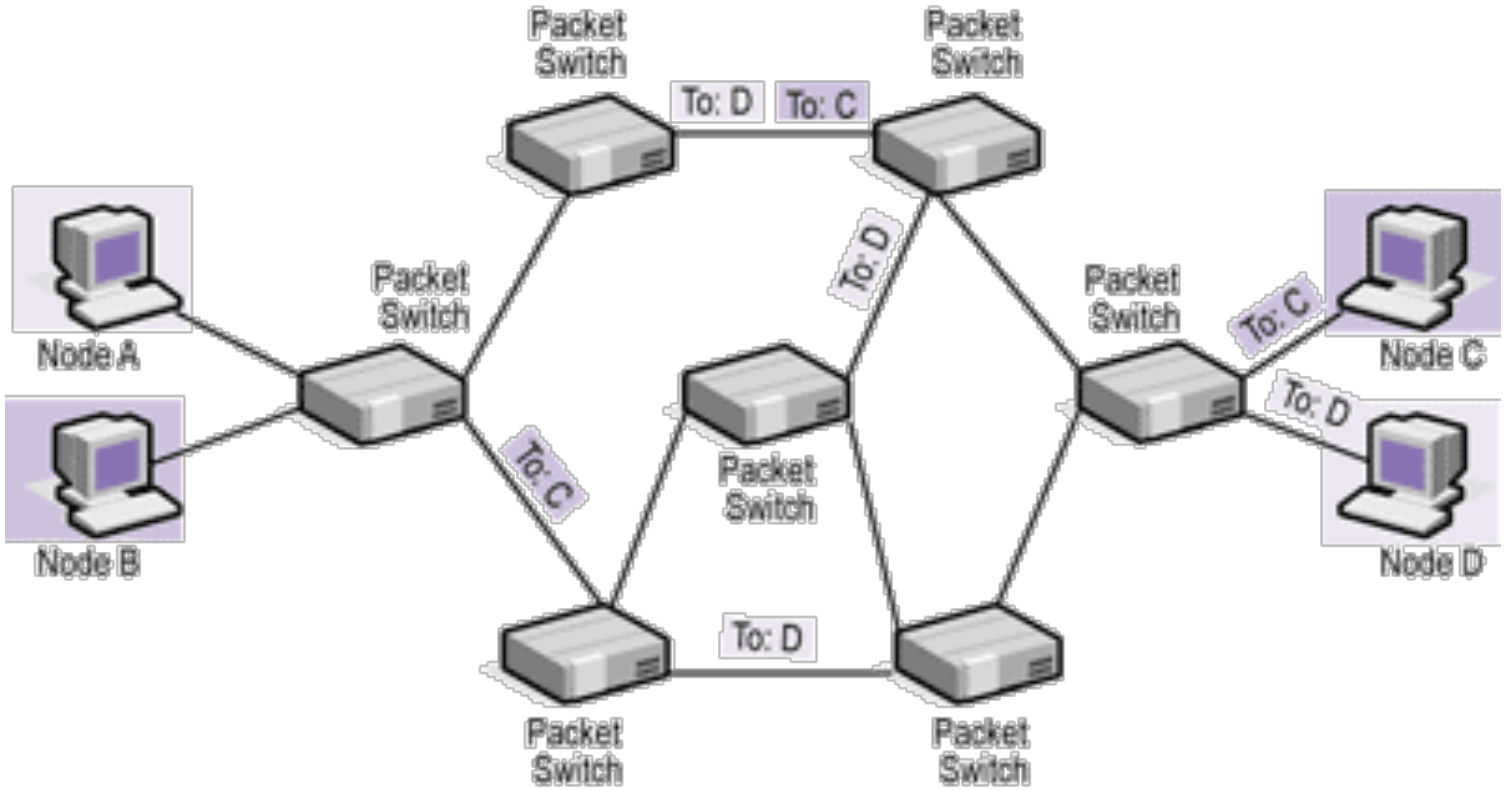

Broadcast based

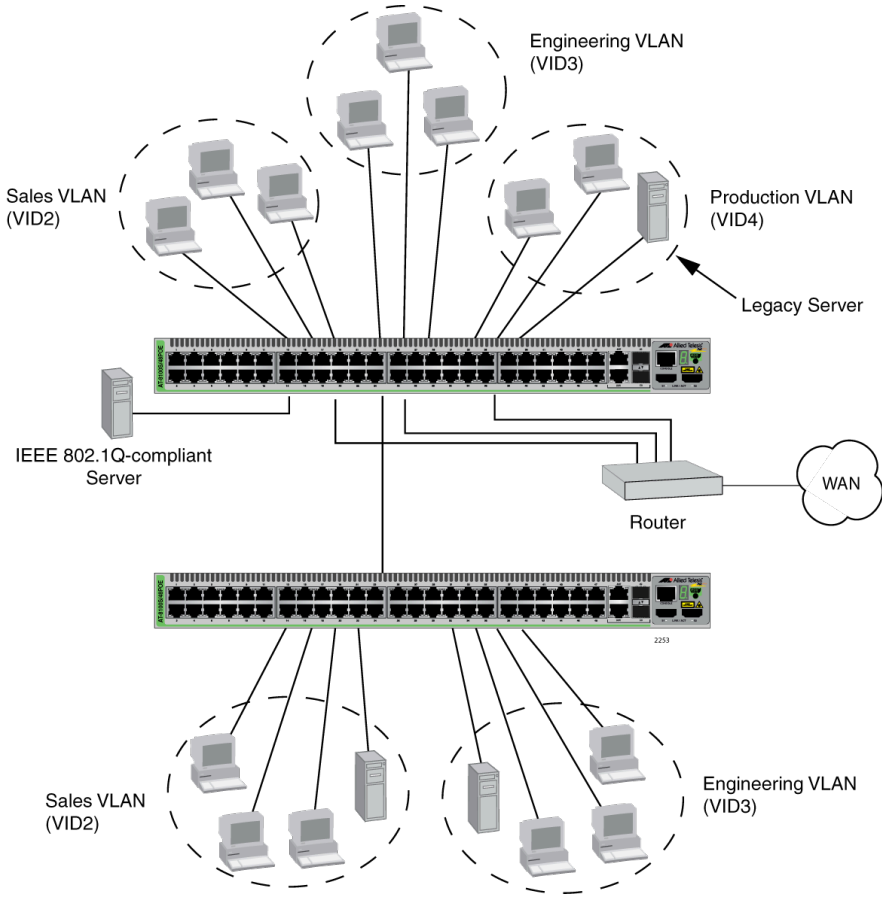
Single collision domain

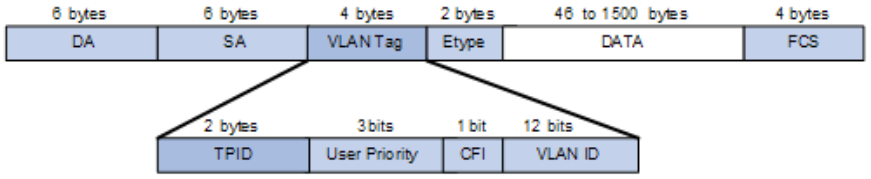
# ARP


I don't know PC-2's MAC address I'll send an ARP
PC-1
I got the ARP and I am 192.168.1.2 let me send my MAC Address to PC-1
PC-2
I got the ARP but I am not 192.168.1.2
PC-3
I got the ARP but I am not 192.168.1.2
PC-4

# To grow the network, add more switches





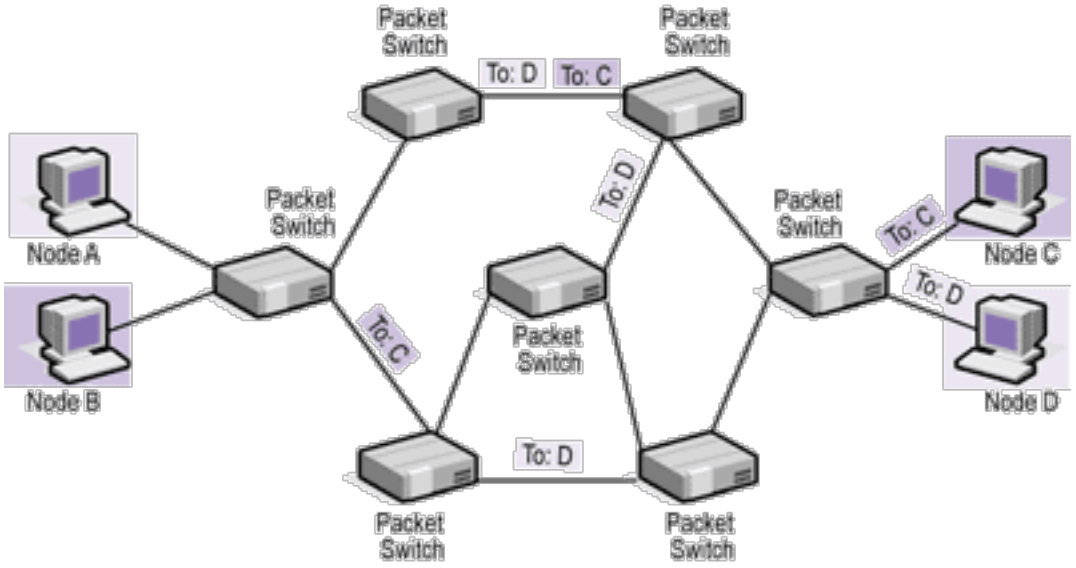




A virtual LAN or ‘VLAN’ allows us to do precisely that.

Multiple logical LANs sharing physical hardware.

Usually a VLAN is a logical unit at both L2 and L3 layers. One subnet per VLAN

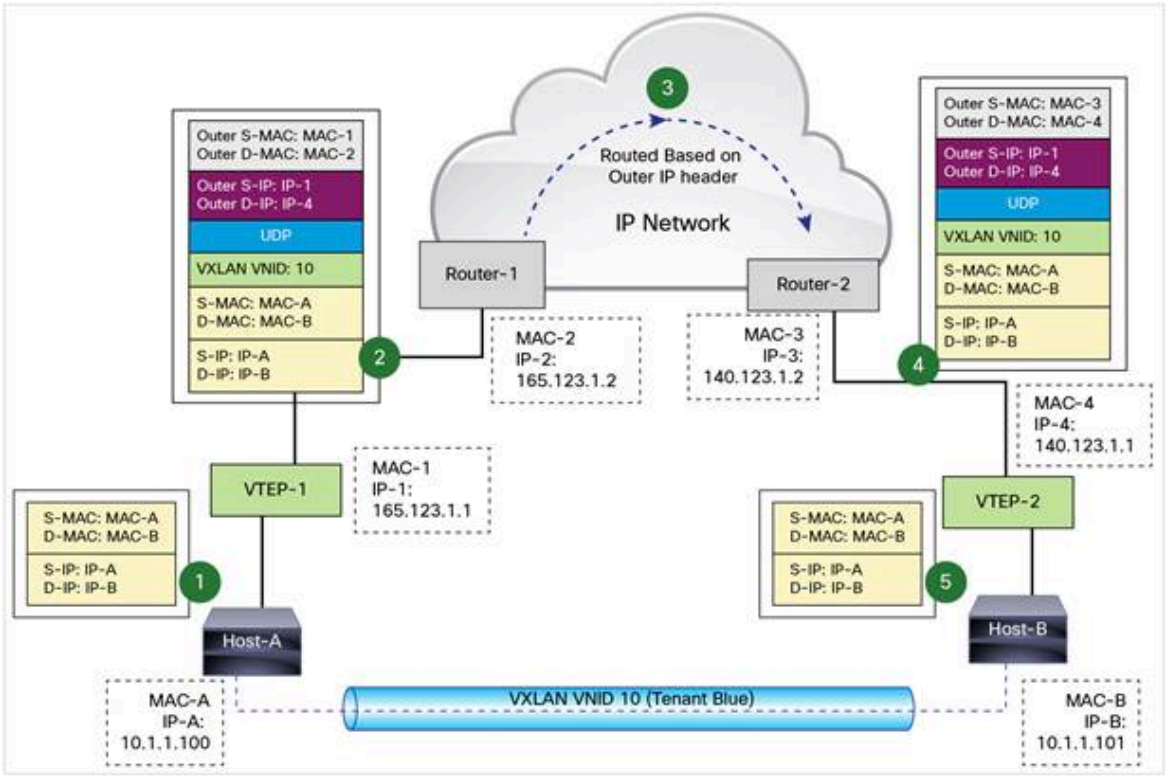
# So, life is good ?




Loops cause LANs to become sick. Only one spanning tree is used at any given time.

Lots of paths, but only one used.

Can’t have a mesh network.

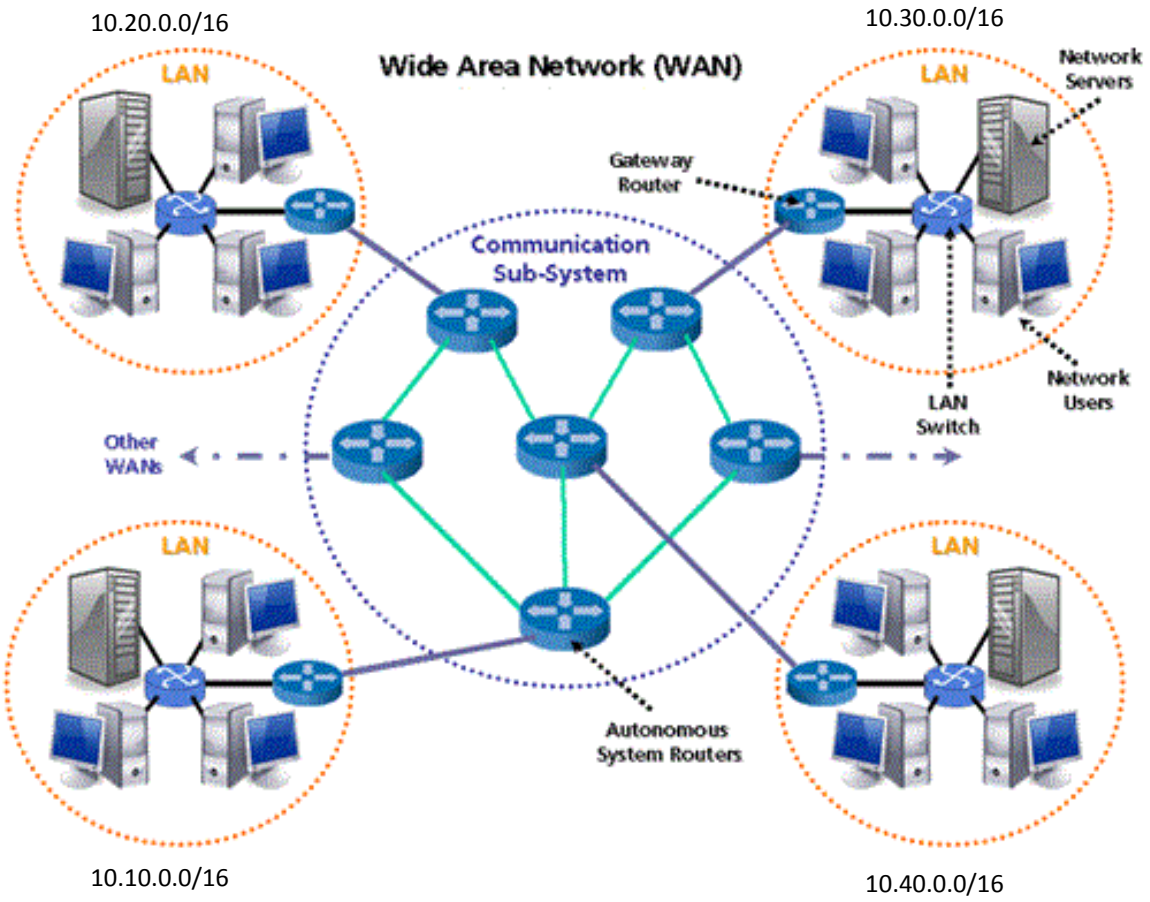



Solution.

Use L3 fabric to transport L2 frames

Many incarnations, VXLAN, NVGRE..

# WAN

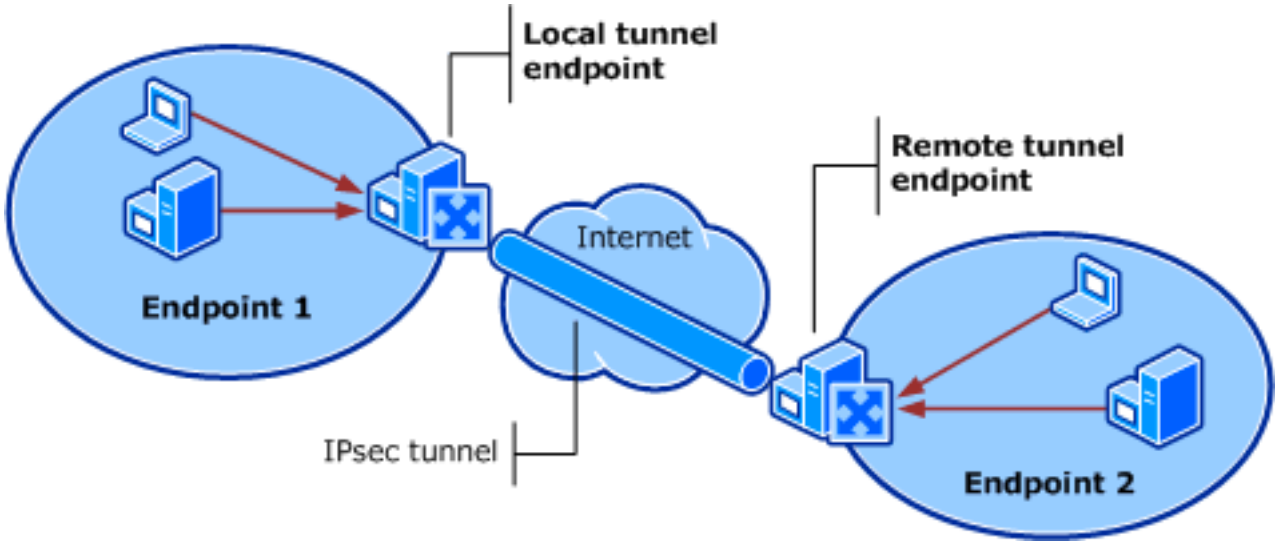



Need to connect multiple geographically disparate networks into one logical network.

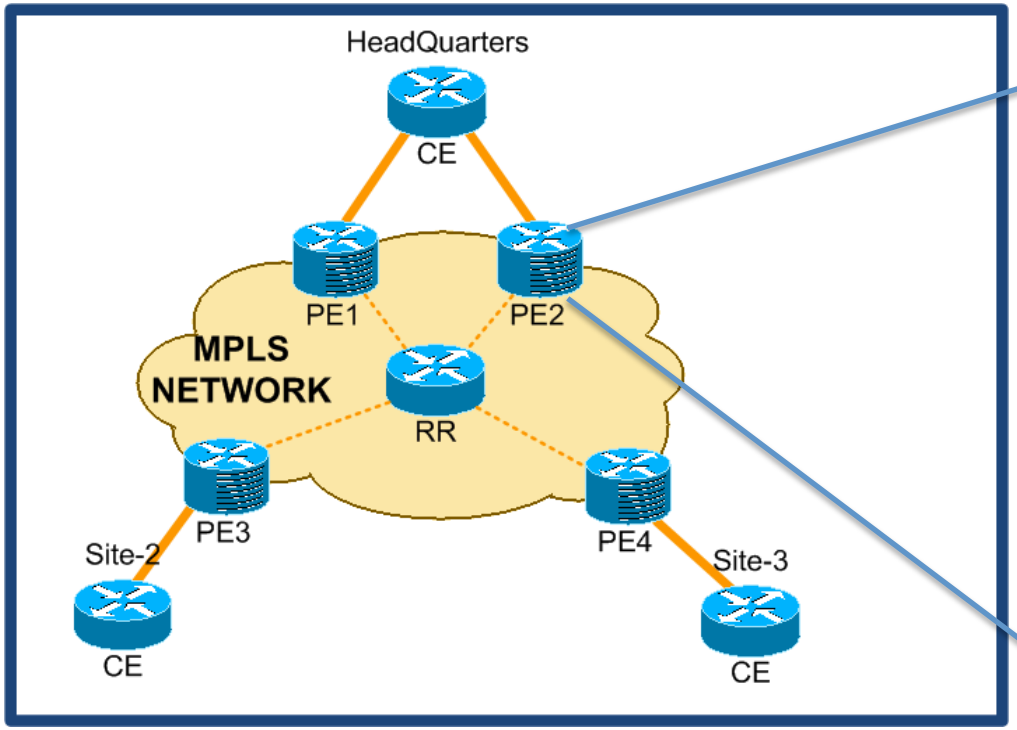
e.g. connect all company offices

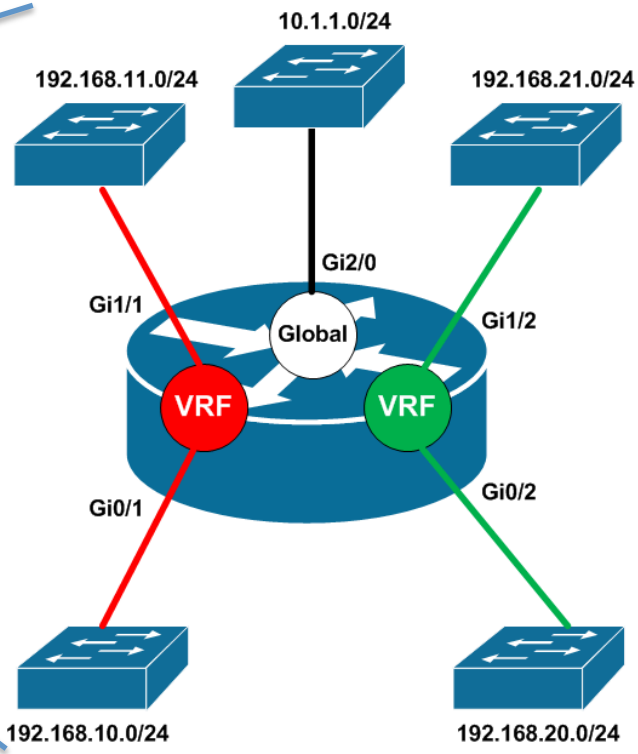
Solution

Create tunnels across sites




# Any problems ?

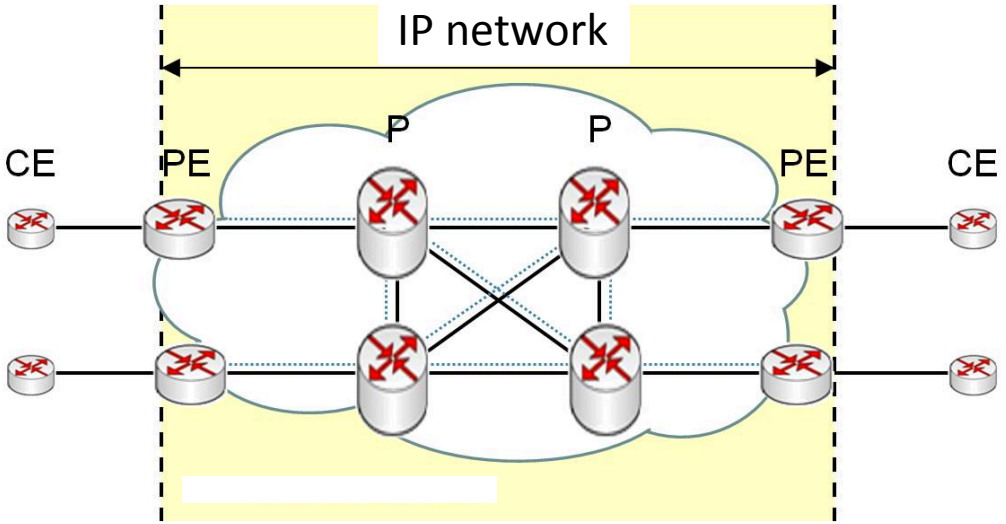


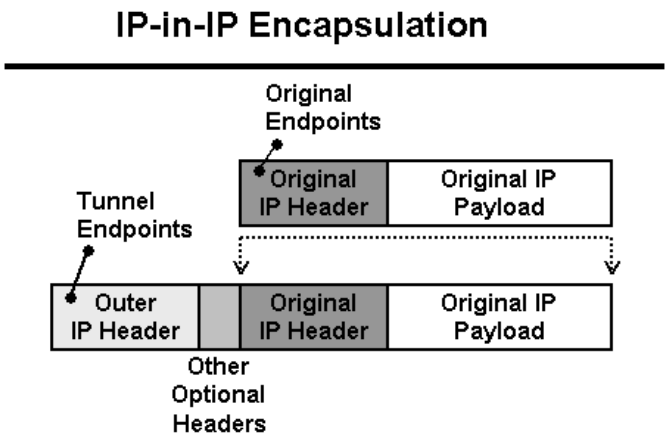




Have the service provider routers maintain a distinct route table per customer.

IPSec link replaced with MPLS

# Another implementation



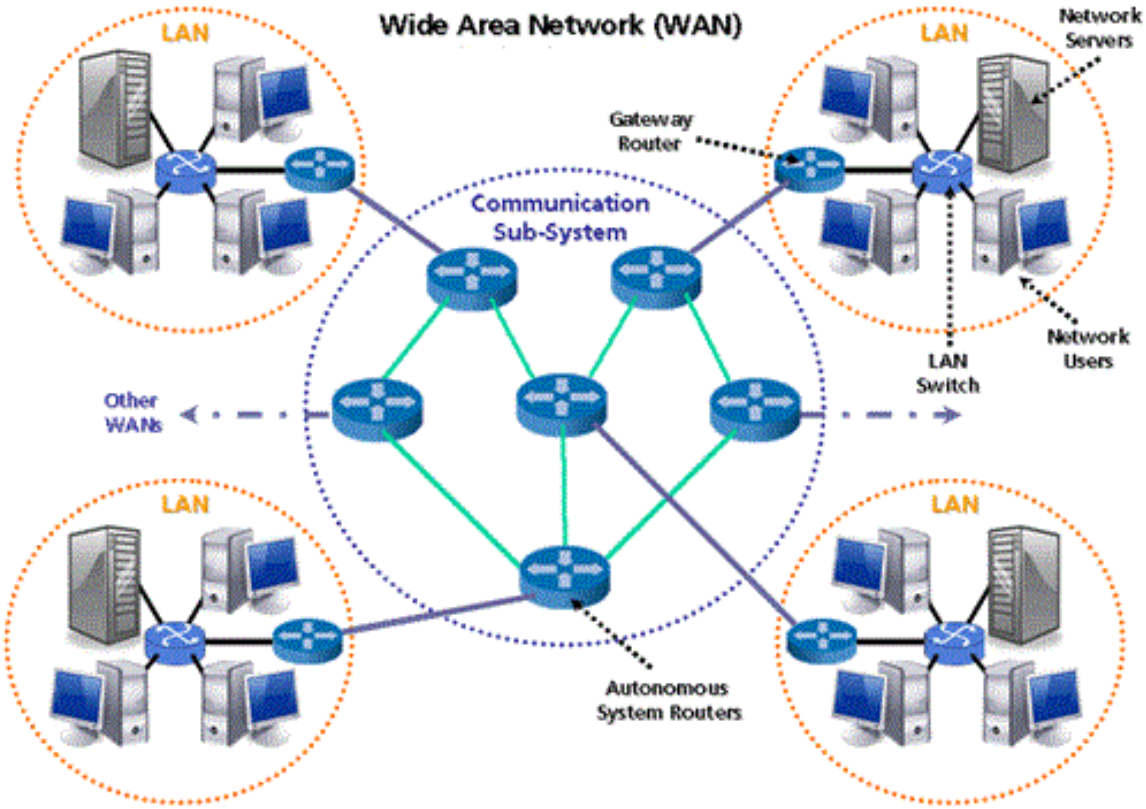
# Can we do this all in software ?

- Yes !
  - X86 is much faster now. Encapsulation and De-capsulation is (relatively) inexpensive
  - Linux networking stack is very flexible and robust

# Recap

- Networks are much more flexible than they appear to apps
  - Functionality provided by a network doesn't always need to 'physically' exist.
  - Interfaces matter, implementation can change
- Thanks to Moore's law, possible to do quite a bit in software.
- Apps know intent much better than networks

# What does this have to do with Mesos ?



So why don’t we have more dynamic networks which can be partly app specified ?

# Problem

- Network control is not scoped. All or nothing
  - Network control plane only understands network entities(IP, mac etc). Can't easily differentiate between apps, so config capabilites can't easily be scoped.
- No uniform interface
- Some network resources are scarce and need to be globally managed (external IP addresses). E.g. what if every app wanted AF4 traffic

# Thought experiment

- What if the strong separation between apps and network was weakened. What could we do ?

- Robust, fine grained access control.
  - Only allow the web servers to talk to the app servers.
- Easy discovery
  - IP per container
  - Always give this entity the same address regardless of it's physical location.
- Tailored QOS
  - Make sure traffic between app servers and database get's the highest priority.
- Many more.

# Solution

- If only we had an entity that
  - Was aware of all apps.
  - In charge of scheduling and orchestration.
  - Managed app lifecycles.
  - Had a global view of scarce resources and could arbitrate conflicting requests.
  - Present a uniform interface to apps.
  - Trusted by the operator.

# Can I do this today ?


Weave

Flynn


Netplugin


https://github.com/mesosphere/net-modules

More coming soon !

# Warning, opinions ahead

# Open questions

- What app intents should be captured ?
  - ACLs
    - Only app servers should be able to talk to database servers
  - QOS
    - Make sure traffic between DB master and slave gets high priority.
  - Reachability
    - How should the app be reachable ?
      - Fixed DNS name
      - Random IP from available pool
      - Floating IP from pool
      - Etc
  - Others ?

# Open questions

- What should the API between Mesos and the network virtualization system look like.
- Standard interfaces are critical !

# Open questions

- What information should flow between Mesos and the network virtualization infrastructure?
- What information should flow in the other direction ?

# Open questions

- How should responsibilities be divided ?
  - Master
    - Arbitrate global network resources ?
  - ~~Slave~~ Agent
    - Integrate with local network agent ?
  - Framework
  - Do we need anything else ?

# Open questions

- How should schedulable network resources be handled ?
  - Work in early stages
    - https://docs.google.com/document/d/1peqI7w9d9KZ0ZkAyF_uS9KlFbVX63VlkaRsIP4PWzOc/edit#heading=h.yvd9qbi4swb4

# Open questions

- How do we express and integrate operator policies.

# Conclusion

- Integration of network virtualization techniques with Mesos has tremendous potential.
- Possible to do a lot of cool things.
- Lot's of unanswered questions.
- Community needs to discuss and weigh in.

# Thanks !

- Feel free to reach out

agh@moz.com