OpenFlow and SDN: hype, useful tools or panacea?

Ivan Pepelnjak (ip@ioshints.info)
Chief Technology Advisor

NIL Data Communications

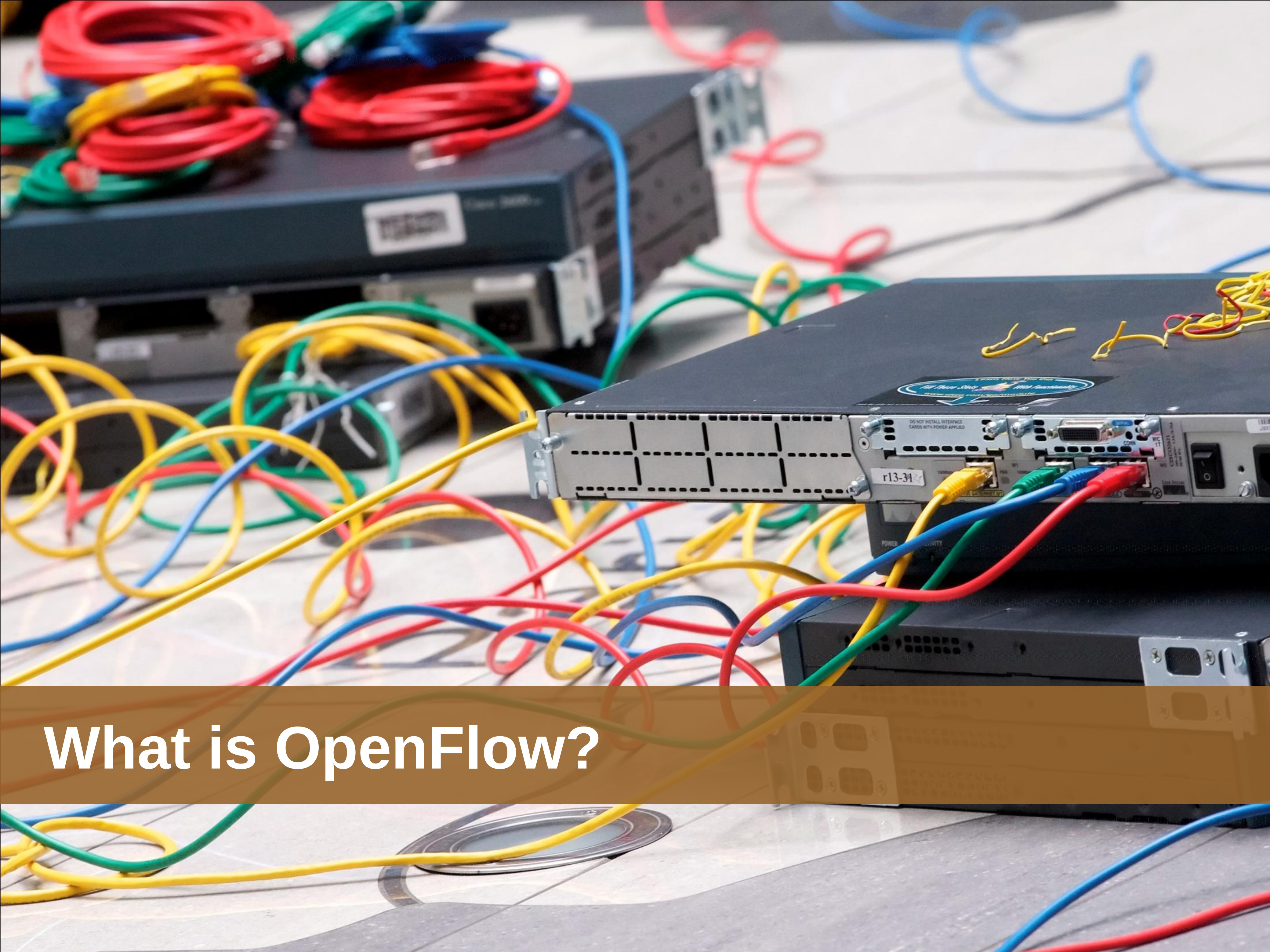
Who is Ivan Pepelnjak (@ioshints)

- Networking engineer since 1985
- Technical director, later Chief Technology Advisor @ NIL Data Communications
- Consultant, blogger (blog.ioshints.info), book and webinar author
- Currently teaching "Scalable Web Application Design" at University of Ljubljana

Focus:

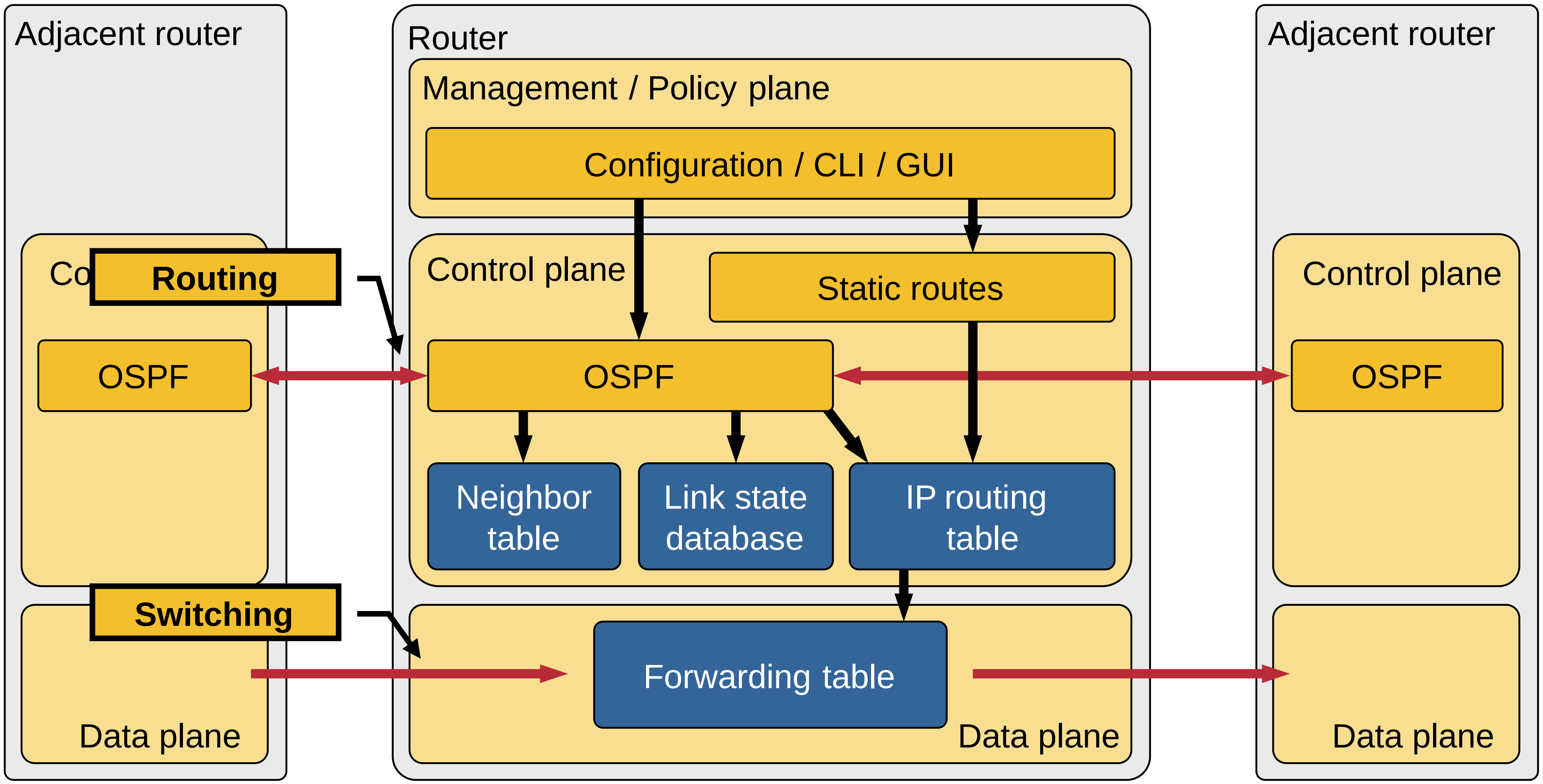
- Large-scale data centers and network virtualization
- Networking solutions for cloud computing
- Scalable application design
- Core IP routing/MPLS, IPv6, VPN

More @ ipSpace.net/About and ipSpace.net/Webinars

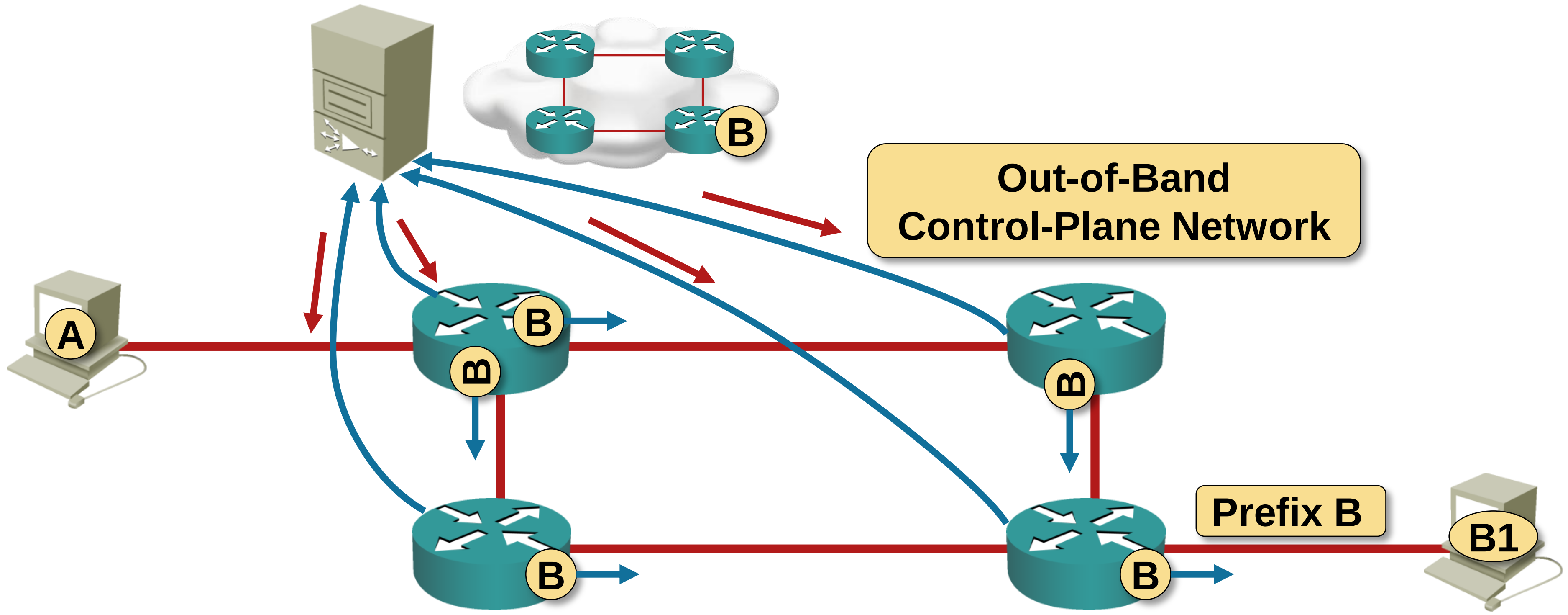


What is OpenFlow?

Management, Control and Data Planes



OpenFlow = Control / Data Plane Separation



Basic principles:

- Control / Management plane in a dedicated *controller*
- Networking devices perform forwarding and maintenance functions
- IP / SSL connectivity between controller and OpenFlow switch
- OpenFlow = Forwarding table (TCAM) download protocol

OpenFlow Protocol Details

S-Port	D-Port	L4P	ToS	D-IP	S-IP	V	ET	PCP	VLAN	S-MAC	D-MAC

Message types:

- Configuration
- Feature requests
- Flow/Port/Table modifications
- Statistics
- Barriers (~ transactions)
- Packet In/Out

Flow classifiers:

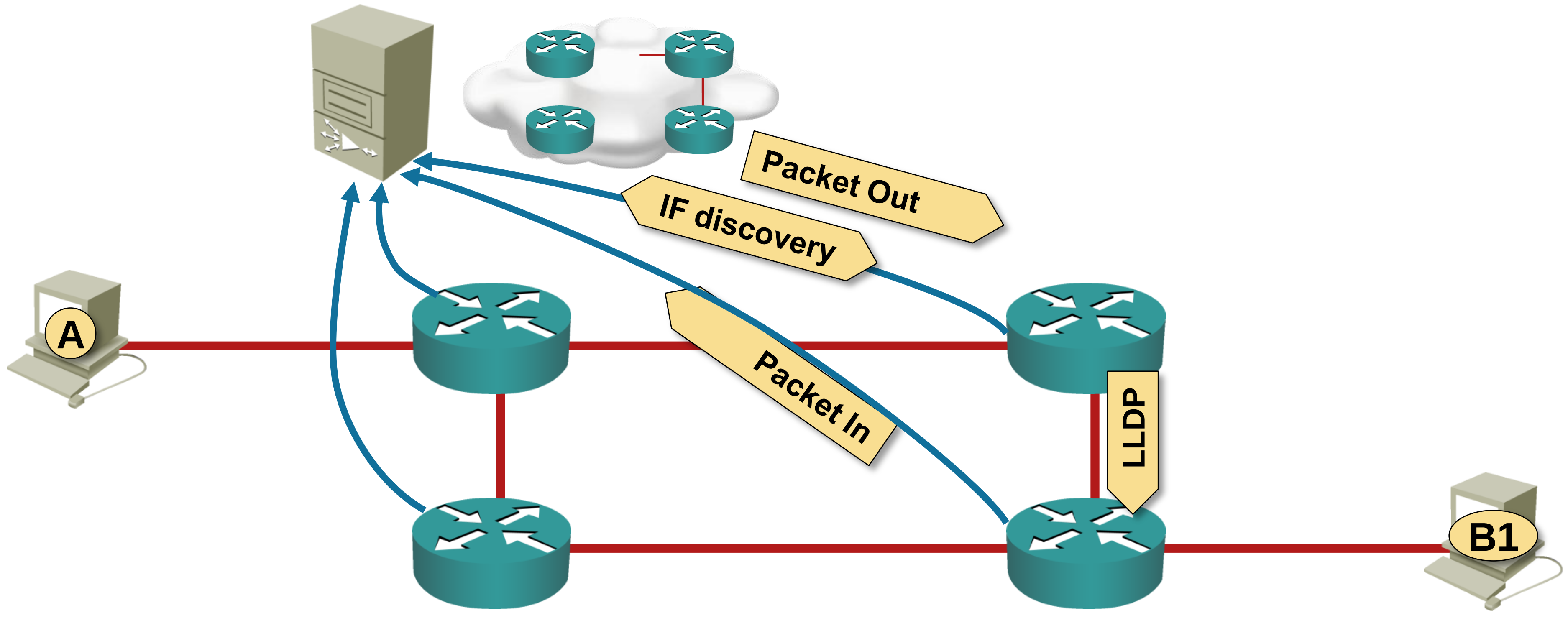
- Any combination of supported packet header fields
- IP and MAC address wildcards
- Other fields (OF 1.2, experimental)
- IPv6 extension headers (OF 1.3)

Flow actions:

- Header rewrites (ex: NAT/SLB)
- Push/pop VLAN/PBB/MPLS tags (OF 1.2)
- Output to a port
- Send to *normal processing*
- Send to controller

Hardware support usually limited to subset of OpenFlow 1.0 with extensions

OpenFlow Topology Discovery



- Controller builds the network model as devices connect to it
- OpenFlow control packets used for interface
- *Packet Out* message used to send a packet through an interface
- *Packet In* message used by the switch when it receives unknown packet

This Does Not Make Sense (Unless You're Google)

Claim: OpenFlow will replace existing routing protocols

Routing Protocol Drawbacks

- Loosely coupled
- Eventual consistency
- Destination-only
- Not load-aware
- Resistant to change and control

Routing Protocols Benefits

- Reliable
- Proven
- Deterministic
- Self-Healing
- Autonomous
- Scalable

An SDN/OpenFlow controller must

- Reinvent all the wheels (scalability, resilience, reliability, auto-discovery, fast convergence, fast control loops)
- Provide added value

It's a Déjà-Vu All Over Again (RFC 1925, sect 2.11)

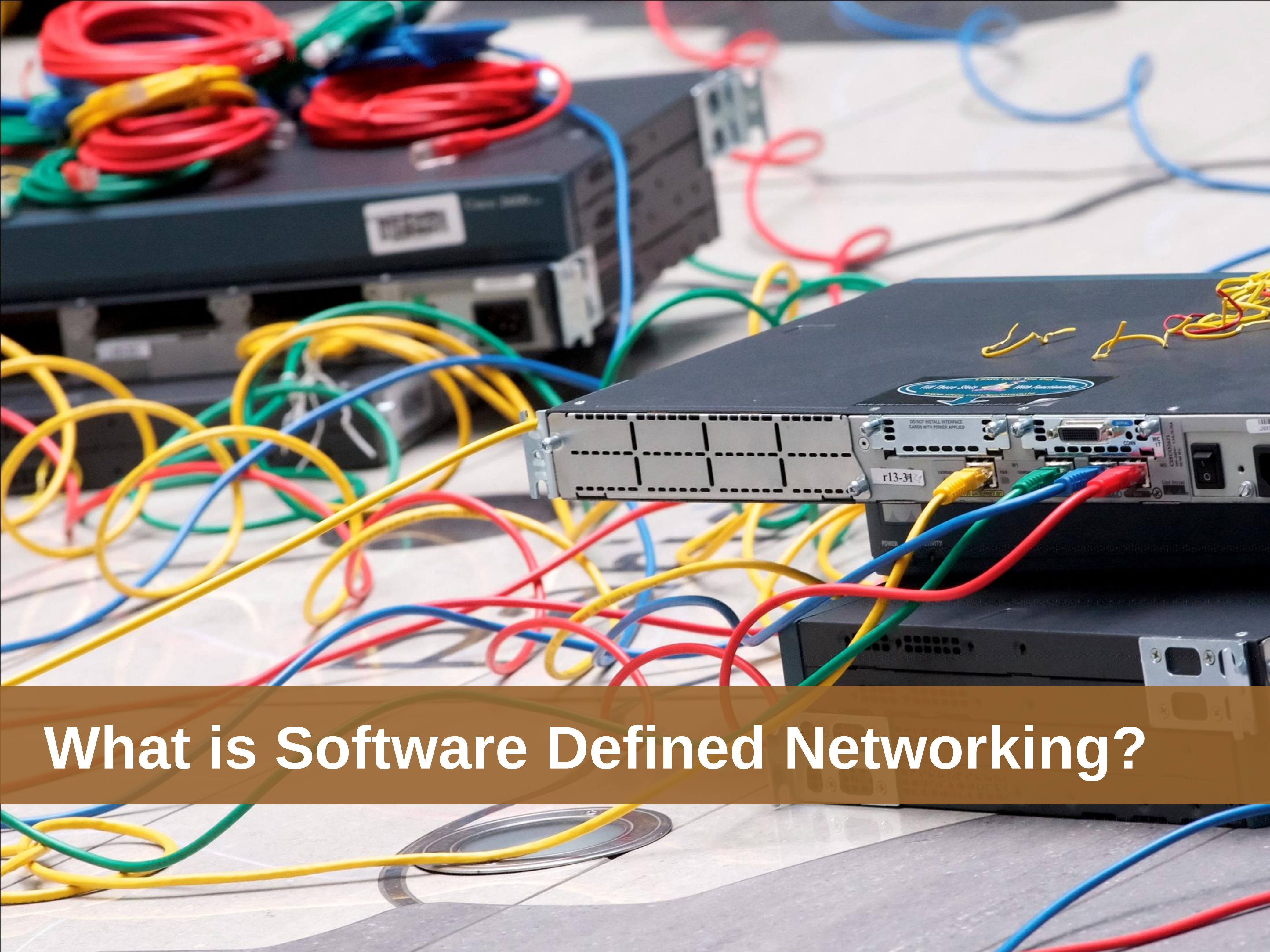
Do you still remember ...

- Frame Relay and ATM networks
- SONET/SDH
- MPLS-TP
- ForCES

The problems are always the same:

- Forwarding state abstraction / scalability
- Distributed network resilience with centralized control plane
- Fast feedback loops
- Fast convergence (FRR, PIC)
- Linecard protocols (BFD, LACP, LLDP ...)

The important difference this time: customer pressure



What is Software Defined Networking?

What is SDN?

In the SDN architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications.

Open Networking Foundation white paper

Let’s call whatever we can ship today SDN

Vendor X

SDN is the magic buzzword that will bring us VC funding

Startup Y

Dear vendor, an API does not SDN make

SDN Advantages / Perfect Use Cases

Solving hard problems that require centralized view or synchronization

Things we do well:

- Destination-only hop-by-hop L3 forwarding

Things we don’t do so well:

- Layer-2 forwarding (spanning tree limitations)
- Optimal traffic engineering (MPLS-TE) – the knapsack problem
- Routing of elephant flows

Things we don’t do at all:

- Synchronized distributed policies (security, QoS ...)
- QoS- or load-based forwarding adaptations
- L3/L4-based or source+destination-based forwarding (policy-based routing)
- Insertion of security features in the forwarding path

Best approach: combine SDN/OpenFlow with traditional mechanisms

More @ http://blog.ioshints.info/2011/11/openflow-enterprise-use-cases.html , http://networkheresy.wordpress.com/2011/11/17/is-openflowsdn-good-at-forwarding/

OpenFlow/SDN Deployment Models

Native OpenFlow

- Works well at the edge (single set of uplinks)
- Too many complications at the core (OOB management, fast failure detection ...)

OpenFlow with vendor-specific extensions

- Link bundling
- Load balancing
- Linecard functionality (LLDP, LACP, BFD ...)

Ships in the night

- OpenFlow in parallel with traditional forwarding
- Some ports / VLANs dedicated to OpenFlow
- Fallback from OpenFlow to *normal*
- Solves OOB management and linecard functionality

Integrated

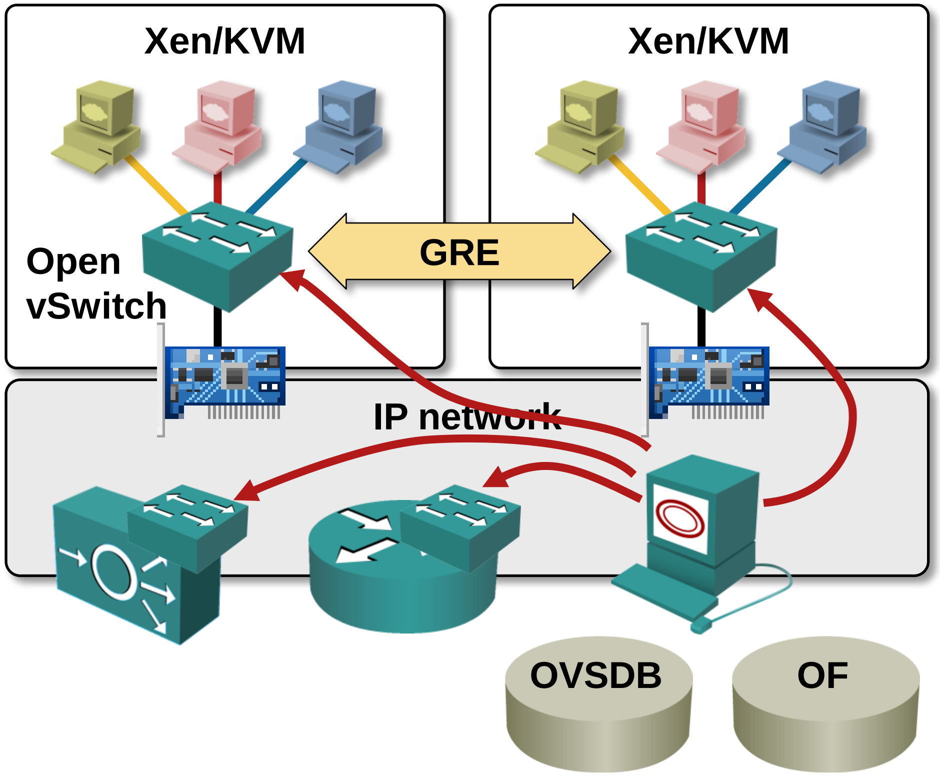
- OpenFlow classifiers/actions become part of regular packet processing
- OpenFlow provides *ephemeral state* configuration

More @ http://blog.ioshints.info/2011/11/openflow-deployment-models.html

Typical Use Case: Virtual Networking (Nicira NVP)

MAC-over-IP with control plane

- OpenFlow-capable vSwitches (OVS)
- P2P GRE tunnels provisioned with OVSDB
- MAC-to-IP mapping downloaded to OVS with OpenFlow
- Third-party physical devices with OVS



Benefits

- OpenFlow-based scalable control plane
- No interaction with transport fabric
- No IP multicast in the core

OpenFlow Is Not the Only SDN Tool

Tool/Standard	Functionality
OpenFlow (ONF)	FIB/TCAM manipulation
NETCONF (IETF)	Configuration management
OF-Config	OpenFlow switch configuration management (YANG schema)
Internet Routing System (IRS, IETF non-WG)	Routing table interaction/manipulation

Vendor APIs

- Cisco: Open Networking Environment (ONE), EEM (Tcl), Python scripting
- Juniper: Junos XML API and SLAX (human-readable XSLT)
- Arista EOS: XMPP, Linux scripting (including Python and Perl)
- Dell Force10: Open Automation Framework (Perl, Python, NetBSD shell)
- F5: iRules (Tcl-based scripts)

(Almost) Shipping OpenFlow Products

Switches – Commercial

- Brocade MLX/NetIron products
- Extreme BlackDiamond X8
- HP ProCurve
- IBM BNT G8264
- NEC ProgrammableFlow switches
- Juniper MX-Series (SDK)
- Cisco (roadmapped)
- Smaller vendors

Switches – Open Source

- Open vSwitch (Xen, KVM)
- NetFPGA reference implementation
- OpenWRT
- Mininet (emulation)

Controllers – Commercial

- Big Switch Networks (EFT?)
- NEC ProgrammableFlow Controller
- Nicira NVP

Controllers – Open Source

- NOX (C++/Python)
- Beacon (Java)
- Floodlight (Java)
- Maestro (Java)
- RouteFlow (NOX, Quagga, ...)

More @ http://www.sdncentral.com/shipping-sdn-products/
http://www.sdncentral.com/comprehensive-list-of-open-source-sdn-projects/

Conclusions

- SDN is an interesting concept
- Centralized computation and management plane makes more sense than centralized control plane
- OpenFlow is just a low-level tool
- Initial use cases: large data centers @ portals or cloud providers (cost cutting or virtualized networking)
- Still a very immature technology
- Northbound controller API is missing (but badly needed) ➔ Creating controller vendor lock-in
- Already crossed the academic ➔ commercial gap

If you want to get involved, NOW is a good time

More Information

OpenFlow standards, tools and projects

- opennetworking.org
- openflow.org
- openflowhub.org

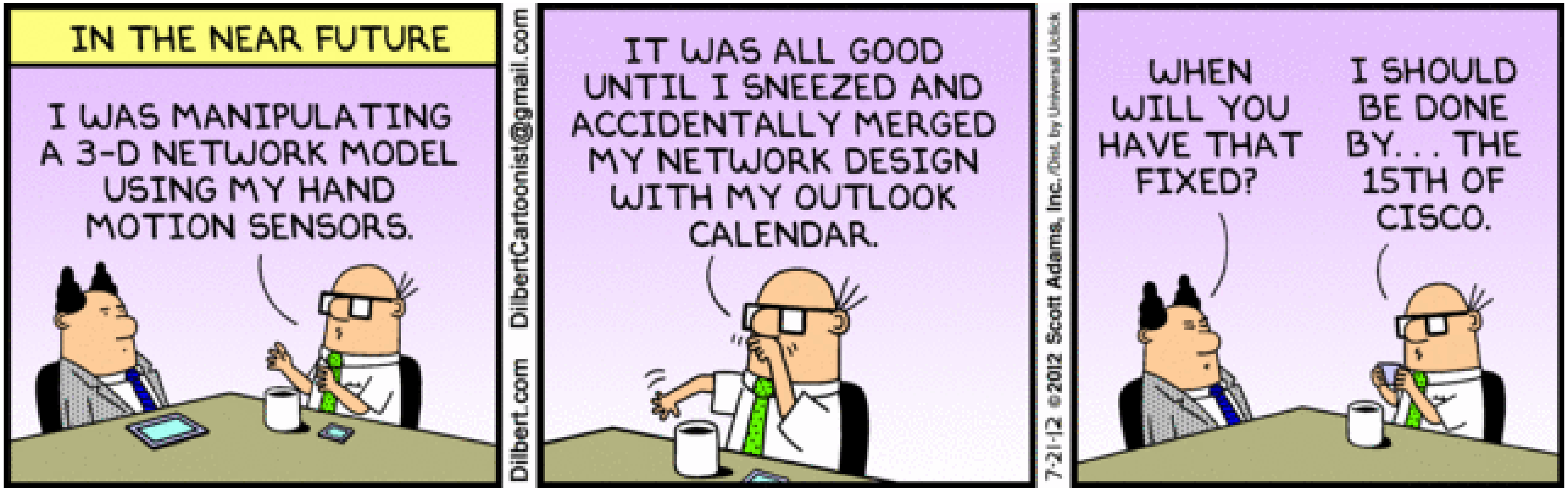
Web sites

- SDN Central (sdncentral.com)
- InCNTRE (Indiana University)

Blogs

- Networkstatic.net (Brent Salisbury, University of Kentucky)
- Networkheresy.com (Martin Casado, Nicira)
- Packet Pushers (packetpushers.net)
- Twilight in the Valley of the Nerds (Brad Casemore)
- blog.ioshints.info (yours truly)
- demo.ipSpace.net/get/OpenFlow (free OpenFlow webinar by Greg Ferro)

A Brief Look into the SDN Future


IN THE NEAR FUTURE
I WAS MANIPULATING A 3-D NETWORK MODEL USING MY HAND MOTION SENSORS.
Dilbert.com DilbertCartoonist@gmail.com
IT WAS ALL GOOD UNTIL I SNEEZED AND ACCIDENTALLY MERGED MY NETWORK DESIGN WITH MY OUTLOOK CALENDAR.
7-21-12 ©2012 Scott Adams, Inc./Dist. by Universal Uclick
WHEN WILL YOU HAVE THAT FIXED?
I SHOULD BE DONE BY... THE 15TH OF CISCO.

Source: http://dilbert.com/strips/comic/2012-07-21/

Send questions to ip@ipSpace.net or @ioshints