2015 Spring: Future Internet Class
2015.6.2

SDN for Future Internet: part 1

Jun Bi
Tsinghua University
The Driving Force for Transforming the Network Architecture

- Considering the "Genes" of the Success of the Internet

Software Defined Networking (SDN)

- SDN Has Attracted High Attention from the International Academic Society
  - What's SDN?
  - SDN Standards
  - SDN Research Overview
Why Internet is so successful?
Network Architecture is about Design

- Computer Science is quite different than other subjects in nature science
  - Refer to Prof. Lixia Zhang’s Opinion
- A design must satisfy the goals
  - To satisfy 1-2 main goals
  - Leaves other goals as Best Efforts
- Different design goals leads to different network architecture design

David Clark, Design Philosophy of the DARPA Internet Protocols, SIGCOMM1988论文

- The design goals
- How they designed the Internet to satisfy those goals
Scientific Methodology

Physicists:
1. Make a hypothesis
2. Run experiments
3. Analyze data
4. Draw conclusions
   - If data does not support the hypothesis, go back to step-1 with revised hypothesis

To get one correct conclusion the cycle often needs to be repeated $10^n$ times
Scientific Methodology

Network researchers:

1. Hypothesis = Sketch a design
   1.a Implement the design
2. Run experiments
3. Analyze data
4. Draw conclusions
   • If the results ≠ what one expected, go back to step-1 with revised design

To get a **good** design, the cycle often needs to be repeated many times
One major difference

Between physics and networking (or computing in general): the direct goals

◊ Physicists: interpret how the world works
  ◦ A newly gained understanding can then be used to build products and solve problems

◊ Network researches: design and build artifacts that solve problems/meet a need
  ◦ Scientific studies are needed to gain insights on how to make the design right (slide 22)
Another major difference

Between physics and networking:

◊ Newton’s Laws have not changed since their discovery more than 300 years ago

◊ Computing technology has seen changes of many orders of magnitude over its short 60-year history
  ◦ Technology change
    → Usages change
    → design tradeoffs change
Mandatory Goal: Connect existing networks **
- problem statement is very important

2nd Level Goal:
- Survivability **
- Support multiple types of services (TCP提供可靠服务, UDP提供不可靠的服务）*
- Must accommodate a variety of networks **
- Allow distributed management
- Be cost effective
- Allow host attachment with a low level of effort
- Allow resource accountability

Now looking back, we can see that many requirements, such as mobility, security, and scalability, were not taken into account at that time

Why you think Internet is not good enough in QoS, Security, etc. , because those goals were not the main goals)
The Internet is successful due to the OTT. The Success of OTT Depends on the *Openness* of the Internet in the **Application Layer**

**Hourglass Model (Technical Factor for the Success of the Internet)**

The transport layer provides logical communication channels between applications.
The "genes" of the Internet are openness and innovation.

What brings the success of the Internet:
The excellent network architecture allows for
the openness and innovation of network
applications.
The innovation of network applications does not require global deployment or international standards, and are independent of device vendors.

The innovation of the IP core layer, however, requires global deployment and International standards, and are dependent of device vendors.

It is easy to implement innovation and competition for network applications.

It is difficult to implement innovation and competition at the IP core layer.

There is a driving force for transforming the Internet architecture because the development of the Internet has gone beyond the requirement of its original design.

In the Internet Plus era, innovation is also required for the network core layer.
The network core layer has disadvantages in the following aspects due to its design limitations:

- **Scale and scalability**
  - Small network -> Internet of Everything
- **Security and reliability**
  - Users know each other -> Secure and trusted
- **QoS**
  - Simple services -> Differentiated services
- **PC era -> Cloud Computing, Mobile Computing era**
  - Mobility, high performance, and enhanced management, control, and customization***
- **Fast service provisioning (Apple-alike model) and new economic model for networks**
Challenges Faced Today

- Those are the new requirements. What if the requirements change again?
  - Revolution (non-IP)? Evolution (SDN)?
  - Evolution capability (smooth innovation of the network core layer)***
  - The network infrastructure should not be thrown away and started all over again.

In the era of Internet of Everything and innovation by all, higher requirements are posed on the Internet core layer. The Internet core layer must keep pace with the times, no longer standing still.
Retrospective Review and Progress of the International Research on the Future Internet Architecture

- FIND (2005-2009) -> NeTS long term FIND-like project
- NeTS Special FIA (2010-2013)
- What standards are used to verify the future Internet architecture?
  - Whether the future Internet architecture inherits the "gene" of innovation of the Internet
  - Whether the Internet is thrown away and started all over again, or whether the Internet is being inherited
Two main approaches to evolve Internet

- Approach 1: Growing a new Narrow Waist
- Approach 2: Growing an open networking architecture/ecosystem (to support multiple new narrow waists)
  - “Fat-Waist”
Approach# 1

• Growing a new narrow waist
  – Always moves the narrow waist up: three steps
    1. Design a new narrow waist to meet a requirement (could be evolutionary design like ILNP or clean-slate design like NDN)
    2. Deploy in the current Internet with “overlay network” (upper layer than today’s narrow waist layer) as a “rider” with “lower performance”
    3. If market accepts it, then every node installs it eventually— which means new “narrow waist” replaces existing narrow waist, with “native support” and “higher performance”
  – Examples: how IP replaced phone network, IPv6’ plan
  – Potentials
    • New “ICN naming layer” for content-centric requirement
FIA-NDN Project

- Where -> What
- Internetworking -> Named Data Networking (NDN)
• Alternative approach: growing an “open” networking architecture
  – Growing an ecosystem that many narrow waists can co-exist and compete (enabling diversity then species evolve)
  – IP is doing good at an ecosystem for Application and Link layers (diversity then evolve), but not open enough for the narrow waist layer (even IPv6 is hard to deploy)
  – SDN (or other names) is the exact case for approach 2
    • To support different core functions of IP layer and also for ILNP, NDN…(Goal of China’s FINE project)
    • Multi-requirements can be satisfied simultaneously
• **Personal opinions**
  • Core requirement 1: customization of network architecture or core functions
    • Varied network core technologies required by different users
      • For example, security is highly required by some, whereas QoS by others.
    • Construct an ecological environment (non-zero-sum) with support for creativity
      • The network architecture is easy to design, develop, and deploy, just like network applications.
  • Core requirement 2: smooth evolution capability
    • Capable to satisfy future requirements
  • If I was able to re-design the network architecture, I believe these two requirements must be satisfied. The core to implement such an architecture is the ability to abstract network functions.
  • It is exactly SDN that supports the most important cut-through points and benefits for the future network!
  • SDN will also usher in a new epoch when network programmers will create a history!
Software Defined Networking (SDN)

- SDN is Hot!
- What's SDN?
- Four Important Relations about SDN
- SDN Standards
- SDN Research Overview
SDN Attracted High Attention from the Industry and Academia

- \( \text{Ethane(07)} \rightarrow \text{OpenFlow(08)} \rightarrow \text{SDN(09/10)} \)

- Industrial Circle
  - In 2011, the Open Networking Foundation (ONF) was established (watershed).
  - Since 2011, the Open Networking Summit (ONS) has been held, and other similar conferences have been held in Europe and Asia.
  - IRTF established the SDNRG in Oct, 2012.
  - In 2012, China SDN Commission (ChinaSDN) was established. It has set up strategic cooperation relationship with ONF.
  - In April, 2013, Open Daylight Project (OPD) was established, and most members were from equipment vendors.

- Academic Circle
  - In 2012, SIGCOMM established the HotSDN Workshop.
  - Since 2013, SDN has been added as a topic in reputable International academic conferences.
  - In 2013, about 1/5 topics in the academic conferences of SIGCOMM were about SDN.
What HotSDN’14 is about

• SDN has quickly become pretty main-stream
  – 7/38 papers in SIGCOMM 2013
  – 9/45 papers in SIGCOMM 2014

• HotSDN serves as a forum for …
  – … early-stage work
  – … specific contributions on key technologies
  – … positions papers, “out there” ideas
  – Focused debate and discussion, involving industry and academia
disappeared

jump up

stuck

2013

Gartner 2013
Outline

Software Defined Networking (SDN)
- SDN is Hot!
- What's SDN?
- Four Important Relations about SDN
- SDN Standards
- SDN Research Overview
What Is SDN?

About SDN, Various Opinions

- The standard constitutors saw OpenFlow interfaces.
- The equipment vendors saw separation of data plane and control plane.
- Network operation and maintenance personnel saw centralized control.
- Application providers saw virtualized separation of network infrastructure and functions.
- Investors saw the change of game rules.
There are a thousand Hamlets in a thousand people’s eyes
Is it SDN?
From Divide to Tightly Coupled Feedback Loop

- **From Software Defined Networks**, O'Reilly, Aug. 2013

- **Programmable requirements**
  - Tightly coupled
    - Managed object is changed from single devices to the entire network.
    - Control speed, cycle
    - Automation
  - Bidirectional: manage->interact
    - Help developers create modern application program interfaces
    - Auto-adaptive
    - Various information presentation formats, such as Json
Life in the Fast Lane: the confluence lens

George Varghese, Microsoft Research
Confluence Definition for this talk

MAIN STREAM

IMPACTING STREAM

Inflection Point

NEW STREAM

Context Change

Transformed Ideas
Example 1: Impressionism

Realistic Painting

Photography

Impressionism

Psychology

Ideas to Canvas

Thin to thick strokes
**Example 2: Randomized Algorithms**

![Diagram showing relationships between Algorithms, Crypto, Probability, and R. Algorithms with arrows pointing to 'Always to sometimes'.]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time on 10^100 + 267</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miller-Rabin (100 trials)</td>
<td>0.3 seconds</td>
</tr>
<tr>
<td>Best Deterministic (AKS)</td>
<td>37 weeks</td>
</tr>
</tbody>
</table>
Network Verification as a confluence

Networking

Prog. Languages

Cloud services

Programs→networks

1 Solution to many, SAT to AllSAT
Line to rule coverage for testing
What Is SDN? (In My Opinion)

- There is no agreement on the definition of SDN, yet.

- In my opinion, SDN is defined in two aspects: in board sense and in narrow sense.

- In board sense, SDN means the programmability of network structure (including APIs and tools).
  - Separation or integration of data forwarding and control is not the essence of SDN. The essence is abstraction and programmability.
  - The presentation format is decided by objects, measures, and development level.
  - Some simple SDN objects do not need separation, integration, virtualization, or OpenFlow.
  - Some advanced SDN objects may need separation, integration...today, but do not need these when cutting-edge technologies are invented (for example, centralized programming point -> distributed programming).
What Is SDN? (In My Opinion)

- In narrow sense, SDN is the system of various SDN entities.
  - The "granularity and interfaces" of abstracted entities are different (for programming control).
  - Revolutionary SDN: Abstraction of new data plane. fine-granularity (such as Flow), computing, storage, etc
  - Evolutionary SDN: Open interface + existing data plane: routing table (I2RS), LabelPath (PCE), VPN (VxLAN/NVGRE/NVO3 for DCN)
  - Mixed SDN: Old + new (Software-Driven Network)

- No matter whether in broad or narrow sense, SDN is in underway. There is huge space in innovation and development.
Outline

Software Defined Networking (SDN)

- SDN is Hot!
- What's SDN?
- Four Important Relations about SDN
- SDN Standards
- SDN Research Overview
SDN and Centralized Control

- Traditional network
- Distributed management
- Difficult management
- Limited policies

- SDN network
- Centralized management
- Easy management
- Flexible, fine-granularity policies
If a Human Body is SDN: Centralized Programming Control?

- **Central nervous system (complex)**
  - Brain, spinal cord

- **Nervous system**
  - Body nervous (animal nervous) system
  - Autonomic nervous (plant nervous) system
    - Reaction, rest: heartbeat, digestion, metabolism, etc
    - An untrained person cannot control nervous system by conscious.
Do We Need Zombie or Martian?

Yesterday's network

Today's or tomorrow's network?

Future network
Centralized and distributed are not two opposite sides, but can transit to each other gradually.

The distributed control plane can carry services (plant nervous system)
- Network connectivity
- Basic functions

The centralized control plane can be used for programming control (brain and animal nervous system)
- A control logic is added to the network carrying services
- Complicated functions
- Basic functions are not affected even if the centralized control plane fails (vegetable patient)
An Example: SDN TE using the existing data plane

- Compute the traffic class routing
- Control the upgraded router
- Collect traffic traces
- Compute traffic matrices
- Collect FIB information
- Collect the topology
- Send to the controller
Shortest Path Routing

Traffic Matrix

<table>
<thead>
<tr>
<th>Source Node</th>
<th>Destination Node</th>
<th>Traffic Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>D</td>
<td>120</td>
</tr>
<tr>
<td>B</td>
<td>D</td>
<td>40</td>
</tr>
<tr>
<td>C</td>
<td>D</td>
<td>20</td>
</tr>
</tbody>
</table>
Example: calculating FIB to migrate some traffic

IP-granularity Traffic from A to D

<table>
<thead>
<tr>
<th>Destination IP Address</th>
<th>Traffic Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1.1</td>
<td>20</td>
</tr>
<tr>
<td>1.1.2.2</td>
<td>30</td>
</tr>
<tr>
<td>2.2.2.2</td>
<td>30</td>
</tr>
<tr>
<td>3.3.3.3</td>
<td>40</td>
</tr>
</tbody>
</table>

Revised FIB by STE

<table>
<thead>
<tr>
<th>Destination Prefix</th>
<th>Next Hop</th>
<th>Matching Destination IP</th>
<th>Amount of Matching Traffic</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.0.0/16</td>
<td>D</td>
<td>1.1.2.2</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>2.2.0.0/16</td>
<td>D</td>
<td>2.2.2.2</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>3.3.0.0/16</td>
<td>C</td>
<td>3.3.3.3</td>
<td>40</td>
<td>Static</td>
</tr>
<tr>
<td>1.1.1.0/24</td>
<td>B</td>
<td>1.1.1.1</td>
<td>20</td>
<td>Static</td>
</tr>
</tbody>
</table>
# Performance of Configuring Static Routes

<table>
<thead>
<tr>
<th>Number of Static Routes</th>
<th>Installing Time</th>
<th>Uninstalling Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.339</td>
<td>0.510</td>
</tr>
<tr>
<td>100</td>
<td>3.459</td>
<td>5.206</td>
</tr>
<tr>
<td>200</td>
<td>6.717</td>
<td>9.955</td>
</tr>
<tr>
<td>1000</td>
<td>30.242</td>
<td>46.508</td>
</tr>
</tbody>
</table>

**Router:** Cisco-Catalyst-3750G  
**Script:** Telnet Protocol
MLU (Maximum Link Utilization)

- **CERNET2 Data set**

- **Internet2**

- **GEANT2**
Incrementally Deplorable

CERNET2

Average MLU vs. The number of deployed nodes

Interent2

Average MLU vs. The number of deployed nodes

GEANT2

Average MLU vs. The number of deployed nodes
Relationship Between SDN and Future Network

• Due to SDN’s programmability to network architecture, SDN is a meta-architecture:
  • Metadata are data about data (who has produced them, when, what format the data are in and so on)
  • Meta-language is a sort of language for describing another language
  • SDN is a meta-architecture which is an architecture about architecture to describe or support another architecture

▪ Tsinghua University’s FINE project (Future network INnovation Environment), supported by China’s 863 high-tech program, is an example that how to describe a specific network architecture for research, design, experiment, operation, innovation, and evolution

▪ Therefore, SDN could support future network architecture
  • SDN is just a tool to describe a specific function.
Relationships between SDN and IPv6

An example I presented at IPv6 summit in April, 2013

- SDN could program a new network architecture, but in the current phase, it is mainly used to program a new IP protocol for research, experiment and operation
- Therefore, SDN supports IPv6 research and experiment
- SDN could help to accelerate IPv6’s launch
IPv6 Research and Experiments based on SDN

- Tsinghua University
  - Intra-domain IPv6 source address validation based on SDN
  - IPv6 source address validation in access network (SAVI) based on SDN
  - IPv6 transition technology based on SDN (collaborating with Huawei)
• OpenFlow is just one of the southbound interfaces of SDN. There are other programming objects/interfaces (conservative, existing, or revolutionary).

• OpenFlow-based SDN is a typical example of SDN in narrow sense (SDN system entity). For example, C language is a programming language.

• OpenFlow-based SDN is a revolutionary SDN structure. It features fine-granularity.

• OpenFlow is significant for SDN researching.
Outline

- Software Defined Networking (SDN)
  - SDN is Hot!
  - What's SDN?
  - Four Important Relations about SDN
  - SDN Standards
  - SDN Research Overview
ONF Members

- 6Wind
- A10 Networks
- Active Broadband
- ADVA Optical
- Alcatel-Lucent/Nuage
- Alibaba
- Aricent Group
- Arista
- Beijing Internet Inst.
- Big Switch Networks
- Broadcom
- Brocade
- Centec Networks
- Ceragon
- China Mobile
- Ciena
- Cisco
- Citrix
- CohesiveFT
- Colt
- Coriant
- Corsa Technology
- Cyan
- Dell/Force10
- Deutsche Telekom
- Ericsson
- ETRI
- Extreme Networks
- F5 / LineRate Systems
- Facebook
- Fiberhome Technologies
- Freescale Semi
- Fujitsu
- Gigamon
- Goldman Sachs
- Google
- Hitachi
- HP
- Huawei
- IBM
- Infinera
- Infoblox/FlowForwarding
- Intel
- Inst. for Info. Industry
- Intune Networks
- IP Infusion
- Ixia
- Juniper Networks
- KDDI
- Kemp Technologies
- Korea Telecom
- L-3 Comms-East
- Lancope
- Level3 Comms
- LSI
- Luxoft
- Marvell
- MediaTek
- Mellanox
- Metaswitch Networks
- Microsoft
- Midokura
- NCL Comms K.K.
- NEC
- Netgear
- Netronome
- Netscout Systems
- Nokia Siemens Netw.
- NoviFlow
- NTT Communications
- Optelian
- Oracle
- Orange
- Overture Networks
- Pica8
- Plexxi
- Procera Networks
- Qosmos
- Rackspace
- Radware
- Riverbed Technology
- Samsung
- SK Telecom
- Spirent
- Sunbelt AG
- Swisscom
- Tail-f Systems
- Tallac
- Tekelec
- Telecom Italia
- Telefonica
- Tellabs
- Tencent
- Texas Instruments
- Thales
- Tilera
- TorreyPoint
- Transmode
- Turk Telecom/Argela
- TW Telecom
- Vello Systems
- VeriSign
- Verizon
- Virtela
- VMware/Nicira
- Xpliant
- Yahoo!
- ZTE
ONF Members

ONF Operator/User Members

Equinix
Facebook
Goldman Sachs
Google
Level 3
Microsoft
Rackspace
TW Telecom
Verizon
Virtela
Yahoo!

Colt
Deutsche Telekom
ECI Telecom
Orange
Swisscom
Telecom Italia
Telefonica
Vodafone

Alibaba
Baidu
China Mobile
China Telecom
ETRI
Korea Telecom
KDDI
NTT Communications
PCCW Global
SK Telecom
Tata Communications
Telecom Malaysia
Tencent
ONF Organization

- Board
- TAG
- Chip Makers Advisory Board
- WG/DG
- Research Associate
ONF WG

- Extensibility WG
  - HP/VMare
- Config & Management WG
  - Microsoft/NTT
- Testing & Interop WG
  - Ixia/IU
- Architecture & Framework WG
  - NEC
- Forwarding Abstractions WG
  - Brocade
- Migration WG (2013 Apr)
  - Huawei
- NBI WG (2013 Oct)
  - HP/Huawei
- Wireless & Mobility WG (2013 Oct)
  - Huawei
- Optical Transport WG
  - Ciena
- Market Education Committee - Ciena/Netronome
- Two Member’s Work Days
ONF DG

- Security DG
- Goldman Sachs
- Carrier Grade SDN DG
- Layer 4-7 DG (ETSI NFV)
- East-west Interface
Mature Industrial Standard OpenFlow

(1) The control plane and data plane are separated.

(2) Protocols run on an independent controller.

(3) A switch is abstracted into the processing procedures of multiple flow tables (matching + action).

(4) External protocols interact with the device through the OpenFlow protocol running on the controller.
OpenFlow 1.0 Flow Table

**Match Fields**

- Ingress Port
- Ethernet: SA, DA, Type, ID, Priority
- VLAN: SA, DA, Protocol, TOS
- IP: SA, DA, Protocol
- TCP/UDP: Src, Dst

**Flow Table OF1.0 style**

- Classifier, Action, Statistics
- Classifier, Action, Statistics
- Classifier, Action, Statistics
- Classifier, Action, Statistics

**Actions**

- Forward
  - Virtual Port
    - ALL
    - CONTROLLER
    - LOCAL
    - TABLE
    - IN_PORT
- Drop
  - NORMAL
  - FLOOD
- Enqueue
- Modify Field

**Mandatory Action**

**Optional Action**
OpenFlow1.3 Table

Packets are matched against multiple tables.

-combinations complexity $O(n! \times a(2^l))$ paths where $n =$ number of tables, $a =$ number of actions and $l =$ width of match fields

---

**Match Fields** | **Priority** | **Counters** | **Instructions** | **Timeouts** | **Cookie**
POF proposed by Huawei

- Protocol Specific Application
- Protocol Agnostic Tables/Instructions

Controller

OpenFlow+

- Flow Tables
- POF Instructions

POF Data Path

- Novel Applications & Services
- Programming Languages
- Application API
- Compiler
- Flow Instruction Set

Forwarding Elements

- NPU
- Hardware Abstraction Layer
- Flex Flow Processor
- Driver
- CPU
- ASIC

- Programmable
- Network optimized
- Runtime & Remote reprogrammable
- Table driven & protocol blind
- Flow instruction set
- Flexible
- Generic
- Standard
- Low level instruction set

- High performance
OpenFlow Standards - EXT WG

- **OpenFlow 1.0.0**  2009.12.31
  - OpenFlow 1.0.1  2012.6.7  OpenFlow 1.0.2  2013.10.04

- **OpenFlow 1.1.0**  2011.02.28
  - Pipeline  （Not compatible with OF1.0）

- **OpenFlow 1.2**  2011.12
  - OXM （Based on OF1.1）

- **OpenFlow 1.3.0**  2012.06.25 （Based on OF1.2 and 1.1）
  - Stable target for hardware vendors /long term support ONF
  - Extensions for 1.3.X : new features
    - OpenFlow 1.3.1  2012.09.06  OpenFlow 1.3.2  2013.04.25
    - OpenFlow 1.3.3  2013.10.18  OpenFlow 1.3.4  2014.03.27
    - OF 1.3.5  2015.4

- **OpenFlow 1.4.0**  2013.10.15  OF 1.4.1  2015.4

- **OpenFlow 1.5.0**  2014.12.19  OF 1.5.1  2015.4
Config WG

- OF-Config 1.0  2009.12.23
- OF-Config 1.1  2012.01.25
- OF-Config 1.1.1  2013.03.23
- OpenFlow Notifications Framework 1.0  2013.10.15
- OF-Config 1.2  2014.1.9
  - Logical switch NDM (Negotiable Datapath Models)
Test WG

- Conformance Test Specification for OpenFlow Switch Specification 1.0.1  2013.08.13
  - 2013.10.17, the 1\textsuperscript{st} certified: NEC ProgrammableFlow® Switches PF5248

- OpenFlow v1.3 Switch Certification

- Plugfest
  - In USA and China

- Certification Center
  - IU InCNTRE (USA)
  - BII (China)
  - UNH-IOL (USA)
Outline

Software Defined Networking (SDN)
- SDN is Hot!
- What's SDN?
- Four Important Relations about SDN
- SDN Standards
- SDN Research Overview
• **Areas of Interest**
  • Classification of SDN models, including
    • Definitions
    • Taxonomies
    • Relationship to work ongoing in the [IETF](https://www.ietf.org) and other SDOs
  • SDN model scalability and applicability
  • Multi-layer programmability and feedback control systems
  • System Complexity
  • Network description languages, abstractions, interfaces and compilers
    • Including methods and mechanisms for (on-line) verification of correct operation of network/node function.
• Security
• **Potential Work Items**
  • Survey of SDN approaches and Taxonomies
  • Open Issues in Software-Defined Networking Research
SDN Research

**Architecture**
- Ethane: Taking Control of the Enterprise, SIGCOMM07
- OpenFlow: Enabling Innovation in Campus Networks, CCR08
- The Future of Networking, and the Past of Protocols, Scott Shenker’s PPT at ONS11
- Software-Defined Internet Architecture: Decoupling Architecture from Infrastructure, HotNet12
- SDN 2.0

**Theory**
- Using Routers to Build Logic Circuits: How Powerful is BGP?, ICNP13 (BGP has the same computing power as a Turing Machine)
**SDN Correctness: system platform based**

- **Data plane**
  
  - Header Space Analysis: Static Checking For Networks, NSDI12 (Header Space Bits, not real-time)
  
  - Real Time Network Policy Checking using Header Space Analysis (NetPlumber, based on graph, real-time), NSDI13
  
  - Veriflow: Verifying Network-Wide Invariants in Real Time, NSDI13 (Veriflow, trie tree based)

- **Control plane based: verifying APPs**
  
  - A NICE Way to Test OpenFlow Applications, NSDI12 (Nice, model checker based)
SDN Correctness: Programming based

- Programming language based
  - Frenetic: A Network Programming Language, ICFP11 (Frenetic)
  - Composing Software-Defined Networks, NSDI13 (Pyretic)
  - A Compiler and Run-time System for Network Programming Languages, POPL13 (NetCore, declarative language)
  - Machine-Verified Network Controllers, PLDI13 (over NetCore)
  - Maple: Simplifying SDN Programming Using Algorithmic Policies, SIGCOMM13 (Maple, new algorithmic language)
  - Participatory Networking: An API for Application Control of SDNs, SIGCOMM13 (PANE, tree based)

- Update mechanism based
  - Abstractions for Network Update, SIGCOMM12
Freneticire

Applications

Frenetic API

Runtime System
(Implemented with NetCore)

Python (Pyretic)

OCaml

OpenFlow

[Diagram showing the integration of Frenetic API with the runtime system using Python and OCaml, and the connection to OpenFlow.]
Controller Design

- Onix: A Distributed Control Platform for Large-scale Production Networks, OSDI10 (distributed database)
- Scalable Flow-based Networking with DIFANE, SIGCOMM10 (special switch)
- DevoFlow: Scaling Flow Management for High-performance Networks, SIGCOMM11 (flow aggregation)

Data Plane Design

- Forwarding Metamorphosis: Fast Programmable Match-Action Processing in Hardware for SDN, SIGCOMM13
Testing, Debugging, and Management

- A SOFT Way for OpenFlow Interoperability Testing, CoNEXT12
- Software Defined Traffic Measurement with OpenSketch, NSDI 13
- StEERING: A Software-Defined Networking for Inline Service Chaining, ICNP2013
SDN Research

- **Application in Campus**
  - Can the Production Network Be the Testbed?, OSDI10

- **Application in Network Security**
  - FRESCO: Modular Compostable Security Services for Software-Defined Networks, NDSS13

- **Application in MiddleBox**
  - SIMPLE-fying MiddleBox Policy Enforcement Using SDN, SIGCOMM13

- **Application in Private WAN**
  - B4: Experience with a Globally-Deployed Software Defined WAN, SIGCOMM13
  - Achieving High Utilization with Software-Driven WAN, SIGCOMM13
SDN Research

Application in Data Centers
- Hedera: Dynamic Flow Scheduling for Data Center Networks, NSDI10
- ElasticTree: Saving Energy in Data Center Networks, NSDI10
- ServerSwitch: A Programmable and High Performance Platform for Data Center Networks, NSDI11
- zUpdate: Updating Data Center Networks with Zero Los, SIGCOMM13

Application in Video Streaming
SDN deployment - Google Private WAN

G-Scale WAN Usage

- Exit testing "opt in" network
- SDN rollout
- SDN fully Deployed
- Central TE Deployed

Traffic

9. CONCLUSIONS

This paper presents the motivation, design, and evaluation of B4, a Software Defined WAN for our data center to data center connectivity. We present our approach to separating the network plane from the data plane to enable rapid deployment of user control services. Our first such service, centralized engineering allocates bandwidth among competing services application priority, dynamically shifting communication and prevailing failure conditions.

Our Software Defined WAN has been in production for three years, now serves more traffic than our public facing WAN, and has a higher growth rate. B4 has enabled us to deploy substantial cost-effective WAN bandwidth, running many links at near 100% utilization for extended periods. At the same time, SDN is not a cure-all. Based on our experience, bottlenecks in bridging protocol packets from the control plane to the data plane and overheads in hardware programming are important areas for future work.

Figure 1: B4 worldwide deployment (2011).

Figure 2: B4 architecture overview.
SDN deployment - Microsoft Private WAN

Figure 8: Our testbed. (a) Partial view of the equipment. (b) Emulated DC-level topology. (c) Closer look at physical connectivity for a pair of DC.

Figure 10: SWAN achieves near-optimal throughput.

Figure 13: SWAN is fairer than MPLS TE.

Figure 17: Time for network update.

Figure 18: (a) SWAN carries close to optimal traffic even during updates. (b) Frequent updates lead to higher throughput.

Figure 12: SWAN carries more traffic than MPLS TE.
Future SDN Research Direction
(SW or HW? Core or Edge?)

HW vs SW
(Shao-lin) (Wu-dang)

P4 -> PIF (Barefoot)

Figure 1: The source host sends a packet to an edge switch, which after providing network services, sends it across the fabric for the egress switch to deliver it to the destination host. Neither host sees any internals of the fabric. The control planes of the edge and fabric are similarly decoupled.
The "genes" of the Internet are openness and innovation.

The Internet has become a success because the excellent network architecture supports the openness and innovation of network applications.

The driving force for transforming the network architecture is the ever-changing requirement. The Internet Plus era has put forward new requirements for the network core layer.

SDN is playing an important role in the new round network architecture transformation, and inherits Internet's openness and innovation at the network core layer.

The success of Internet applications is created by PC programmers.

The success of the network core layer in the Internet Plus era will be created by network programmers!