

Network Troubleshooting

Colin Scott, Andreas Wundsam, Justine Sherry, Scott Shenker

Request

But don't
carrier hotels
blah-dee-
blah?



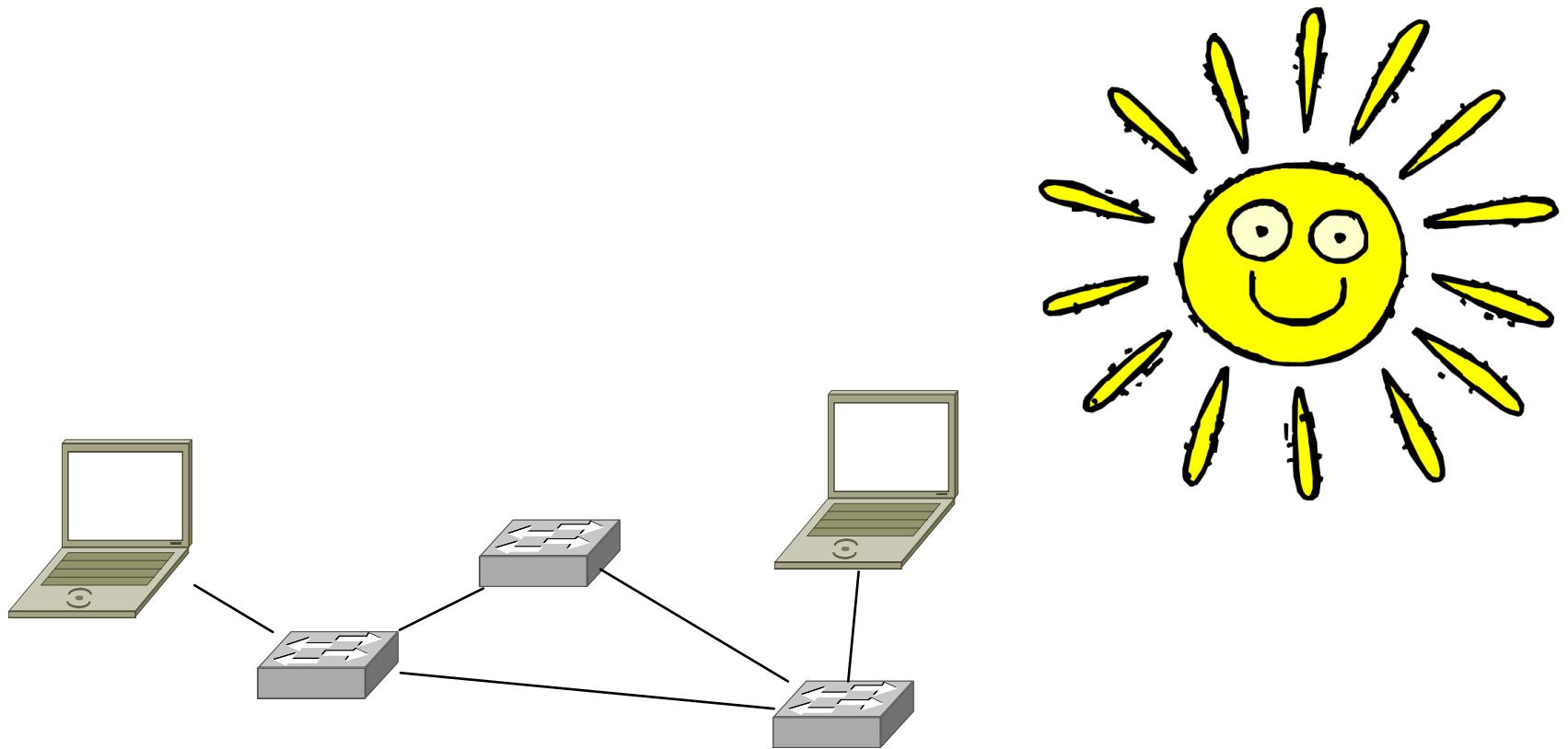
Crazy idea!
What if you
did blah-
dee-blah?

Request Invitation

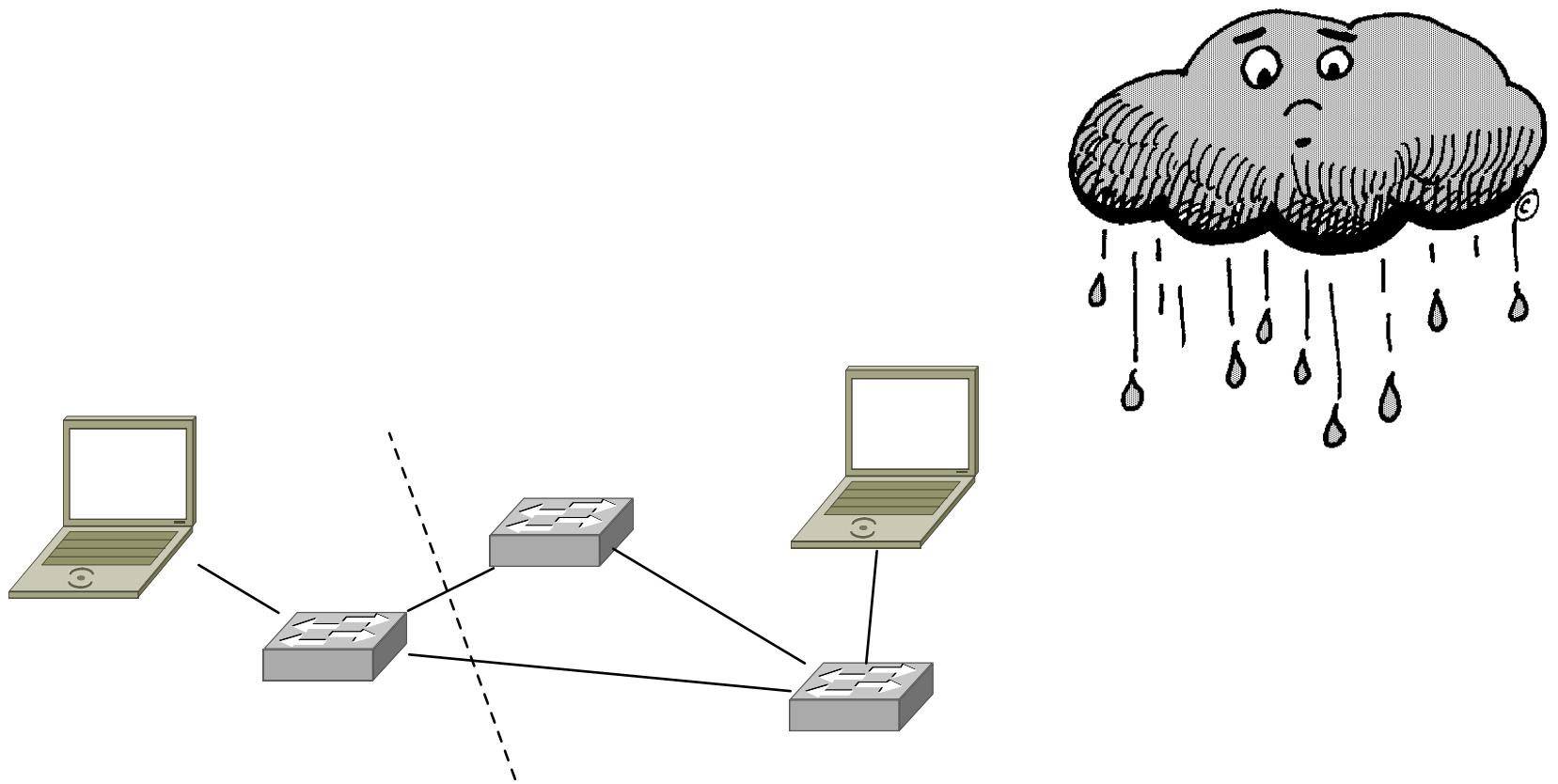
- ❑ Derail my talk!



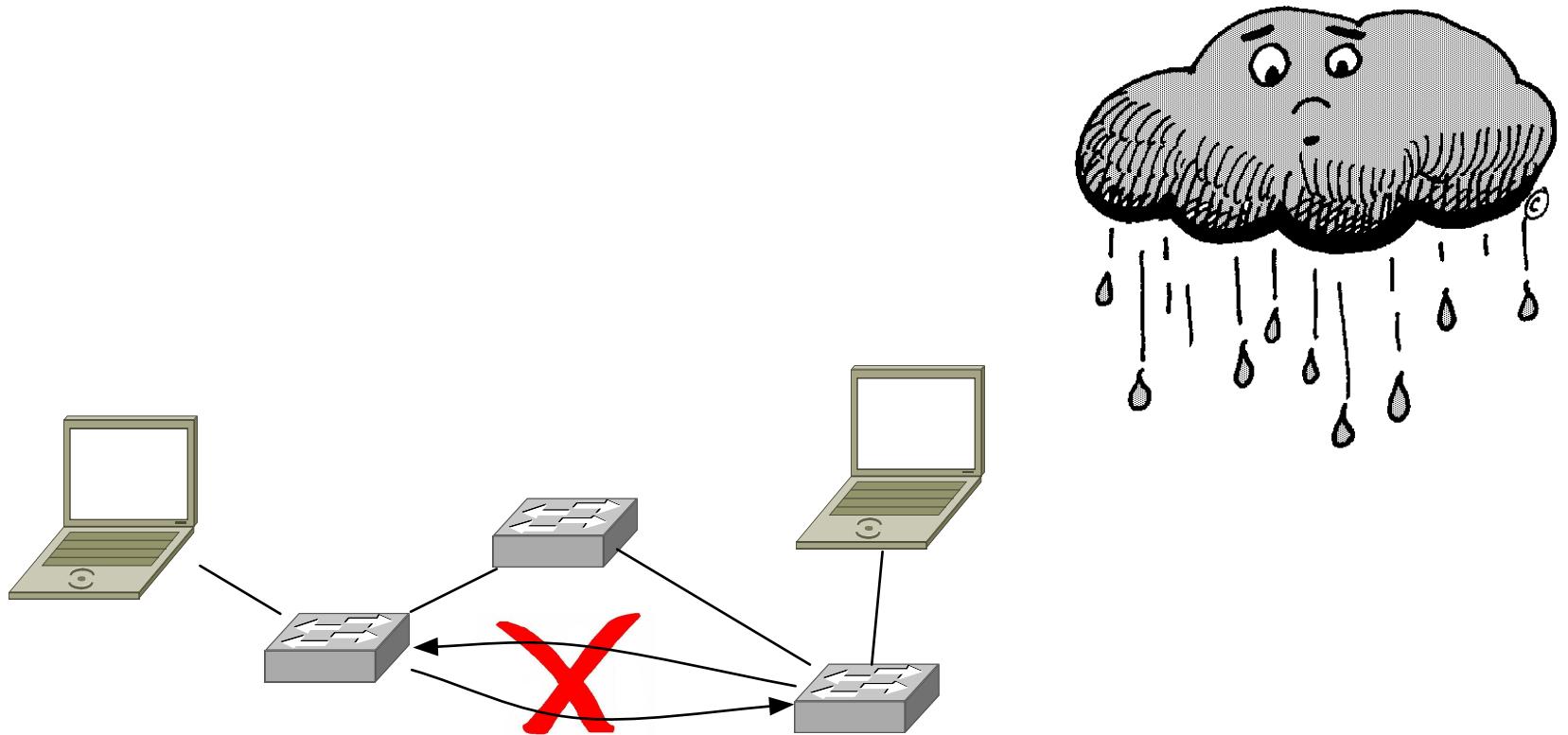
Nice Day



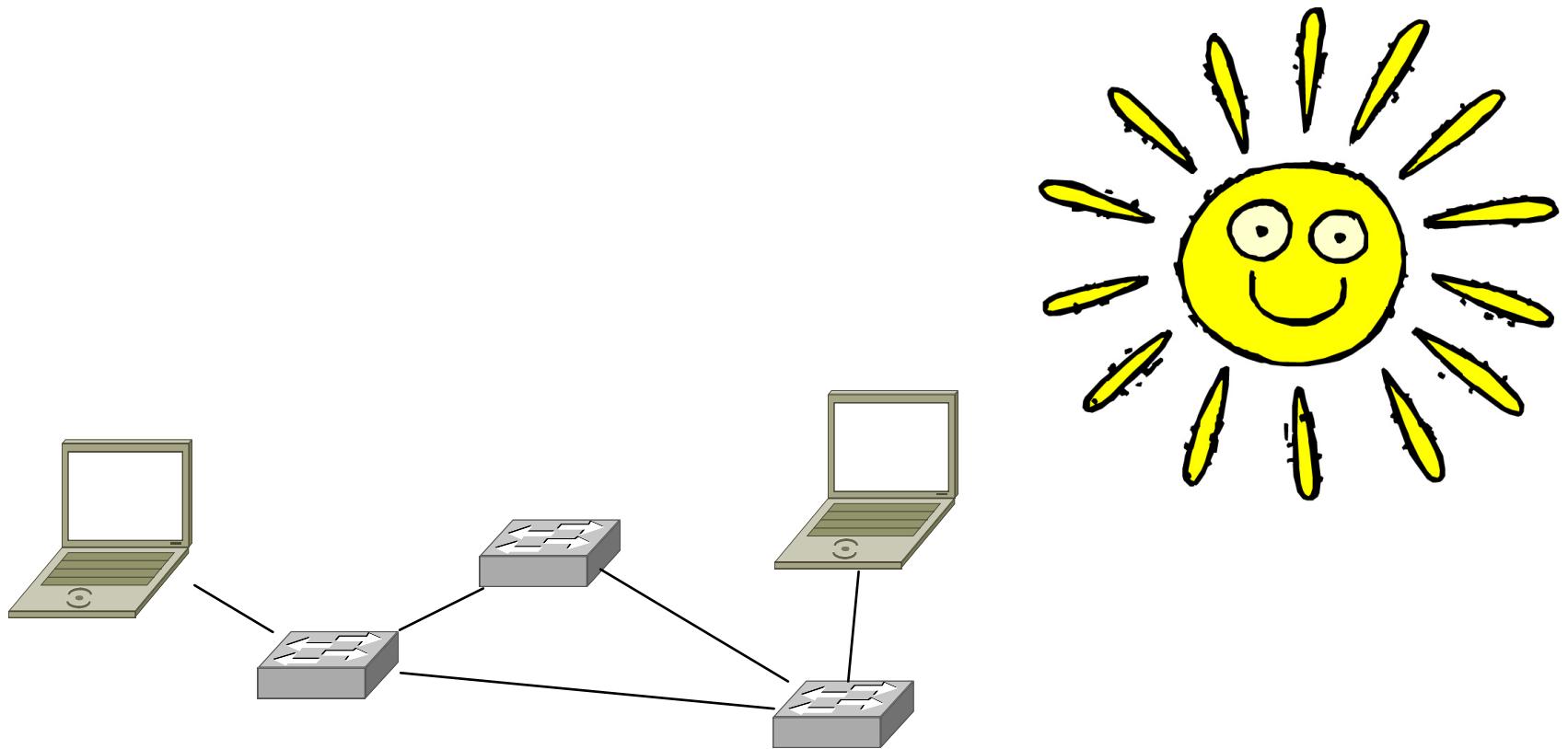
Bad Day



Bad Day



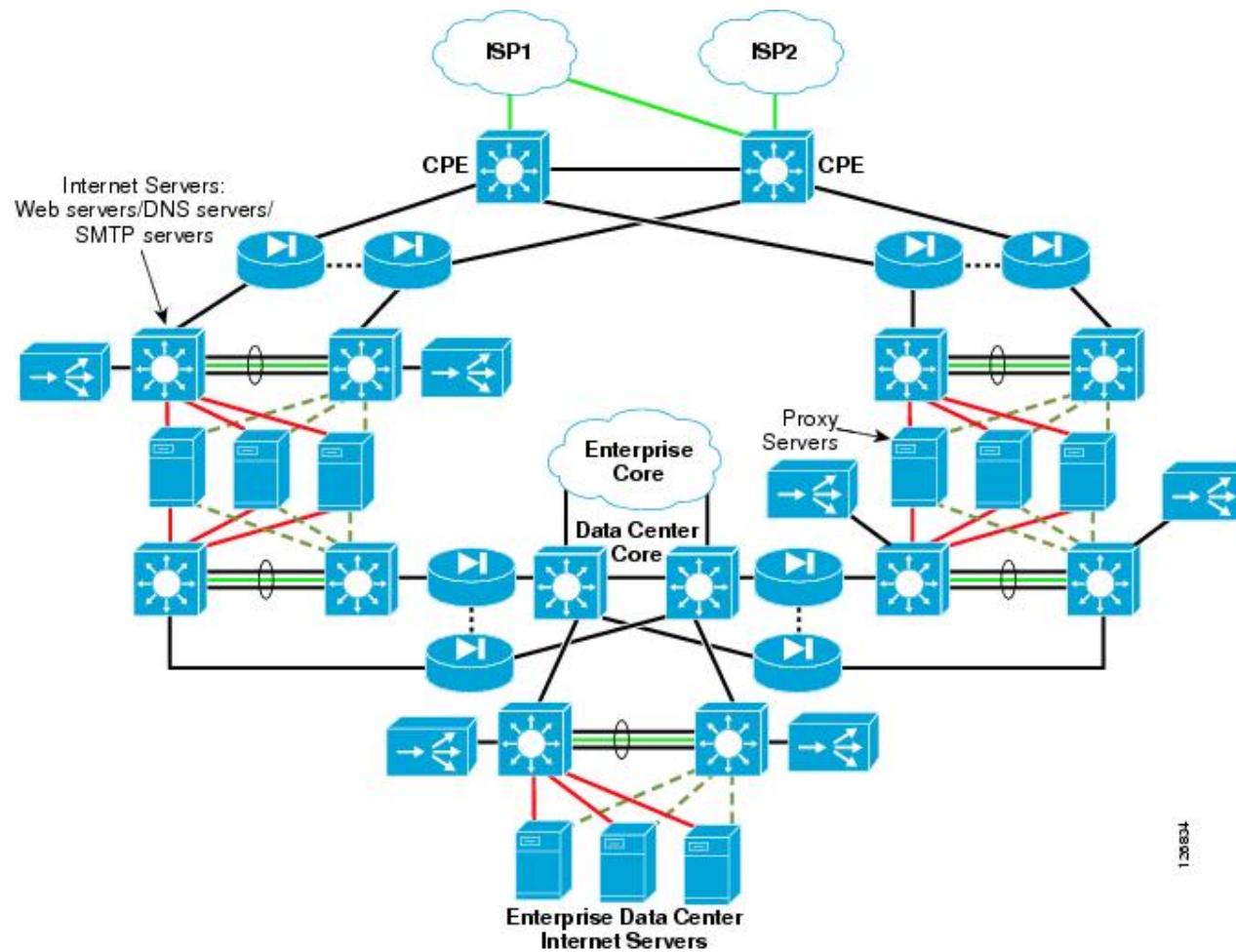
Nice Day



Reality

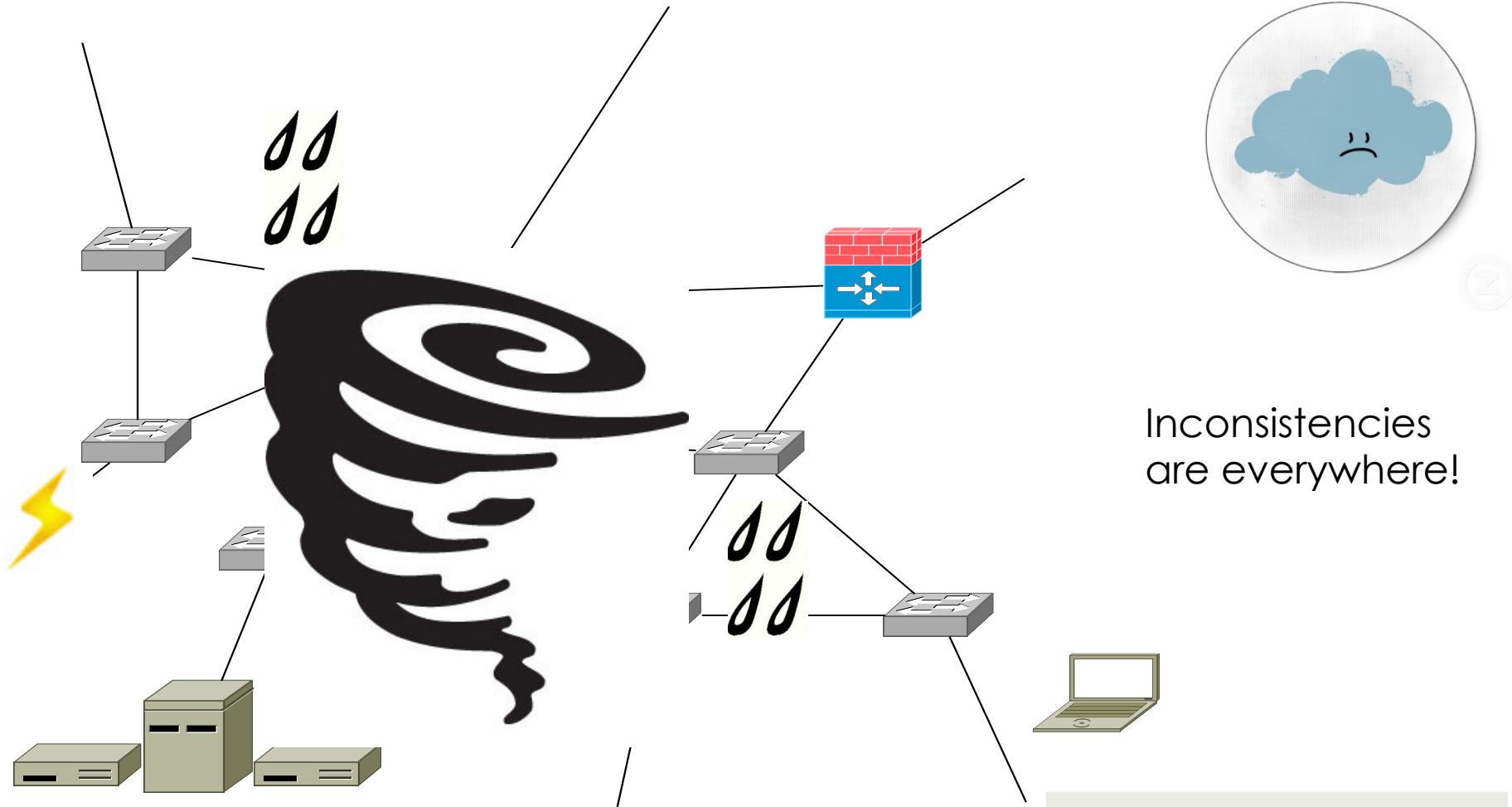


Reality



- Huge Scale
- Completely Distributed
- Highly Dynamic
- Heterogeneous
- Virtualized
- Multi-tenant
-

Reality



Research Question

- ❑ How do **pernicious** inconsistencies come about?
- ❑ How do they differ from **benign** inconsistencies?



Tools

❑ Barometer

“Invariant Checker”

Mechanism to
detect
inconsistencies



❑ Lightning rod

“Simulator”

Platform to experiment
with:

- ❑ Network configuration
- ❑ Traffic mixes
- ❑ Event orderings
- ❑ Failures
- ❑ ...



Talk Overview

- ❑ SDN as an enabler
- ❑ Barometer:
 - ❑ Formalism for invariant checking
 - ❑ Mechanism
- ❑ Lightning Rod:
 - ❑ Simulator for exercising the barometer
- ❑ Conclusion

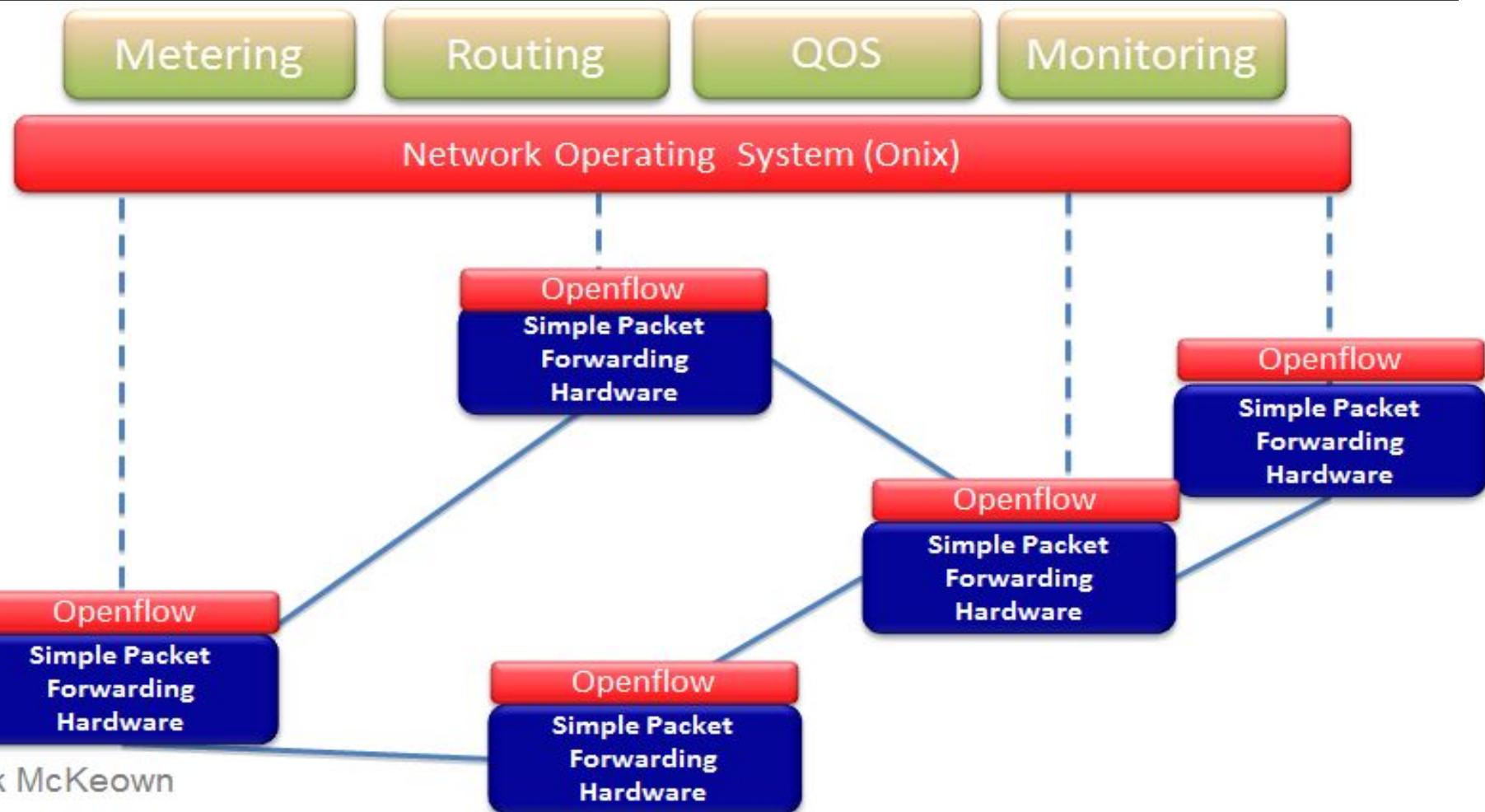
What we'll need from the network

- ❑ Policy specification
- ❑ Global view of network state

Enter Software-defined Networking



Software-defined Networking



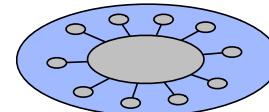
SDN Layering

Control Application

$$f(\text{view})$$

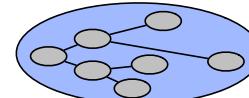
- Very simple policy specification

Virtual View

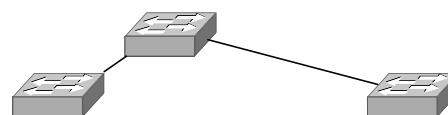


- Global view of the network state

Physical View



Physical Network



Barometer

- ❑ Inconsistency:
 - ❑ “The network doesn’t do what the control program tells it to”



Formally

Networks:

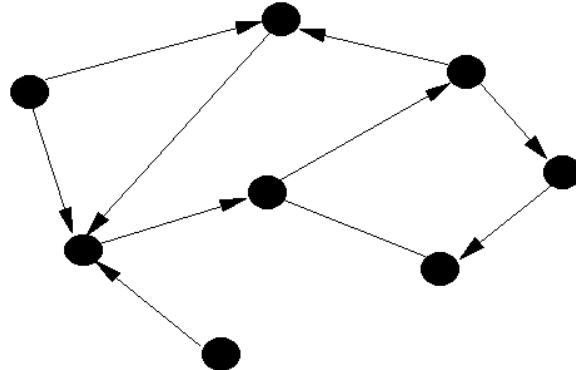
Packets:

Switches:

$$G = (V, E)$$

$$h \in \{0,1\}^L = H$$

$$T : (H \times E) \rightarrow (H \times E_\phi)$$



0110110101101
1001101110110
10011110001...



Formally

Network Forwarding:

$$\Psi = \{ \begin{array}{l} T_1 \\ \dots \\ T_n \end{array}$$

Network Traversal:

$$\Phi^k(h, e) = \Psi(\dots\Psi(\Psi(h, e))\dots)$$

* Peyman et al., Header Space Analysis, NSDI '12

Formally

Network Behavior:

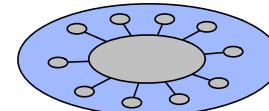
$$\Omega(h, e) = \Phi^\infty(h, e)$$

SDN Layering

Control Application

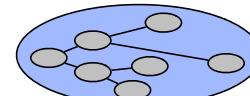
$$f(\text{view})$$

Virtual View



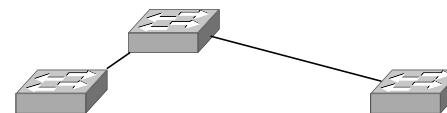
$$G^{\text{Virtual}}$$

Physical View



$$G^{\text{Logical}}$$

Physical Network



$$G^{\text{physical}}$$

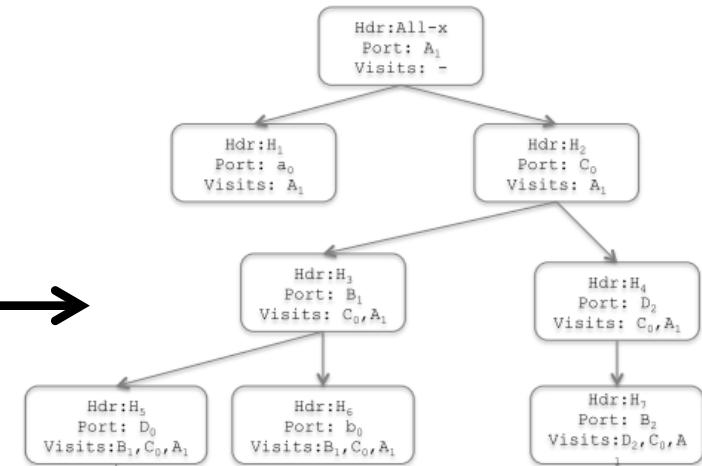
Formal Invariant

$$\Omega^{\text{virtual}} \sim \Omega^{\text{physical}}$$

“All paths in the virtual network should have a corresponding path in the physical network”

Checking Correspondence

- Take snapshot of $G^{virtual}, G^{physical}$
- Compute $\Psi^{virtual}, \Psi^{physical}$
- Compute $\Omega^{virtual}, \Omega^{physical}$
 - For each access link
 - Insert symbolic packet x^L
 - Iteratively apply Ψ
- Check $\Omega^{virtual} \sim \Omega^{physical}$



Barometer Wrap Up

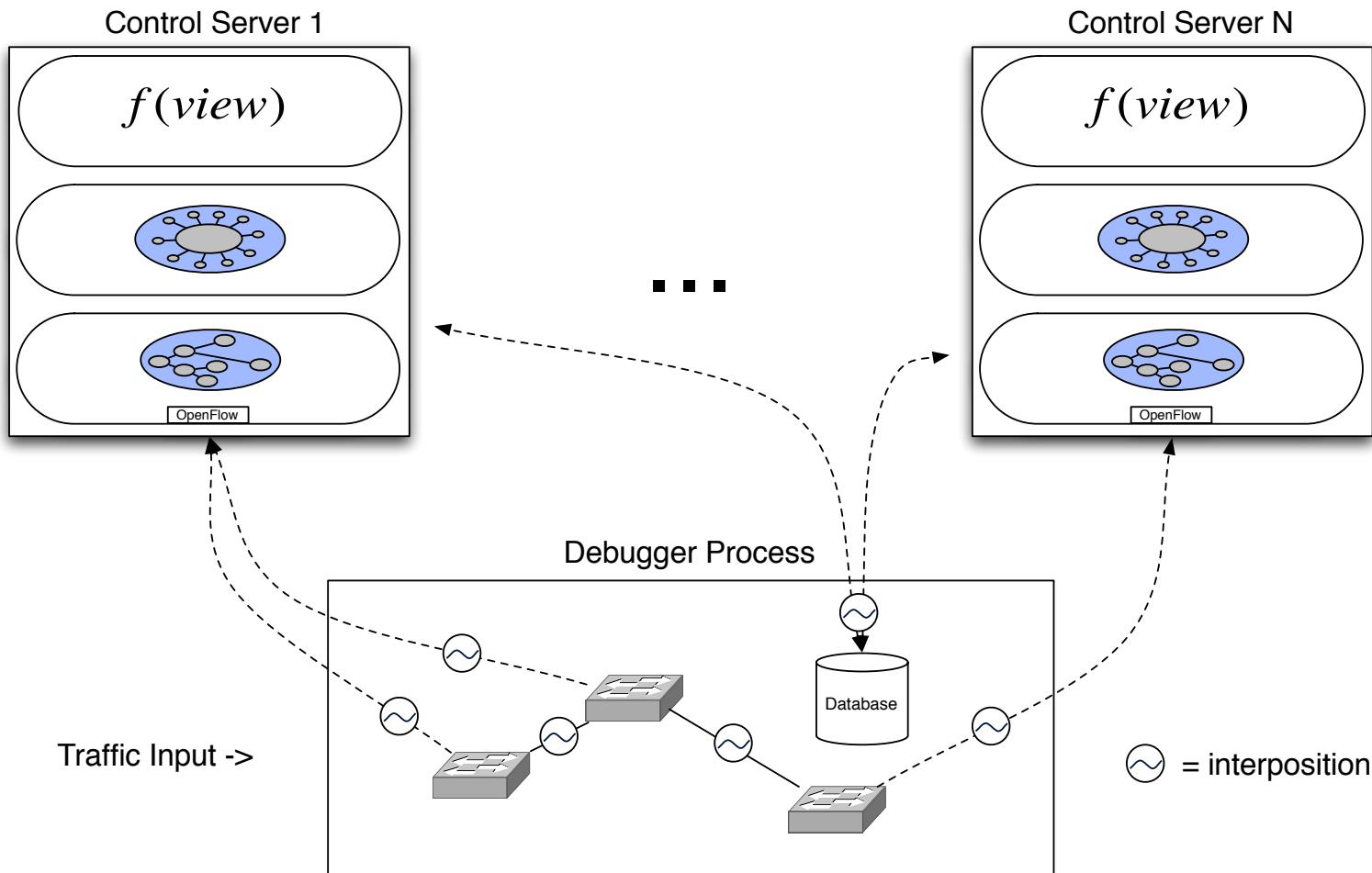
- ❑ Mechanism to detect inconsistencies *at a particular point in time.*
- ❑ Now what?

Lightning Rod

- We want to explore system executions
- In particular, we're looking for corner cases

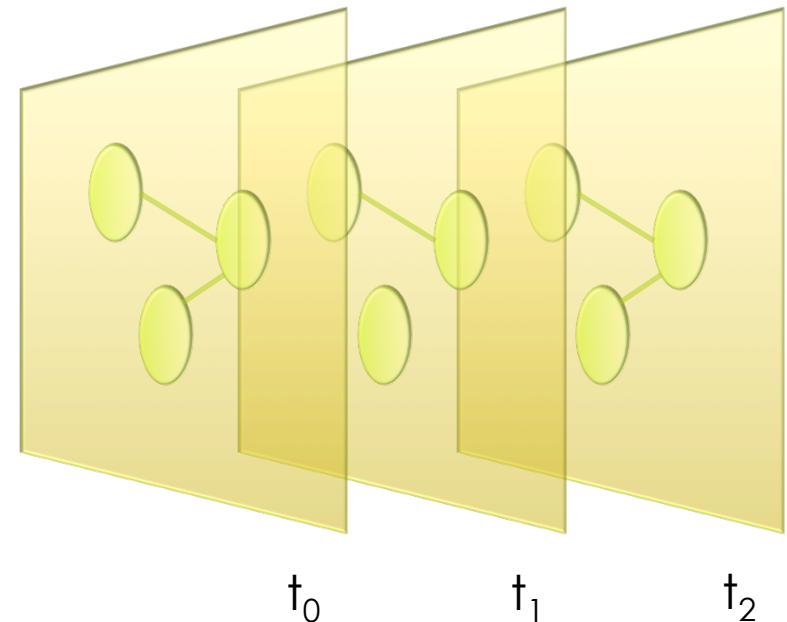


Lightning Rod



Using the Lightning Rod

- ❑ Insert input traffic. Either:
 - ❑ Fuzzed
 - ❑ Production traces
- ❑ Manipulate events:
 - ❑ Cause delays, drops
 - ❑ Induce failures
 - ❑ Filter out irrelevant events
- ❑ Track inconsistencies



Putting it all together

- Goal:
 - Understand inconsistencies in the network
- Barometer: “Invariant checker”
 - Infer inconsistencies at any point in time
- Lightning Rod: “Simulator”
 - Explore system executions
- Enabled by observation: SDN provides layering

Future Work

- ❑ Evaluate: run debugger on third-party controllers
 - ❑ Multi-tenant virtualized network
 - ❑ Scale-out BGP router
 - ❑ Ethane
 - ❑ ...
- ❑ Gather library of corner-case traces
- ❑ Get people to use it!

Thanks!

