

Software Defined Networks

SIGCOMM'15 Topic Preview



Laurent Vanbever

ETH Zürich

SIGCOMM'15

August, 17 2015

3 213

3 213

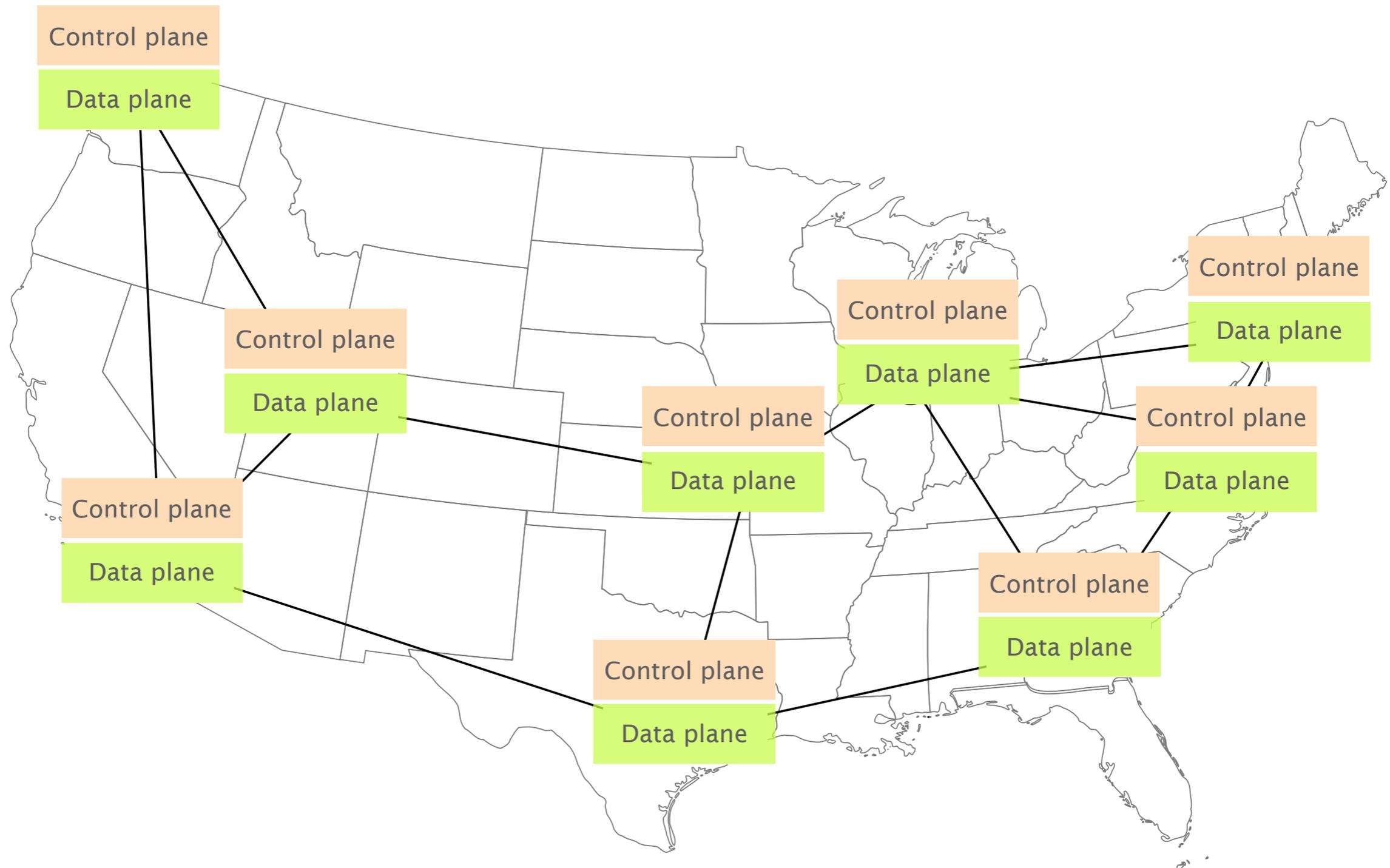
of citations of the original
OpenFlow paper in ~6 years

Software Defined Networks

Software Defined Networks

What is this thing?

A network is a distributed system whose behavior depends on each element configuration

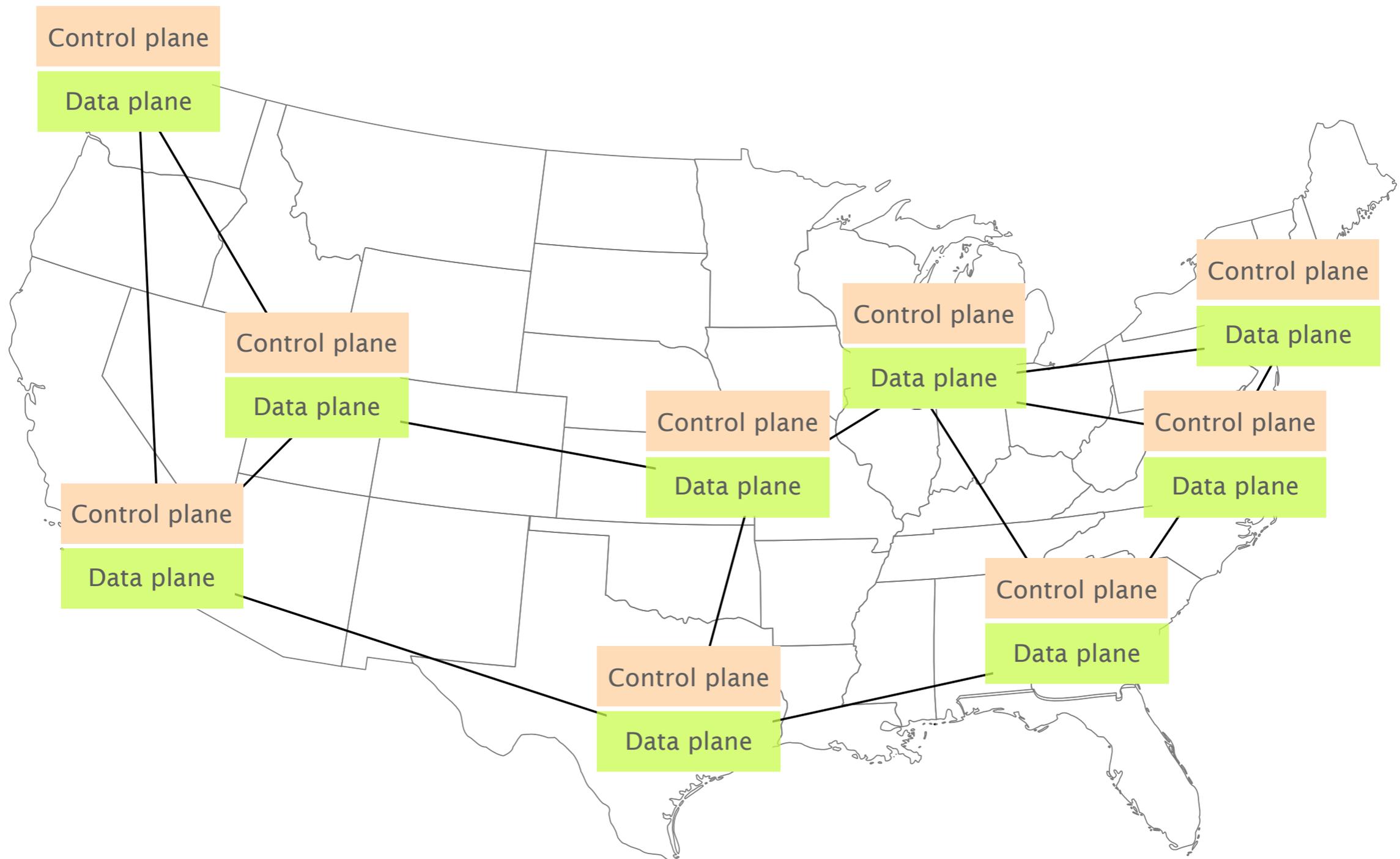


Configuring each element is often done manually,
using arcane low-level, vendor-specific “languages”

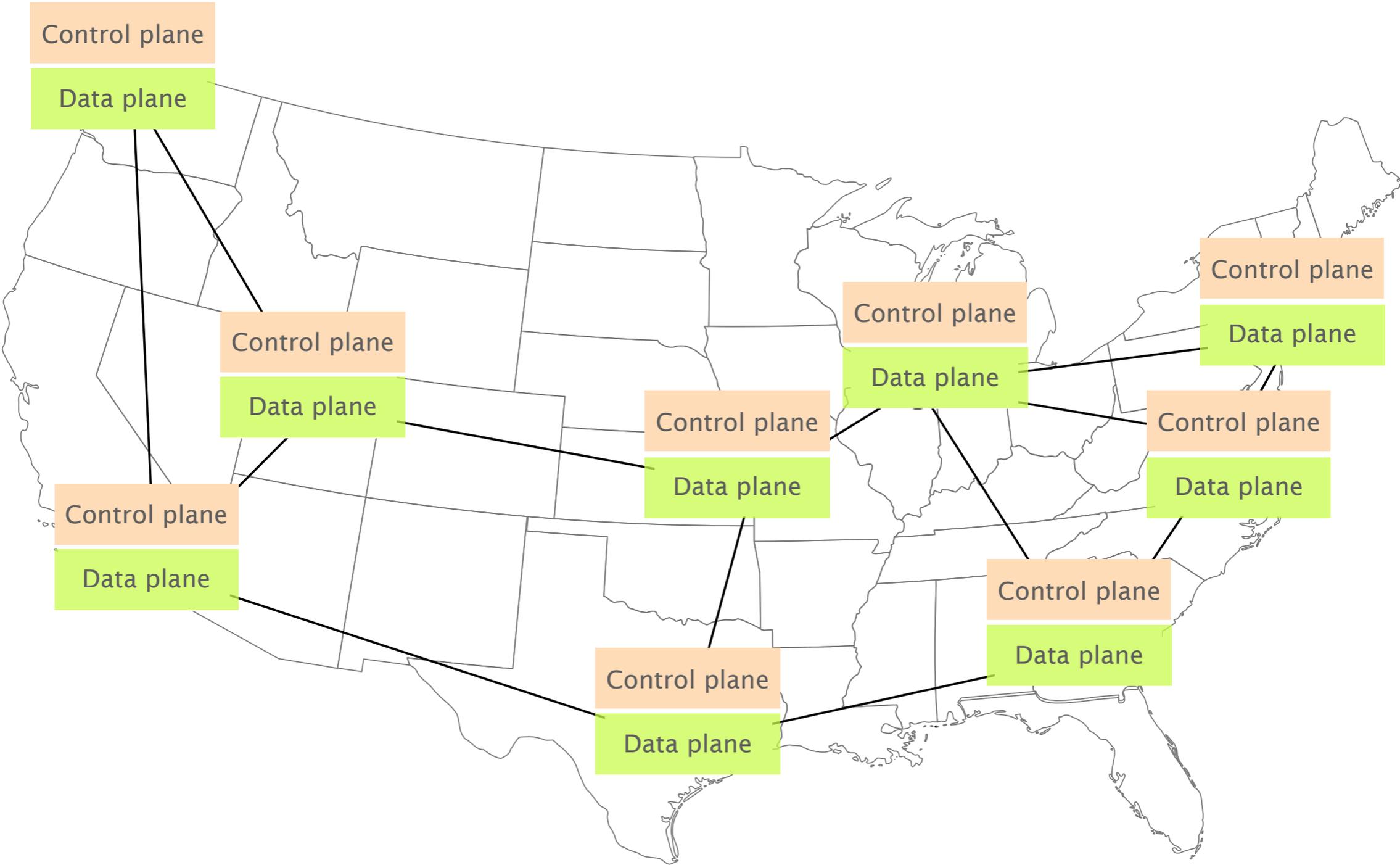
“Human factors are responsible
for 50% to 80% of network outages”

Juniper Networks, *What's Behind Network Downtime?*, 2008

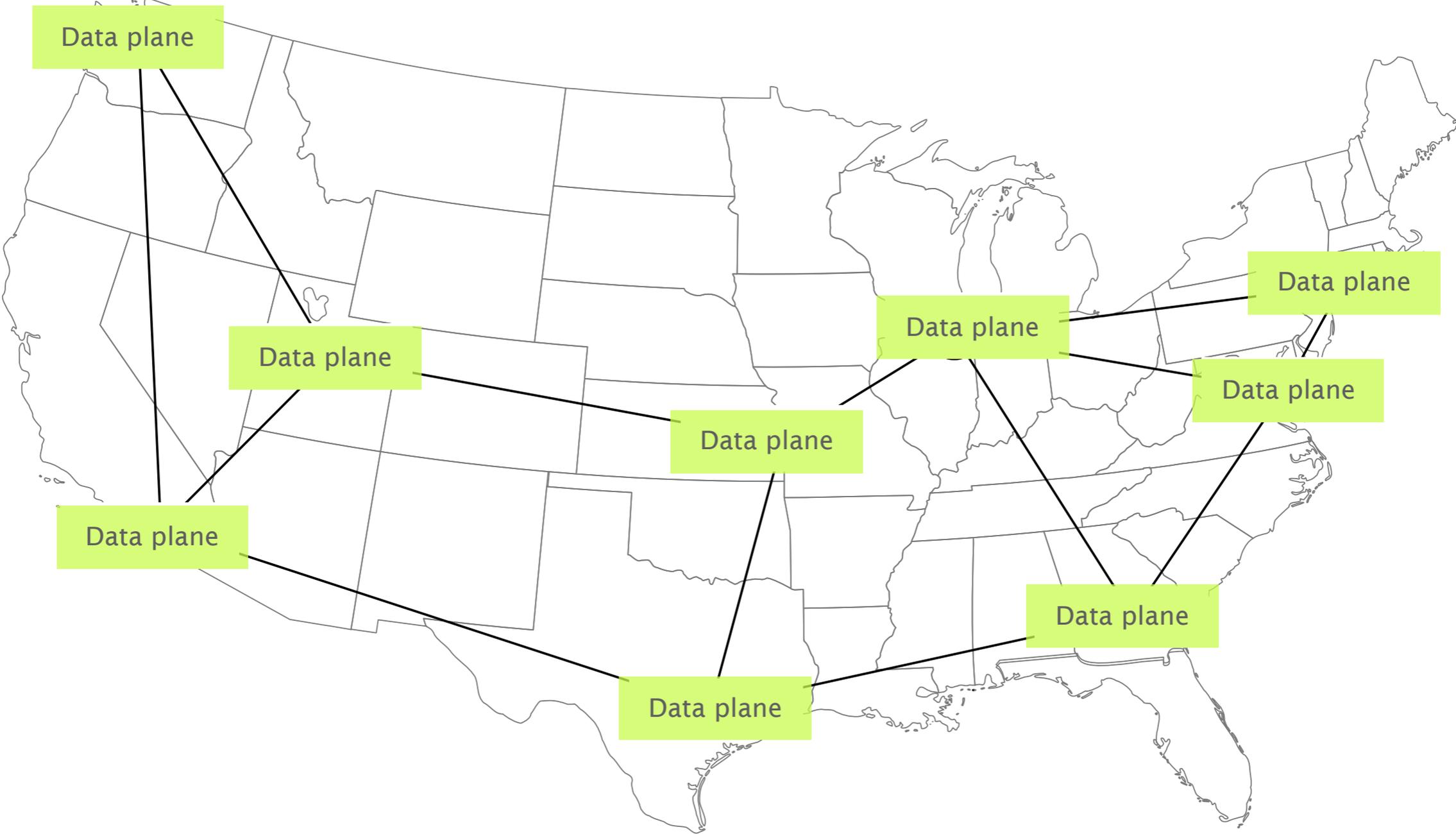
In contrast, SDN simplifies networks management...



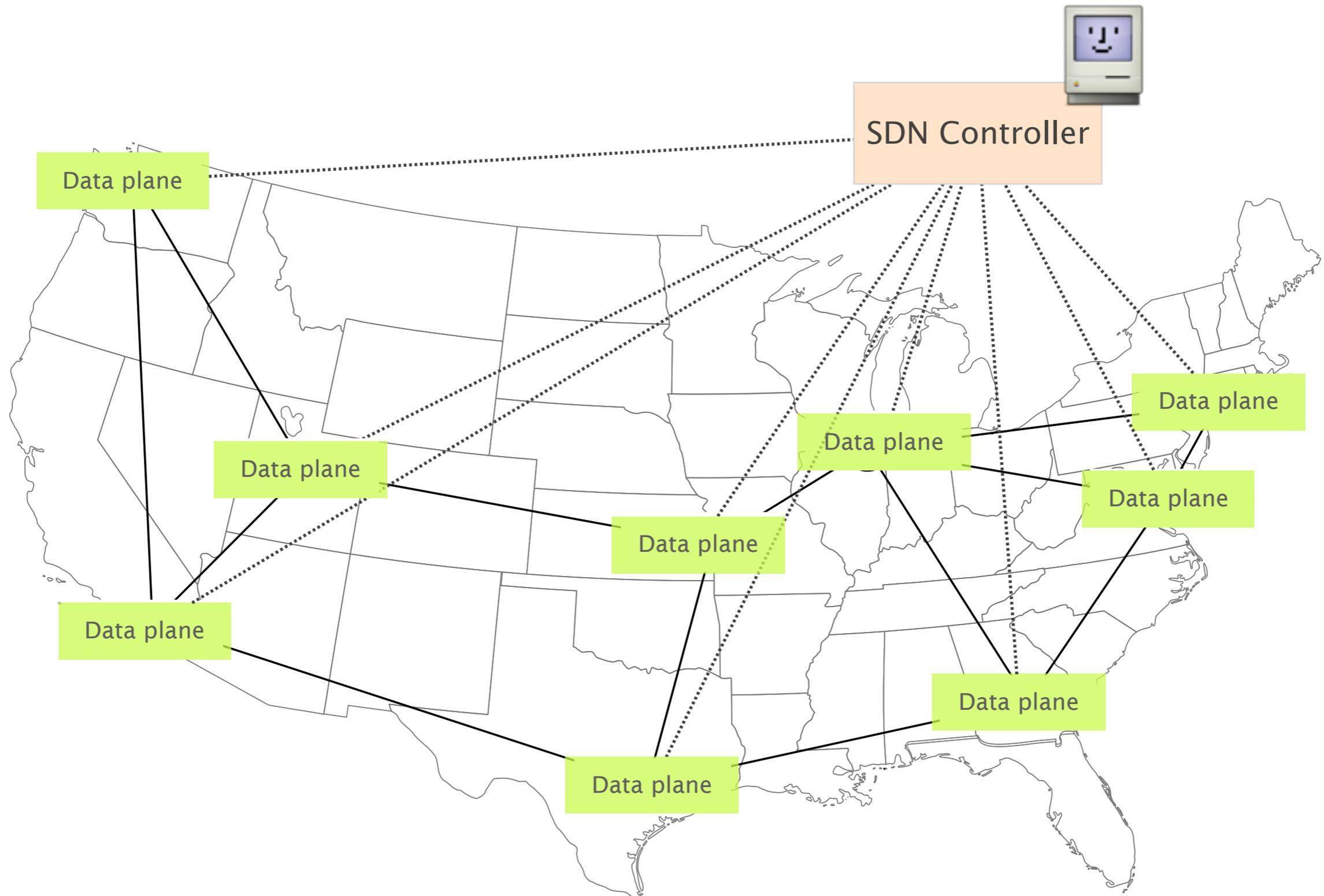
...by removing the intelligence from the equipments



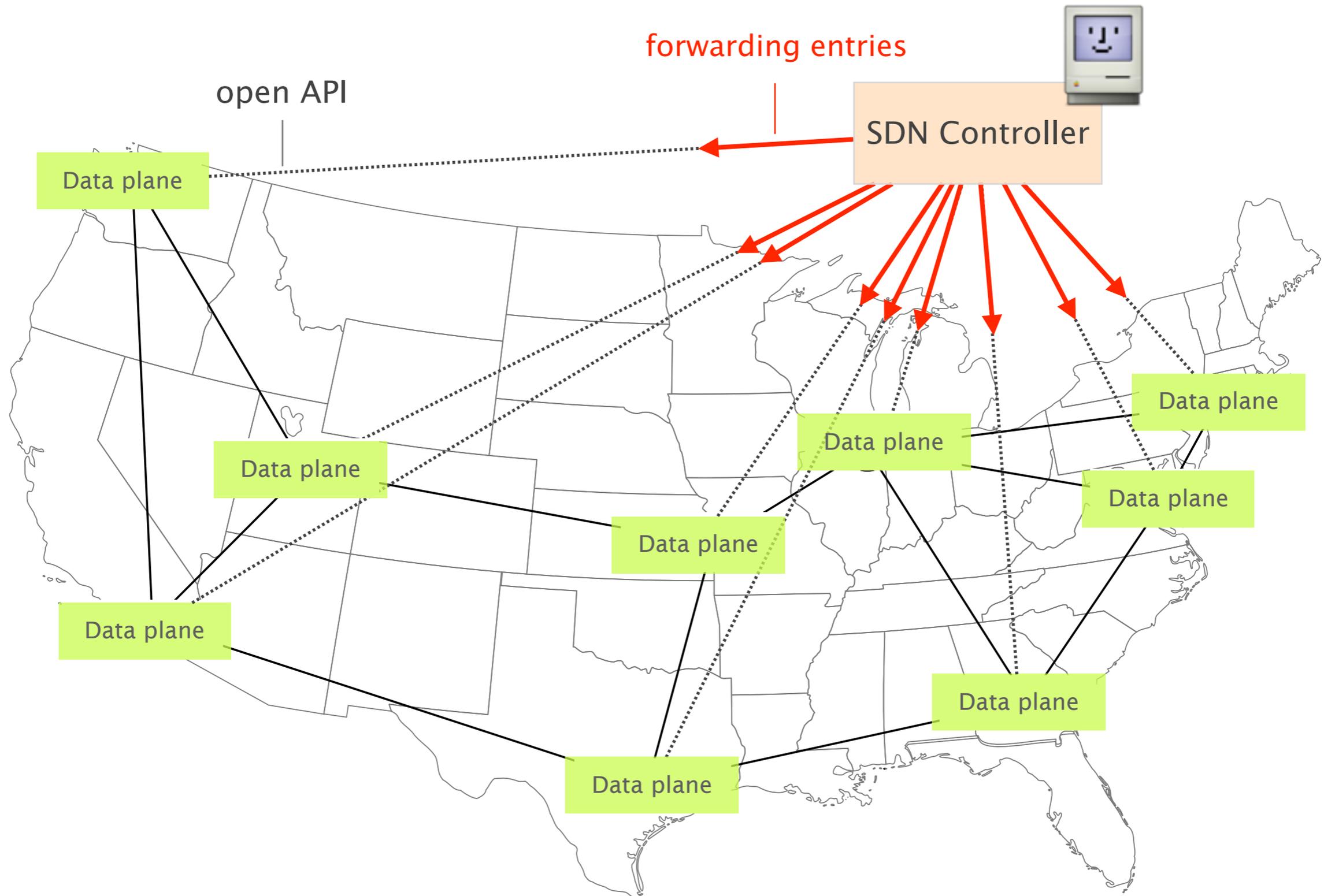
...by removing the intelligence from the equipments



... and centralizing it in a SDN controller that can run arbitrary programs



The SDN controller **programs** forwarding state in the devices using an open API (e.g., OpenFlow)



Software Defined Networks

Why should you care?!

SDN enables us, researchers,
to innovate, at a much faster pace

Before SDN

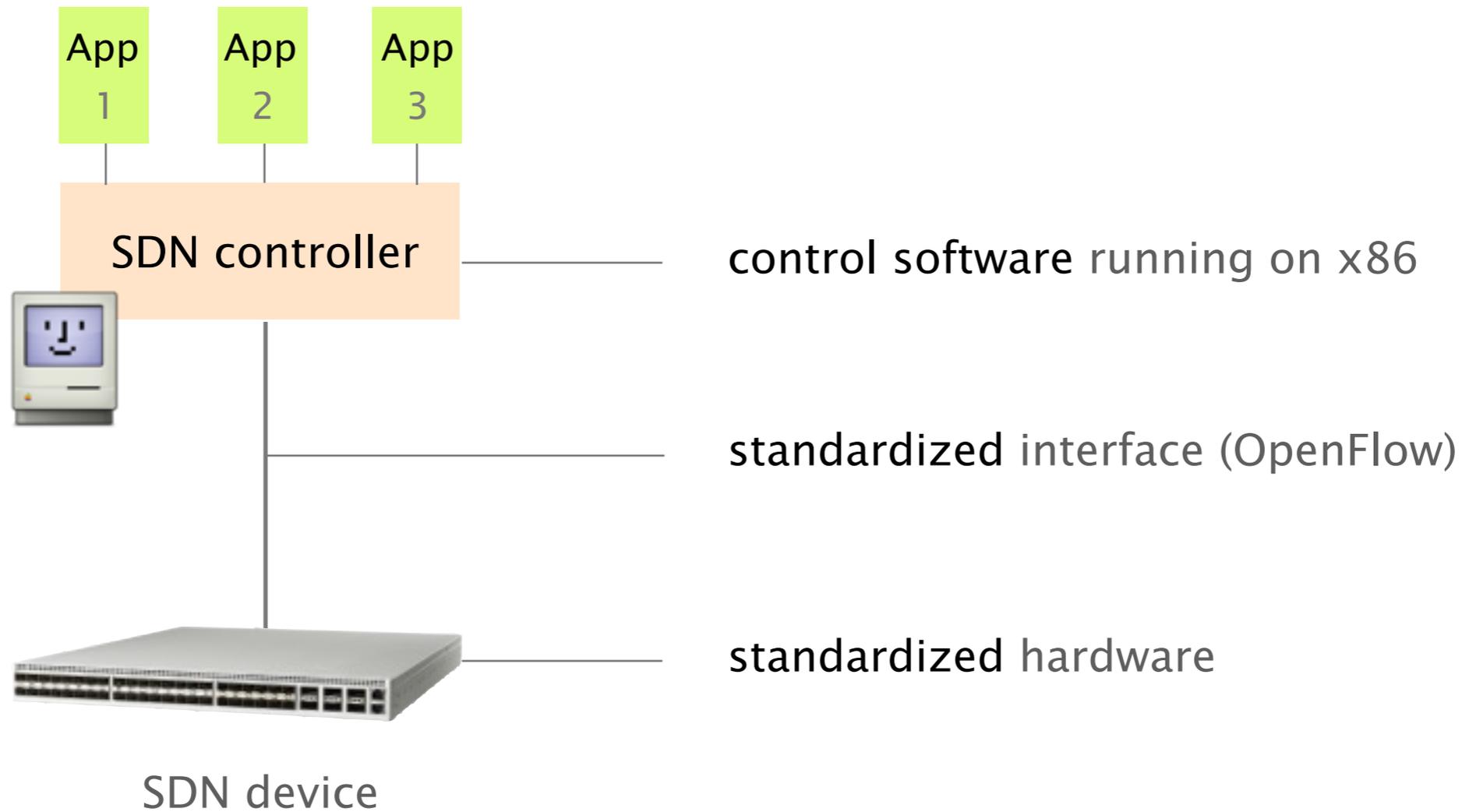


closed software

closed hardware

Cisco™ device

After SDN



SDN track @SIGCOMM'15

BwE: Flexible, Hierarchical Bandwidth Allocation for WAN Distributed Computing

Alok Kur
Nikhil Kasin
Björn Ca

A Declarative and Expressive Approach to Control Forwarding Paths in Carrier-Grade Networks

Renaud Hartert^{*}, Stefano Vissicchio^{*}, Pierre Schaus^{*}, Olivier Bonaventure^{*},
Clarence Filisfilis[†], Thomas Telkamp[†], Pierre Francois[‡]

^{*} Université catholique de Louvain [†] Cisco Systems, Inc. [‡] IMDEA Networks Institute
^{*} firstname.lastname@uclouvain.be [†] {cifsfil,thtelkam}@cisco.com [‡] pierre.francois@irdea.org

ABSTRACT
WAN bandwidth requirements are economically infeasible. It is important to allocate bandwidth efficiently and based on traffic. For example, service to receive it such an allocation, favoring allocation reservation that is the ideal basis for design and implementation. BwE supports (i) setting prioritized bandwidth reservation an arbitrary circuit and delegation archy, all accounting are conditions, (ii) an engineered network to override (perhaps) conditions. BwE has del utilization and simple users.

CCS Concepts
•Networks → Network architectures; Traffic engineering algorithms; Network management; Routing protocols; •Theory of computation → Constraint and logic programming;

Keywords
SDN; traffic engineering; service chaining; segment routing; MPLS; ISP; optimization
^{*}R. Hartert is a research fellow of F.R.S.-FNRS, and S. Vissicchio is a postdoctoral researcher of F.R.S.-FNRS.

CCS Concepts
•Networks → Network architectures; Traffic engineering algorithms; Network management; Routing protocols; •Theory of computation → Constraint and logic programming;

Keywords
SDN; traffic engineering; service chaining; segment routing; MPLS; ISP; optimization
^{*}R. Hartert is a research fellow of F.R.S.-FNRS, and S. Vissicchio is a postdoctoral researcher of F.R.S.-FNRS.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom
© 2015 ACM. ISBN 978-1-4503-3513-0. \$15.00
DOI: http://dx.doi.org/10.1145/2785956.2787495

PGA: Using Graphs to Express and Automatically Reconcile Network Policies

Chaitan Prakash^{1,2}, Jeongkeun Lee¹, Yoshio Turner^{2,3}, Joon-Myung Kang¹, Aditya Akella⁴,
Sujata Banerjee¹, Charles Clark¹, Yadi Ma¹, Puneet Sharma¹, Ying Zhang¹
¹University of Wisconsin-Madison, ²HP Labs, ³Banyan, ⁴HP Networking

ABSTRACT

Software Defined Networking (SDN) and cloud automation enable a large number of diverse parties (network operators, application admins, tenants/end-users) and control programs (SDN Apps, network services) to generate network policies independently and dynamically. Yet existing policy abstractions and frameworks do not support natural expression and automatic composition of high-level policies from diverse sources. We tackle the open problem of automatic, correct and fast composition of multiple independently specified network policies. We first develop a high-level Policy Graph Abstraction (PGA) that allows network policies to be expressed simply and independently, and leverage the graph structure to detect and resolve policy conflicts efficiently. Besides supporting ACL policies, PGA also models and composes service chaining policies, i.e., the sequence of middleboxes to be traversed, by merging multiple service chain requirements into conflict-free composed chains. Our system validation using a large enterprise network policy dataset demonstrates practical composition times even for very large inputs, with only sub-millisecond runtime latencies.

CCS Concepts

•Networks → Programming interfaces; Network management; Middle boxes / network appliances; Network domains; Network manageability; Programmable networks; Data center networks;

Keywords

Policy graphs; Software-Defined Networks
^{*}This work was performed while at HP Labs.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom
© 2015 ACM. ISBN 978-1-4503-3513-0. \$15.00
DOI: http://dx.doi.org/10.1145/2785956.2787506

1. INTRODUCTION

Computer networks, be they ISPs, enterprise, datacenter, campus or home networks, are governed by high-level policies derived from network-wide requirements. These network policies primarily relate to connectivity, security and performance, and dictate who can have access to what network resources. Further, policies can be static or dynamic (e.g., triggered). Traditionally, network admins translate high level network policies into low level network configuration commands and implement them on network devices, such as switches, routers and specialized network middleboxes (e.g., firewalls, proxies, etc.). The process is largely manual, often internalized by experienced network admins over time. In large organizations, multiple policy sub-domains exist (e.g., server admins, network engineers, DNS admins, different departments) that set their own policies to be applied to the network components they own or manage. Admins and users who share a network have to manually coordinate with each other and check that the growing set of policies do not conflict and match their individually planned high level policies when deployed together.

Given this current status of distributed network policy management, policy changes take a long time to plan and implement (often days to weeks) as careful semi-manual checking with all the relevant policy sub-domains is essential to maintain correctness and consistency. Even so, problems are typically detected only at runtime when users unexpectedly lose connectivity, security holes are exploited, or applications experience performance degradation.

And the situation can get worse as we progress towards more automated network infrastructures, where the number of entities that generate policies independently and dynamically will increase manifold. Examples include SDN applications in enterprise networks, tenants/users of virtualized cloud infrastructures, and Network Functions Virtualization (NFV) environments, details in §2.1.

In all of these settings, it would be ideal to eagerly and automatically detect and resolve conflicts between individual policies, and compose them into a coherent conflict-free policy set, well before the policies are deployed on the physical infrastructure. Further, having a high level policy abstraction and decoupling the policy specification from the underlying physical infrastructure would significantly reduce the burden

Central Control Over Distributed Routing

http://fibbing.net

Stefano Vissicchio^{*}, Olivier Tilmans^{*}, Laurent Vanbever[†], Jennifer Rexford[‡]
^{*} Université catholique de Louvain, [†] ETH Zurich, [‡] Princeton University
^{*} name.surname@uclouvain.be, [†]lvvanbever@ethz.ch, [‡]jrex@cs.princeton.edu

ABSTRACT

Centralizing routing decisions offers tremendous flexibility, but sacrifices the robustness of distributed protocols. In this paper, we present *Fibbing*, an architecture that achieves both flexibility and robustness through central control over distributed routing. *Fibbing* introduces fake nodes and links into an underlying link-state routing protocol, so that routers compute their own forwarding tables based on the augmented topology. *Fibbing* is expressive, and readily supports flexible load balancing, traffic engineering, and backup routes. Based on high-level forwarding requirements, the *Fibbing* controller computes a compact augmented topology and injects the fake components through standard routing-protocol messages. *Fibbing* works with any unmodified routers speaking OSPF. Our experiments also show that it can scale to large networks with many forwarding requirements, introduces minimal overhead, and quickly reacts to network and controller failures.

CCS Concepts

•Networks → Routing protocols; Network architectures; Programmable networks; Network management;

Keywords

Fibbing; SDN; link-state routing

1. INTRODUCTION

Consider a large IP network with hundreds of devices, including the components shown in Fig. 1a. A set of IP addresses (D_1) see a sudden surge of traffic, from multiple entry points (A , D , and E), that congests a

^{*}S. Vissicchio is a postdoctoral researcher of F.R.S.-FNRS.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom
© 2015 ACM. ISBN 978-1-4503-3513-0. \$15.00
DOI: http://dx.doi.org/10.1145/2785956.2787497

part of the network. As a network operator, you suspect a denial-of-service attack (DoS), but cannot know for sure without inspecting the traffic as it could also be a flash crowd. Your goal is therefore to: (i) isolate the flows destined to these IP addresses, (ii) direct them to a scrubber connected between B and C , in order to “clean” them if needed, and (iii) reduce congestion by load-balancing the traffic on unused links, like (B , E).

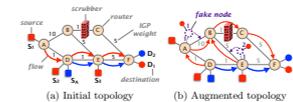


Figure 1: Fibbing can steer the initial forwarding paths (see (a)) for D_1 through a scrubber by adding fake nodes and links (see (b)).

Performing this routine task is very difficult in traditional networks. First, since the middlebox and the destinations are not adjacent to each other, the configuration of multiple devices needs to change. Also, since intra-domain routing is typically based on shortest path algorithms, modifying the routing configuration is likely to impact many other flows not involved in the attack. In Fig. 1a, any attempt to reroute flows to D_1 would also reroute flows to D_2 since they home to the same router. Advertising D_1 from the middlebox would attract the right traffic, but would not necessarily alleviate the congestion, because all D_1 traffic would traverse (and congest) path (A, D, E, B), leaving (A, B) unused. Well-known Traffic-Engineering (TE) protocols (e.g., MPLS RSVP-TE [1]) could help. Unfortunately, since D_1 traffic enters the network from multiple points, many tunnels (three, on A , D , and E , in our tiny example) would need to be configured and signaled. This increases both control-plane and data-plane overhead. Software Defined Networking (SDN) could easily solve the problem as it enables centralized and direct control of the forwarding behavior. However, moving away from distributed routing protocols comes at a cost. In-

bandwidth management

network policies

programmability

SDN track @SIGCOMM'15

BwE: Flexible, Hierarchical Bandwidth Allocation for WAN Distributed Computing

Alok Kur
Nikhil Kasin
Björn Ca

A Declarative and Expressive Approach to Control Forwarding Paths in Carrier-Grade Networks

Renaud Hartert*, Stefano Vissicchio*, Pierre Schaus*, Olivier Bonaventure*,
Clarence Filisfilis†, Thomas Telkamp†, Pierre Francois‡

* Université catholique de Louvain † Cisco Systems, Inc. ‡ IMDEA Networks Institute
* firstname.lastname@uclouvain.be † {cfilifil,htelkam}@cisco.com ‡ pierre.francois@imdea.org

ABSTRACT

WAN bandwidth requirements are often infeasible. It is important to allocate bandwidth efficiently and based on traffic conditions. For example, service to receive it such an allocation, favoring allocation reservation that is the ideal basis for design and implementation. BwE supports (i) setting prioritized bandwidth reservation that is an arbitrary criterion and delegation of bandwidth to applications, (ii) accounting for network conditions, (iii) engineering network to override (perhaps) bandwidth allocations. BwE has del utilization and simple years.

CCS Concepts

•Networks → Network architectures; Network management; Routing protocols; •Theory of computation → Constraint and logic programming;

Keywords

SDN; traffic engineering; service chaining; segment routing; MPLS; ISP; optimization

*R. Hartert is a research fellow of F.R.S.-FNRS, and S. Vissicchio is a postdoctoral researcher of F.R.S.-FNRS.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom
© 2015 ACM. ISBN 978-1-4503-2159-8... \$15.00
DOI: http://dx.doi.org/10.1145/2785956.2787495

ABSTRACT

SDN simplifies network management by relying on declarativity (high-level interface) and expressiveness (network flexibility). We propose a solution to support those features while preserving high robustness and scalability as needed in carrier-grade networks. Our solution is based on (i) a two-layer architecture separating connectivity and optimization tasks; and (ii) a centralized optimizer called DEFO, which translates high-level goals expressed almost in natural language into compliant network configurations. Our evaluation on real and synthetic topologies shows that DEFO improves the state of the art by (i) achieving better trade-offs for classic goals covered by previous works, (ii) supporting a larger set of goals (refined traffic engineering and service chaining), and (iii) optimizing large ISP networks in few seconds. We also quantify the gains of our implementation, running Segment Routing on top of IS-IS, over possible alternatives (RSVP-TE and OpenFlow).

CCS Concepts

•Networks → Network architectures; Traffic engineering algorithms; Network management; Routing protocols; •Theory of computation → Constraint and logic programming;

Keywords

SDN; traffic engineering; service chaining; segment routing; MPLS; ISP; optimization

*R. Hartert is a research fellow of F.R.S.-FNRS, and S. Vissicchio is a postdoctoral researcher of F.R.S.-FNRS.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom
© 2015 ACM. ISBN 978-1-4503-2159-8... \$15.00
DOI: http://dx.doi.org/10.1145/2785956.2787495

1. INTRODUCTION

By promising to overcome major problems of traditional per-device network management (e.g., see [1]), centralized architectures enabled by protocols like OpenFlow [2] and segment routing [3] are attracting huge interest from both researchers and operators. Two features are key to this success: declarativity and expressiveness. The former improves manageability, promoting abstract and high-level interfaces to configuration. The latter enables flexibility of network behavior, e.g., in terms of packet forwarding and modification. Unfortunately, prior works on Software Defined Networking (SDN) do not cover carrier-grade networks, i.e., geographically-distributed networks with hundreds of nodes like Internet Service Provider (ISP) ones. Those networks have special needs: Beyond manageability and flexibility, ISP operators also have to guarantee high scalability (e.g., to support all the Internet prefixes at tens of Points of Presence) and preserve network performance upon failures (e.g., to comply with Service Level Agreements). Moreover, the large scale and geographical distribution of those networks exacerbates SDN challenges, like controller reactivity, controller-to-switch communication and equipment upgrade. Consequently, SDN solutions targeting campuses [2], enterprises [4] and data-centers (DCs) [5], cannot be easily ported to carrier-grade networks. Even approaches designed for wide area and inter-DC networks [6, 7, 8] do not fit. Indeed, they assume that (i) the scale of the network (e.g., number of devices and geographical distances) is small, (ii) scalability and robustness play a more limited role (e.g., because of the small number of destinations [6]), and (iii) the SDN controller may apply some control over traffic sources (e.g., [7]).

Nevertheless, carrier-grade networks would also benefit from an SDN-like approach. Currently, network management (i) relies on protocols with practical limitations, either in terms of expressiveness (as for link-state IGPs, constrained by the adopted shortest-path routing model) or of scalability and overhead (like for MPLS RSVP-TE, based on per-path tunnel signaling); and

PGA: Using Graphs to Express and Automatically Reconcile Network Policies

Chaitan Prakash*, Jeongkeun Lee†, Yoshio Turner*, Joon-Myung Kang†, Aditya Akella*, Sujata Banerjee†, Charles Clark†, Yadi Ma†, Puneet Sharma†, Ying Zhang†

*University of Wisconsin-Madison, †HP Labs, ‡Banyan, †HP Networking

ABSTRACT

Software Defined Networking (SDN) and cloud automation enable a large number of diverse parties (network operators, application admins, tenants/end-users) and control programs (SDN Apps, network services) to generate network policies independently and dynamically. Yet existing policy abstractions and frameworks do not support natural expression and automatic composition of high-level policies from diverse sources. We tackle the open problem of automatic, correct and fast composition of multiple independently specified network policies. We first develop a high-level Policy Graph Abstraction (PGA) that allows network policies to be expressed simply and independently, and leverage the graph structure to detect and resolve policy conflicts efficiently. Besides supporting ACL policies, PGA also models and composes service chaining policies, i.e., the sequence of middleboxes to be traversed, by merging multiple service chain requirements into conflict-free composed chains. Our system validation using a large enterprise network policy dataset demonstrates practical composition times even for very large inputs, with only sub-millisecond runtime latencies.

CCS Concepts

•Networks → Programming interfaces; Network management; Middle boxes / network appliances; Network domains; Network manageability; Programmable networks; Data center networks;

Keywords

Policy graphs; Software-Defined Networks

*This work was performed while at HP Labs.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom
© 2015 ACM. ISBN 978-1-4503-2159-8... \$15.00
DOI: http://dx.doi.org/10.1145/2785956.2787506

1. INTRODUCTION

Computer networks, be they ISPs, enterprise, datacenter, campus or home networks, are governed by high-level policies derived from network-wide requirements. These network policies primarily relate to connectivity, security and performance, and dictate who can have access to what network resources. Further, policies can be static or dynamic (e.g., triggered). Traditionally, network admins translate high level network policies into low level network configuration commands and implement them on network devices, such as switches, routers and specialized network middleboxes (e.g., firewalls, proxies, etc.). The process is largely manual, often internalized by experienced network admins over time. In large organizations, multiple policy sub-domains exist (e.g., server admins, network engineers, DNS admins, different departments) that set their own policies to be applied to the network components they own or manage. Admins and users who share a network have to manually coordinate with each other and check that the growing set of policies do not conflict and match their individually planned high level policies when deployed together.

Given this current status of distributed network policy management, policy changes take a long time to plan and implement (often days to weeks) as careful semi-manual checking with all the relevant policy sub-domains is essential to maintain correctness and consistency. Even so, problems are typically detected only at runtime when users unexpectedly lose connectivity, security holes are exploited, or applications experience performance degradation.

And the situation can get worse as we progress towards more automated network infrastructures, where the number of entities that generate policies independently and dynamically will increase manifold. Examples include SDN applications in enterprise networks, tenants/users of virtualized cloud infrastructures, and Network Functions Virtualization (NFV) environments, details in §2.1.

In all of these settings, it would be ideal to eagerly and automatically detect and resolve conflicts between individual policies, and compose them into a coherent conflict-free policy set, well before the policies are deployed on the physical infrastructure. Further, having a high level policy abstraction and decoupling the policy specification from the underlying physical infrastructure would significantly reduce the burden

Central Control Over Distributed Routing

http://fibbing.net

Stefano Vissicchio*, Olivier Tilmans*, Laurent Vanbever†, Jennifer Rexford‡

* Université catholique de Louvain, † ETH Zurich, ‡ Princeton University
* name.surname@uclouvain.be, † lvanbever@ethz.ch, ‡ jrex@cs.princeton.edu

ABSTRACT

Centralizing routing decisions offers tremendous flexibility, but sacrifices the robustness of distributed protocols. In this paper, we present *Fibbing*, an architecture that achieves both flexibility and robustness through central control over distributed routing. *Fibbing* introduces fake nodes and links into an underlying link-state routing protocol, so that routers compute their own forwarding tables based on the augmented topology. *Fibbing* is expressive, and readily supports flexible load balancing, traffic engineering, and backup routes. Based on high-level forwarding requirements, the *Fibbing* controller computes a compact augmented topology and injects the fake components through standard routing-protocol messages. *Fibbing* works with any unmodified routers speaking OSPF. Our experiments also show that it can scale to large networks with many forwarding requirements, introduces minimal overhead, and quickly reacts to network and controller failures.

CCS Concepts

•Networks → Routing protocols; Network architectures; Programmable networks; Network management;

Keywords

Fibbing; SDN; link-state routing

1. INTRODUCTION

Consider a large IP network with hundreds of devices, including the components shown in Fig. 1a. A set of IP addresses (D_1) see a sudden surge of traffic, from multiple entry points (A , D , and E), that congests a

*S. Vissicchio is a postdoctoral researcher of F.R.S.-FNRS.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom
© 2015 ACM. ISBN 978-1-4503-2159-8... \$15.00
DOI: http://dx.doi.org/10.1145/2785956.2787497

part of the network. As a network operator, you suspect a denial-of-service attack (DoS), but cannot know for sure without inspecting the traffic as it could also be a flash crowd. Your goal is therefore to: (i) isolate the flows destined to these IP addresses, (ii) direct them to a scrubber connected between B and C , in order to "clean" them if needed, and (iii) reduce congestion by load-balancing the traffic on unused links, like (B, E) .

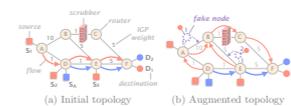


Figure 1: Fibbing can steer the initial forwarding paths (see (a)) for D_1 through a scrubber by adding fake nodes and links (see (b)).

Performing this routine task is very difficult in traditional networks. First, since the middlebox and the destinations are not adjacent to each other, the configuration of multiple devices needs to change. Also, since intra-domain routing is typically based on shortest path algorithms, modifying the routing configuration is likely to impact many other flows not involved in the attack. In Fig. 1a, any attempt to reroute flows to D_1 would also reroute flows to D_2 since they home to the same router. Advertising D_1 from the middlebox would attract the right traffic, but would not necessarily alleviate the congestion, because all D_1 traffic would traverse (and congest) path (A, D, E, B) , leaving (A, B) unused. Well-known Traffic-Engineering (TE) protocols (e.g., MPLS RSVP-TE [1]) could help. Unfortunately, since D_1 traffic enters the network from multiple points, many tunnels (three, on A , D , and E , in our tiny example) would need to be configured and signaled. This increases both control-plane and data-plane overhead.

Software Defined Networking (SDN) could easily solve the problem as it enables centralized and direct control of the forwarding behavior. However, moving away from distributed routing protocols comes at a cost. In-

bandwidth management

Network resources are expensive.

Making the best use of them is key

Network resources are expensive.

Making the best use of them is key, **but hard**

Configuring the network is complex

tons of protocols & mechanisms

Configuration must be adapted frequently

as demands or traffic shift

Lack of router coordination leads to poor utilization

average utilisation of 40-60% [SWAN, SIGCOMM'13]

BwE and DEFO improve network resources utilization

BwE: Flexible, Hierarchical Bandwidth Allocation for WAN Distributed Computing

Alok Kumar
Nikhil Kasinadhuni
Björn Carlin

Sushant Jain
Enrique Cauich Zermeno
Mihai Amarandei-Stavila
Stephen Stuart

Uday Naik
C. Stephen Gunn
Mathieu Robin
Amin Vahdat

Anand Raghuraman
Jing Ai
Aspi Sigantoria

Google Inc.
bwe-sigcomm@google.com

ABSTRACT

WAN bandwidth remains a constrained resource that is economically infeasible to substantially overprovision. Hence, it is important to allocate capacity according to service priority and based on the incremental value of additional allocation. For example, it may be the highest priority for one service to receive 10Gb/s of bandwidth but upon reaching such an allocation, incremental priority may drop sharply favoring allocation to other services. Motivated by the observation that individual flows with fixed priority may not be the ideal basis for bandwidth allocation, we present the design and implementation of Bandwidth Enforcer (BwE), a global, hierarchical bandwidth allocation infrastructure. BwE supports: i) service-level bandwidth allocation following prioritized bandwidth functions where a service can represent an arbitrary collection of flows, ii) independent allocation and delegation policies according to user-defined hierarchy, all accounting for a global view of bandwidth and failure conditions, iii) multi-path forwarding common in traffic-engineered networks, and iv) a central administrative point to override (perhaps faulty) policy during exceptional conditions. BwE has delivered more service-efficient bandwidth utilization and simpler management in production for multiple years.

CCS Concepts

•Networks → Network resources allocation; Network management;

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGCOMM '15 August 17-21, 2015, London, United Kingdom

© 2015 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-3542-3/15/08.

DOI: <http://dx.doi.org/10.1145/2785956.2787478>

Keywords

Bandwidth Allocation; Wide-Area Networks; Software-Defined Network; Max-Min Fair

1. INTRODUCTION

TCP-based bandwidth allocation to individual flows contending for bandwidth on bottleneck links has served the Internet well for decades. However, this model of bandwidth allocation assumes all flows are of equal priority and that all flows benefit equally from any incremental share of available bandwidth. It implicitly assumes a client-server communication model where a TCP flow captures the communication needs of an application communicating across the Internet.

This paper re-examines bandwidth allocation for an important, emerging trend, distributed computing running across dedicated private WANs in support of cloud computing and service providers. Thousands of simultaneous such applications run across multiple global data centers, with thousands of processes in each data center, each potentially maintaining thousands of individual active connections to remote servers. WAN traffic engineering means that site-pair communication follows different network paths, each with different bottlenecks. Individual services have vastly different bandwidth, latency, and loss requirements.

We present a new WAN bandwidth allocation mechanism supporting distributed computing and data transfer. BwE provides work-conserving bandwidth allocation, hierarchical fairness with flexible policy among competing services, and Service Level Objective (SLO) targets that independently account for bandwidth, latency, and loss.

BwE's key insight is that routers are the wrong place to map policy designs about bandwidth allocation onto per-packet behavior. Routers cannot support the scale and complexity of the necessary mappings, often because the semantics of these mappings cannot be captured in individual packets. Instead, following the End-to-End Argument[28], we push all such mapping to the source host machines. Hosts rate limit their outgoing traffic and mark packets using the DSCP field. Routers use the DSCP marking to determine which

A Declarative and Expressive Approach to Control Forwarding Paths in Carrier-Grade Networks

Renaud Hartert *, Stefano Vissicchio *, Pierre Schaus *, Olivier Bonaventure *,
Clarence Filisfilis †, Thomas Telkamp †, Pierre Francois ‡

* Université catholique de Louvain † Cisco Systems, Inc. ‡ IMDEA Networks Institute
*firstname.lastname@uclouvain.be †{cfilfil,thtelkam}@cisco.com ‡ pierre.francois@imdea.org

ABSTRACT

SDN simplifies network management by relying on declarativity (high-level interface) and expressiveness (network flexibility). We propose a solution to support those features while preserving high robustness and scalability as needed in carrier-grade networks. Our solution is based on (i) a two-layer architecture separating connectivity and optimization tasks; and (ii) a centralized optimizer called DEFO, which translates high-level goals expressed almost in natural language into compliant network configurations. Our evaluation on real and synthetic topologies shows that DEFO improves the state of the art by (i) achieving better trade-offs for classic goals covered by previous works, (ii) supporting a larger set of goals (refined traffic engineering and service chaining), and (iii) optimizing large ISP networks in few seconds. We also quantify the gains of our implementation, running Segment Routing on top of IS-IS, over possible alternatives (RSVP-TE and OpenFlow).

CCS Concepts

•Networks → Network architectures; Traffic engineering algorithms; Network management; Routing protocols; •Theory of computation → Constraint and logic programming;

Keywords

SDN; traffic engineering; service chaining; segment routing; MPLS; ISP; optimization

*R. Hartert is a research fellow of F.R.S.-FNRS, and S. Vissicchio is a postdoctoral researcher of F.R.S.-FNRS.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom

© 2015 ACM. ISBN 978-1-4503-3542-3/15/08...\$15.00

DOI: <http://dx.doi.org/10.1145/2785956.2787495>

1. INTRODUCTION

By promising to overcome major problems of traditional per-device network management (e.g., see [1]), centralized architectures enabled by protocols like OpenFlow [2] and segment routing [3] are attracting huge interest from both researchers and operators. Two features are key to this success: declarativity and expressiveness. The former improves manageability, promoting abstractions and high-level interfaces to configuration. The latter enables flexibility of network behavior, e.g., in terms of packet forwarding and modification.

Unfortunately, prior works on Software Defined Networking (SDN) do not cover carrier-grade networks, i.e., geographically-distributed networks with hundreds of nodes like Internet Service Provider (ISP) ones. Those networks have special needs: Beyond manageability and flexibility, ISP operators also have to guarantee high scalability (e.g., to support all the Internet prefixes at tens of Points of Presence) and preserve network performance upon failures (e.g., to comply with Service Level Agreements). Moreover, the large scale and geographical distribution of those networks exacerbates SDN challenges, like controller reactivity, controller-to-switch communication and equipment upgrade. Consequently, SDN solutions targeting campuses [2], enterprises [4] and data-centers (DCs) [5], cannot be easily ported to carrier-grade networks. Even approaches designed for wide area and inter-DC networks [6, 7, 8] do not fit. Indeed, they assume that (i) the scale of the network (e.g., number of devices and geographical distances) is small, (ii) scalability and robustness play a more limited role (e.g., because of the small number of destinations [6]), and (iii) the SDN controller may apply some control over traffic sources (e.g., [7]).

Nevertheless, carrier-grade networks would also benefit from an SDN-like approach. Currently, network management (i) relies on protocols with practical limitations, either in terms of expressiveness (as for link-state IGP, constrained by the adopted shortest-path routing model) or of scalability and overhead (like for MPLS RSVP-TE, based on per-path tunnel signaling); and

BwE and DEFO improve network resources utilization. They do so in **two completely different contexts**

BwE: Flexible, Hierarchical Bandwidth Allocation for WAN Distributed Computing

Alok Kumar
Nikhil Kasinadhuni
Björn Carlin

Sushant Jain
Enrique Cauich Zermeno
Mihai Amarandei-Stavila
Stephen Stuart

Uday Naik
C. Stephen Gunn
Mathieu Robin
Amin Vahdat

Anand Raghuraman
Jing Ai
Aspi Sigantoria

Google Inc.
bwe-sigcomm@google.com

ABSTRACT

WAN bandwidth remains a constrained resource that is economically infeasible to substantially overprovision. Hence, it is important to allocate capacity according to service priority and based on the incremental value of additional allocation. For example, it may be the highest priority for one service to receive 10Gb/s of bandwidth but upon reaching such an allocation, incremental priority may drop sharply favoring allocation to other services. Motivated by the observation that individual flows with fixed priority may not be the ideal basis for bandwidth allocation, we present the design and implementation of Bandwidth Enforcer (BwE), a global, hierarchical bandwidth allocation infrastructure. BwE supports: i) service-level bandwidth allocation following prioritized bandwidth functions where a service can represent an arbitrary collection of flows, ii) independent allocation and delegation policies according to user-defined hierarchy, all accounting for a global view of bandwidth and failure conditions, iii) multi-path forwarding common in traffic-engineered networks, and iv) a central administrative point to override (perhaps faulty) policy during exceptional conditions. BwE has delivered more service-efficient bandwidth utilization and simpler management in production for multiple years.

CCS Concepts

•Networks → Network resources allocation; Network management;

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGCOMM '15 August 17-21, 2015, London, United Kingdom

© 2015 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-3542-3/15/08.

DOI: <http://dx.doi.org/10.1145/2785956.2787478>

Keywords

Bandwidth Allocation; Wide-Area Networks; Software-Defined Network; Max-Min Fair

1. INTRODUCTION

TCP-based bandwidth allocation to individual flows contending for bandwidth on bottleneck links has served the Internet well for decades. However, this model of bandwidth allocation assumes all flows are of equal priority and that all flows benefit equally from any incremental share of available bandwidth. It implicitly assumes a client-server communication model where a TCP flow captures the communication needs of an application communicating across the Internet.

This paper re-examines bandwidth allocation for an important, emerging trend, distributed computing running across dedicated private WANs in support of cloud computing and service providers. Thousands of simultaneous such applications run across multiple global data centers, with thousands of processes in each data center, each potentially maintaining thousands of individual active connections to remote servers. WAN traffic engineering means that site-pair communication follows different network paths, each with different bottlenecks. Individual services have vastly different bandwidth, latency, and loss requirements.

We present a new WAN bandwidth allocation mechanism supporting distributed computing and data transfer. BwE provides work-conserving bandwidth allocation, hierarchical fairness with flexible policy among competing services, and Service Level Objective (SLO) targets that independently account for bandwidth, latency, and loss.

BwE's key insight is that routers are the wrong place to map policy designs about bandwidth allocation onto per-packet behavior. Routers cannot support the scale and complexity of the necessary mappings, often because the semantics of these mappings cannot be captured in individual packets. Instead, following the End-to-End Argument[28], we push all such mapping to the source host machines. Hosts rate limit their outgoing traffic and mark packets using the DSCP field. Routers use the DSCP marking to determine which

A Declarative and Expressive Approach to Control Forwarding Paths in Carrier-Grade Networks

Renaud Hartert *, Stefano Vissicchio *, Pierre Schaus *, Olivier Bonaventure *,
Clarence Filisfilis †, Thomas Telkamp †, Pierre Francois ‡

* Université catholique de Louvain † Cisco Systems, Inc. ‡ IMDEA Networks Institute
*firstname.lastname@uclouvain.be †{cfilfil,thtelkam}@cisco.com ‡ pierre.francois@imdea.org

ABSTRACT

SDN simplifies network management by relying on declarativity (high-level interface) and expressiveness (network flexibility). We propose a solution to support those features while preserving high robustness and scalability as needed in carrier-grade networks. Our solution is based on (i) a two-layer architecture separating connectivity and optimization tasks; and (ii) a centralized optimizer called DEFO, which translates high-level goals expressed almost in natural language into compliant network configurations. Our evaluation on real and synthetic topologies shows that DEFO improves the state of the art by (i) achieving better trade-offs for classic goals covered by previous works, (ii) supporting a larger set of goals (refined traffic engineering and service chaining), and (iii) optimizing large ISP networks in few seconds. We also quantify the gains of our implementation, running Segment Routing on top of IS-IS, over possible alternatives (RSVP-TE and OpenFlow).

CCS Concepts

•Networks → Network architectures; Traffic engineering algorithms; Network management; Routing protocols; •Theory of computation → Constraint and logic programming;

Keywords

SDN; traffic engineering; service chaining; segment routing; MPLS; ISP; optimization

*R. Hartert is a research fellow of F.R.S.-FNRS, and S. Vissicchio is a postdoctoral researcher of F.R.S.-FNRS.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom

© 2015 ACM. ISBN 978-1-4503-3542-3/15/08...\$15.00

DOI: <http://dx.doi.org/10.1145/2785956.2787495>

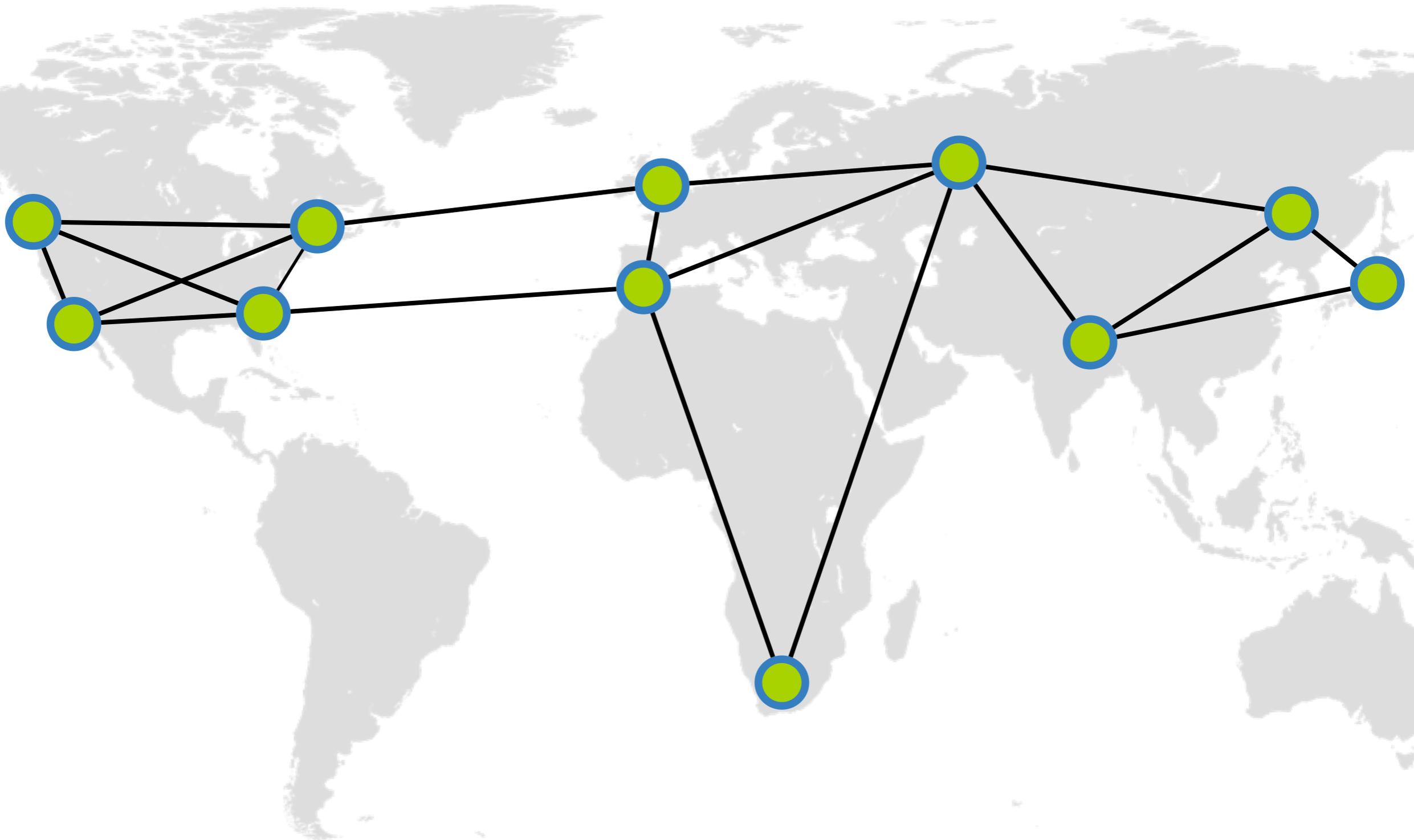
1. INTRODUCTION

By promising to overcome major problems of traditional per-device network management (e.g., see [1]), centralized architectures enabled by protocols like OpenFlow [2] and segment routing [3] are attracting huge interest from both researchers and operators. Two features are key to this success: declarativity and expressiveness. The former improves manageability, promoting abstractions and high-level interfaces to configuration. The latter enables flexibility of network behavior, e.g., in terms of packet forwarding and modification.

Unfortunately, prior works on Software Defined Networking (SDN) do not cover carrier-grade networks, i.e., geographically-distributed networks with hundreds of nodes like Internet Service Provider (ISP) ones. Those networks have special needs: Beyond manageability and flexibility, ISP operators also have to guarantee high scalability (e.g., to support all the Internet prefixes at tens of Points of Presence) and preserve network performance upon failures (e.g., to comply with Service Level Agreements). Moreover, the large scale and geographical distribution of those networks exacerbates SDN challenges, like controller reactivity, controller-to-switch communication and equipment upgrade. Consequently, SDN solutions targeting campuses [2], enterprises [4] and data-centers (DCs) [5], cannot be easily ported to carrier-grade networks. Even approaches designed for wide area and inter-DC networks [6, 7, 8] do not fit. Indeed, they assume that (i) the scale of the network (e.g., number of devices and geographical distances) is small, (ii) scalability and robustness play a more limited role (e.g., because of the small number of destinations [6]), and (iii) the SDN controller may apply some control over traffic sources (e.g., [7]).

Nevertheless, carrier-grade networks would also benefit from an SDN-like approach. Currently, network management (i) relies on protocols with practical limitations, either in terms of expressiveness (as for link-state IGP, constrained by the adopted shortest-path routing model) or of scalability and overhead (like for MPLS RSVP-TE, based on per-path tunnel signaling); and

Wide-Area Networks interconnect
geographically distributed data centers



Wide-Area Networks interconnect
geographically distributed data centers



BwE: Flexible, Hierarchical Bandwidth Allocation for WAN Distributed Computing

Alok Kumar
Nikhil Kasinadhuni
Björn Carlin

Sushant Jain
Enrique Cauich Zermeno
Mihai Amarandei-Stavila
Stephen Stuart

Uday Naik
C. Stephen Gunn
Mathieu Robin
Amin Vahdat

Anand Raghuraman
Jing Ai
Aspi Sigantoria

Google Inc.
bwe-sigcomm@google.com

ABSTRACT

WAN bandwidth remains a constrained resource that is economically infeasible to substantially overprovision. Hence, it is important to allocate capacity according to service priority and based on the incremental value of additional allocation. For example, it may be the highest priority for one service to receive 10Gb/s of bandwidth but upon reaching such an allocation, incremental priority may drop sharply favoring allocation to other services. Motivated by the observation that individual flows with fixed priority may not be the ideal basis for bandwidth allocation, we present the design and implementation of Bandwidth Enforcer (BwE), a global, hierarchical bandwidth allocation infrastructure. BwE supports: i) service-level bandwidth allocation following prioritized bandwidth functions where a service can represent an arbitrary collection of flows, ii) independent allocation and delegation policies according to user-defined hierarchy, all accounting for a global view of bandwidth and failure conditions, iii) multi-path forwarding common in traffic-engineered networks, and iv) a central administrative point to override (perhaps faulty) policy during exceptional conditions. BwE has delivered more service-efficient bandwidth utilization and simpler management in production for multiple years.

CCS Concepts

•Networks → Network resources allocation; Network management;

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGCOMM '15 August 17-21, 2015, London, United Kingdom

© 2015 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-3542-3/15/08.

DOI: <http://dx.doi.org/10.1145/2785956.2787478>

Keywords

Bandwidth Allocation; Wide-Area Networks; Software-Defined Network; Max-Min Fair

1. INTRODUCTION

TCP-based bandwidth allocation to individual flows contending for bandwidth on bottleneck links has served the Internet well for decades. However, this model of bandwidth allocation assumes all flows are of equal priority and that all flows benefit equally from any incremental share of available bandwidth. It implicitly assumes a client-server communication model where a TCP flow captures the communication needs of an application communicating across the Internet.

This paper re-examines bandwidth allocation for an important, emerging trend, distributed computing running across dedicated private WANs in support of cloud computing and service providers. Thousands of simultaneous such applications run across multiple global data centers, with thousands of processes in each data center, each potentially maintaining thousands of individual active connections to remote servers. WAN traffic engineering means that site-pair communication follows different network paths, each with different bottlenecks. Individual services have vastly different bandwidth, latency, and loss requirements.

We present a new WAN bandwidth allocation mechanism supporting distributed computing and data transfer. BwE provides work-conserving bandwidth allocation, hierarchical fairness with flexible policy among competing services, and Service Level Objective (SLO) targets that independently account for bandwidth, latency, and loss.

BwE's key insight is that routers are the wrong place to map policy designs about bandwidth allocation onto per-packet behavior. Routers cannot support the scale and complexity of the necessary mappings, often because the semantics of these mappings cannot be captured in individual packets. Instead, following the End-to-End Argument[28], we push all such mapping to the source host machines. Hosts rate limit their outgoing traffic and mark packets using the DSCP field. Routers use the DSCP marking to determine which

A Declarative and Expressive Approach to Control Forwarding Paths in Carrier-Grade Networks

Renaud Hartert *, Stefano Vissicchio *, Pierre Schaus *, Olivier Bonaventure *,
Clarence Filisfilis †, Thomas Telkamp †, Pierre Francois ‡

* Université catholique de Louvain † Cisco Systems, Inc. ‡ IMDEA Networks Institute
*firstname.lastname@uclouvain.be † {cfilfil, thtelkam}@cisco.com ‡ pierre.francois@imdea.org

ABSTRACT

SDN simplifies network management by relying on declarativity (high-level interface) and expressiveness (network flexibility). We propose a solution to support those features while preserving high robustness and scalability as needed in carrier-grade networks. Our solution is based on (i) a two-layer architecture separating connectivity and optimization tasks; and (ii) a centralized optimizer called DEFO, which translates high-level goals expressed almost in natural language into compliant network configurations. Our evaluation on real and synthetic topologies shows that DEFO improves the state of the art by (i) achieving better trade-offs for classic goals covered by previous works, (ii) supporting a larger set of goals (refined traffic engineering and service chaining), and (iii) optimizing large carrier networks in few seconds. We also quantify the gains of our implementation, running Segment Routing on top of IS-IS, over possible alternatives (RSVP-TE and OpenFlow).

CCS Concepts

•Networks → Network architectures; Traffic engineering algorithms; Network management; Routing protocols; •Theory of computation → Constraint and logic programming;

Keywords

SDN; traffic engineering; service chaining; segment routing; MPLS; ISP; optimization

*R. Hartert is a research fellow of F.R.S.-FNRS, and S. Vissicchio is a postdoctoral researcher of F.R.S.-FNRS.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom

© 2015 ACM. ISBN 978-1-4503-3542-3/15/08...\$15.00

DOI: <http://dx.doi.org/10.1145/2785956.2787495>

1. INTRODUCTION

By promising to overcome major problems of traditional per-device network management (e.g., see [1]), centralized architectures enabled by protocols like OpenFlow [2] and segment routing [3] are attracting huge interest from both researchers and operators. Two features are key to this success: declarativity and expressiveness. The former improves manageability, promoting abstractions and high-level interfaces to configuration. The latter enables flexibility of network behavior, e.g., in terms of packet forwarding and modification.

Unfortunately, prior works on Software Defined Networks (SDN) do not cover carrier-grade networks, i.e., geographically distributed networks with hundreds of nodes like Internet Service Provider (ISP) ones. Those networks have special needs: Beyond manageability and flexibility, ISP operators also have to guarantee high scalability (e.g., to support all the Internet prefixes at tens of Points of Presence) and preserve network performance upon failures (e.g., to comply with Service Level Agreements). Moreover, the large scale and geographical distribution of those networks exacerbates SDN challenges, like controller reactivity, controller-to-switch communication and equipment upgrade. Consequently, SDN solutions targeting campuses [2], enterprises [4] and data-centers (DCs) [5], cannot be easily ported to carrier-grade networks. Even approaches designed for wide area and inter-DC networks [6, 7, 8] do not fit. Indeed, they assume that (i) the scale of the network (e.g., number of devices and geographical distances) is small, (ii) scalability and robustness play a more limited role (e.g., because of the small number of destinations [6]), and (iii) the SDN controller may apply some control over traffic sources (e.g., [7]).

Nevertheless, carrier-grade networks would also benefit from an SDN-like approach. Currently, network management (i) relies on protocols with practical limitations, either in terms of expressiveness (as for link-state IGP, constrained by the adopted shortest-path routing model) or of scalability and overhead (like for MPLS RSVP-TE, based on per-path tunnel signaling); and

Carrier-Grade Networks provide Internet services (often) worldwide



Cogent Network Map

Carrier-Grade Networks provide Internet services (often) worldwide



WAN and CGN differ in terms of scale

	WAN	CGN
# of nodes	$O(10)$	$O(10^2-10^3)$
destinations (forwarding table)	$O(10^3)$	$O(10^6)$

WAN and CGN differ in terms of scale **and control**

	WAN	CGN
# of nodes	$O(10)$	$O(10^2-10^3)$
destinations (forwarding table)	$O(10^3)$	$O(10^6)$
control	end-to-end	network only

Because of these differences,
the two papers differ widely

BwE allocates bandwidth to applications and enforces it hierarchically *starting from the hosts*

BwE: Flexible, Hierarchical Bandwidth Allocation for WAN Distributed Computing

Alok Kumar
Nikhil Kasinadhuni
Björn Carlin

Sushant Jain
Enrique Cauich Zermeno
Mihai Amarandei-Stavila
Stephen Stuart

Uday Naik
C. Stephen Gunn
Mathieu Robin
Amin Vahdat

Anand Raghuraman
Jing Ai
Aspi Siganporia

Google Inc.
bwe-sigcomm@google.com

ABSTRACT

WAN bandwidth remains a constrained resource that is economically infeasible to substantially overprovision. Hence, it is important to allocate capacity according to service priority and based on the incremental value of additional allocation. For example, it may be the highest priority for one service to receive 10Gb/s of bandwidth but upon reaching such an allocation, incremental priority may drop sharply favoring allocation to other services. Motivated by the observation that individual flows with fixed priority may not be the ideal basis for bandwidth allocation, we present the design and implementation of Bandwidth Enforcer (BwE), a global, hierarchical bandwidth allocation infrastructure. BwE supports: i) service-level bandwidth allocation following prioritized bandwidth functions where a service can represent an arbitrary collection of flows, ii) independent allocation and delegation policies according to user-defined hierarchy, all accounting for a global view of bandwidth and failure conditions, iii) multi-path forwarding common in traffic-engineered networks, and iv) a central administrative point to override (perhaps faulty) policy during exceptional conditions. BwE has delivered more service-efficient bandwidth utilization and simpler management in production for multiple years.

CCS Concepts

•Networks → Network resources allocation; Network management;

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGCOMM '15 August 17-21, 2015, London, United Kingdom

© 2015 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-3542-3/15/08.

DOI: <http://dx.doi.org/10.1145/2785956.2787478>

Keywords

Bandwidth Allocation; Wide-Area Networks; Software-Defined Network; Max-Min Fair

1. INTRODUCTION

TCP-based bandwidth allocation to individual flows contending for bandwidth on bottleneck links has served the Internet well for decades. However, this model of bandwidth allocation assumes all flows are of equal priority and that all flows benefit equally from any incremental share of available bandwidth. It implicitly assumes a client-server communication model where a TCP flow captures the communication needs of an application communicating across the Internet.

This paper re-examines bandwidth allocation for an important, emerging trend, distributed computing running across dedicated private WANs in support of cloud computing and service providers. Thousands of simultaneous such applications run across multiple global data centers, with thousands of processes in each data center, each potentially maintaining thousands of individual active connections to remote servers. WAN traffic engineering means that site-pair communication follows different network paths, each with different bottlenecks. Individual services have vastly different bandwidth, latency, and loss requirements.

We present a new WAN bandwidth allocation mechanism supporting distributed computing and data transfer. BwE provides work-conserving bandwidth allocation, hierarchical fairness with flexible policy among competing services, and Service Level Objective (SLO) targets that independently account for bandwidth, latency, and loss.

BwE's key insight is that routers are the wrong place to map policy designs about bandwidth allocation onto per-packet behavior. Routers cannot support the scale and complexity of the necessary mappings, often because the semantics of these mappings cannot be captured in individual packets. Instead, following the End-to-End Argument[28], we push all such mapping to the source host machines. Hosts rate limit their outgoing traffic and mark packets using the DSCP field. Routers use the DSCP marking to determine which

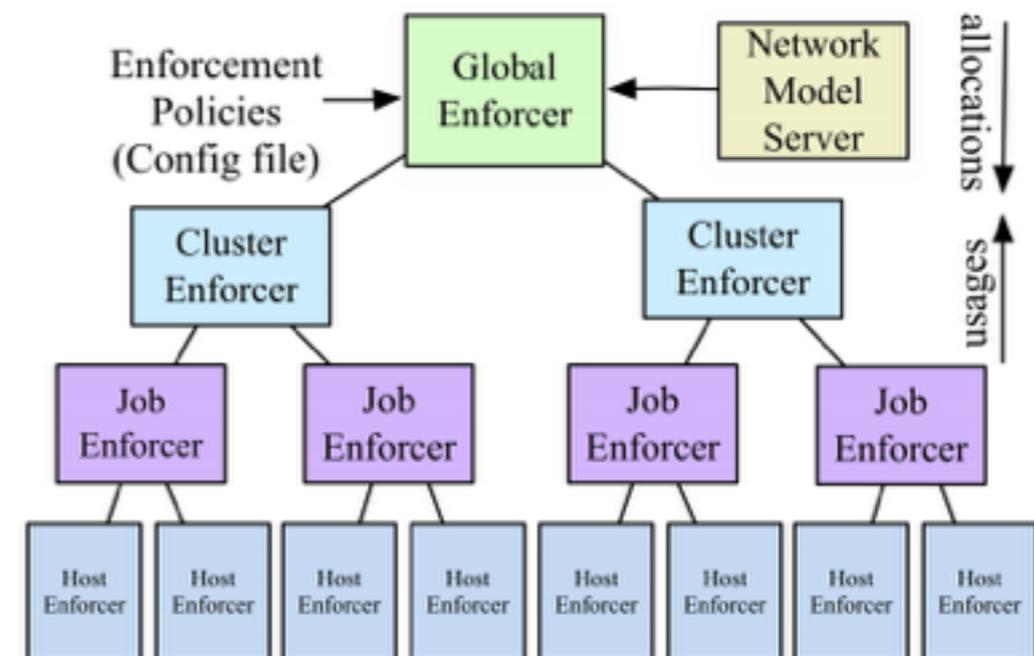


Figure 5: BwE Architecture.

DEFO computes paths compliant with given constraints and programs them *in the network*

function	DSL syntax	semantics
max load	<code>d.load</code>	maximum load of any link in $F(d)$
max delay	<code>d.delay</code>	maximum delay of source-destination paths in $F(d)$
deviations	<code>d.deviations</code>	number of deviations from connectivity paths in $F(d)$
traversal	<code>d passThrough S</code>	true if $F(d)$ crosses any node in S , false otherwise
sequencing	<code>d passThrough S₁ then S₂ ... then S_k</code>	true if $F(d)$ sequentially crosses nodes in $S_1 \dots S_k$
avoid	<code>d avoid S</code>	true if no node in S is also in $F(d)$, false otherwise

```
var MaxLoad = max(for(l<-topology.links){yield l.load})
val goal = new Goal(topology){ minimize(MaxLoad) }
```

A Declarative and Expressive Approach to Control Forwarding Paths in Carrier-Grade Networks

Renaud Hartert ^{*}, Stefano Vissicchio ^{*}, Pierre Schaus ^{*}, Olivier Bonaventure ^{*},
Clarence Filisfil [†], Thomas Telkamp [‡], Pierre Francois [‡]

^{*} Université catholique de Louvain [†] Cisco Systems, Inc. [‡] IMDEA Networks Institute
^{*} firstname.lastname@uclouvain.be [†] {cfilfil,thtelkam}@cisco.com [‡] pierre.francois@imdea.org

ABSTRACT

SDN simplifies network management by relying on declarativity (high-level interface) and expressiveness (network flexibility). We propose a solution to support those features while preserving high robustness and scalability as needed in carrier-grade networks. Our solution is based on (i) a two-layer architecture separating connectivity and optimization tasks; and (ii) a centralized optimizer called DEFO, which translates high-level goals expressed almost in natural language into compliant network configurations. Our evaluation on real and synthetic topologies shows that DEFO improves the state of the art by (i) achieving better trade-offs for classic goals covered by previous works, (ii) supporting a larger set of goals (refined traffic engineering and service chaining), and (iii) optimizing large ISP networks in few seconds. We also quantify the gains of our implementation, running Segment Routing on top of IS-IS, over possible alternatives (RSVP-TE and OpenFlow).

CCS Concepts

•Networks → Network architectures; Traffic engineering algorithms; Network management; Routing protocols; •Theory of computation → Constraint and logic programming;

Keywords

SDN; traffic engineering; service chaining; segment routing; MPLS; ISP; optimization

^{*}R. Hartert is a research fellow of F.R.S.-FNRS, and S. Vissicchio is a postdoctoral researcher of F.R.S.-FNRS.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom
© 2015 ACM. ISBN 978-1-4503-3542-3/15/08...\$15.00
DOI: <http://dx.doi.org/10.1145/2785956.2787495>

1. INTRODUCTION

By promising to overcome major problems of traditional per-device network management (e.g., see [1]), centralized architectures enabled by protocols like OpenFlow [2] and segment routing [3] are attracting huge interest from both researchers and operators. Two features are key to this success: declarativity and expressiveness. The former improves manageability, promoting abstractions and high-level interfaces to configuration. The latter enables flexibility of network behavior, e.g., in terms of packet forwarding and modification.

Unfortunately, prior works on Software Defined Networking (SDN) do not cover carrier-grade networks, i.e., geographically-distributed networks with hundreds of nodes like Internet Service Provider (ISP) ones. Those networks have special needs: Beyond manageability and flexibility, ISP operators also have to guarantee high scalability (e.g., to support all the Internet prefixes at tens of Points of Presence) and preserve network performance upon failures (e.g., to comply with Service Level Agreements). Moreover, the large scale and geographical distribution of those networks exacerbates SDN challenges, like controller reactivity, controller-to-switch communication and equipment upgrade. Consequently, SDN solutions targeting campuses [2], enterprises [4] and data-centers (DCs) [5], cannot be easily ported to carrier-grade networks. Even approaches designed for wide area and inter-DC networks [6, 7, 8] do not fit. Indeed, they assume that (i) the scale of the network (e.g., number of devices and geographical distances) is small, (ii) scalability and robustness play a more limited role (e.g., because of the small number of destinations [6]), and (iii) the SDN controller may apply some control over traffic sources (e.g., [7]).

Nevertheless, carrier-grade networks would also benefit from an SDN-like approach. Currently, network management (i) relies on protocols with practical limitations, either in terms of expressiveness (as for link-state IGPs, constrained by the adopted shortest-path routing model) or of scalability and overhead (like for MPLS RSVP-TE, based on per-path tunnel signaling); and

SDN track @SIGCOMM'15

BwE: Flexible, Hierarchical Bandwidth Allocation for WAN Distributed Computing

Alok Kur
Nikhil Kasin
Björn Ca

A Declarative and Expressive Approach to Control Forwarding Paths in Carrier-Grade Networks

Renaud Hartert^{*}, Stefano Vissicchio^{*}, Pierre Schaus^{*}, Olivier Bonaventure^{*},
Clarence Filisfilis[†], Thomas Telkamp[†], Pierre Francois[‡]

^{*} Université catholique de Louvain [†] Cisco Systems, Inc. [‡] IMDEA Networks Institute
^{*} firstname.lastname@uclouvain.be [†] {cfilisfil,htelkam}@cisco.com [‡] pierre.francois@imdea.org

ABSTRACT

WAN bandwidth requirements are economically infeasible to satisfy. It is important to allocate bandwidth efficiently and based on traffic conditions. For example, to receive a service, it is important to allocate bandwidth favoring allocation of resources that provide the best service. BwE has designed a global, hierarchical bandwidth allocation framework that supports (i) setting prioritized bandwidth reservations, (ii) delegating bandwidth allocation to network devices, and (iii) engineering network devices to override (perhaps) bandwidth reservations. BwE has designed a global, hierarchical bandwidth allocation framework that supports (i) setting prioritized bandwidth reservations, (ii) delegating bandwidth allocation to network devices, and (iii) engineering network devices to override (perhaps) bandwidth reservations.

CCS Concepts

•Networks → Network architectures; •Networks → Network management; •Theory of computation → Constraint and logic programming;

Keywords

SDN; traffic engineering; service chaining; segment routing; MPLS; ISP; optimization

^{*}R. Hartert is a research fellow of F.R.S.-FNRS, and S. Vissicchio is a postdoctoral researcher of F.R.S.-FNRS.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom

© 2015 ACM. ISBN 978-1-4503-3542-3/15/08...\$15.00

DOI: <http://dx.doi.org/10.1145/2785956.2787495>

PGA: Using Graphs to Express and Automatically Reconcile Network Policies

Chaitan Prakash^{1,2}, Jeongkeun Lee¹, Yoshio Turner², Joon-Myung Kang¹, Aditya Akella¹,
Sujata Banerjee¹, Charles Clark¹, Yadi Ma¹, Puneet Sharma¹, Ying Zhang¹

¹University of Wisconsin-Madison, ¹HP Labs, ²Banyan, ¹HP Networking

ABSTRACT

Software Defined Networking (SDN) and cloud automation enable a large number of diverse parties (network operators, application admins, tenants/end-users) and control programs (SDN Apps, network services) to generate network policies independently and dynamically. Yet existing policy abstractions and frameworks do not support natural expression and automatic composition of high-level policies from diverse sources. We tackle the open problem of automatic, correct and fast composition of multiple independently specified network policies. We first develop a high-level Policy Graph Abstraction (PGA) that allows network policies to be expressed simply and independently, and leverage the graph structure to detect and resolve policy conflicts efficiently. Besides supporting ACL policies, PGA also models and composes service chaining policies, i.e., the sequence of middleboxes to be traversed, by merging multiple service chain requirements into conflict-free composed chains. Our system validation using a large enterprise network policy dataset demonstrates practical composition times even for very large inputs, with only sub-millisecond runtime latencies.

CCS Concepts

•Networks → Programming interfaces; •Network management; •Middle boxes / network appliances; •Network domains; •Network manageability; •Programmable networks; •Data center networks;

Keywords

Policy graphs; Software-Defined Networks

^{*}This work was performed while at HP Labs.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom

© 2015 ACM. ISBN 978-1-4503-3542-3/15/08...\$15.00

DOI: <http://dx.doi.org/10.1145/2785956.2787506>

network policies

Central Control Over Distributed Routing

<http://fibbing.net>

Stefano Vissicchio^{*}, Olivier Tilmans^{*}, Laurent Vanbever[†], Jennifer Rexford[‡]

^{*} Université catholique de Louvain, [†] ETH Zurich, [‡] Princeton University
^{*} name.surname@uclouvain.be, [†]lvvanbever@ethz.ch, [‡]jrex@cs.princeton.edu

ABSTRACT

Centralizing routing decisions offers tremendous flexibility, but sacrifices the robustness of distributed protocols. In this paper, we present *Fibbing*, an architecture that achieves both flexibility and robustness through central control over distributed routing. *Fibbing* introduces fake nodes and links into an underlying link-state routing protocol, so that routers compute their own forwarding tables based on the augmented topology. *Fibbing* is expressive, and readily supports flexible load balancing, traffic engineering, and backup routes. Based on high-level forwarding requirements, the *Fibbing* controller computes a compact augmented topology and injects the fake components through standard routing-protocol messages. *Fibbing* works with any unmodified routers speaking OSPF. Our experiments also show that it can scale to large networks with many forwarding requirements, introduces minimal overhead, and quickly reacts to network and controller failures.

CCS Concepts

•Networks → Routing protocols; •Network architectures; •Programmable networks; •Network management;

Keywords

Fibbing; SDN; link-state routing

1. INTRODUCTION

Consider a large IP network with hundreds of devices, including the components shown in Fig. 1a. A set of IP addresses (D_1) see a sudden surge of traffic, from multiple entry points (A , D , and E), that congests a

^{*}S. Vissicchio is a postdoctoral researcher of F.R.S.-FNRS.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom

© 2015 ACM. ISBN 978-1-4503-3542-3/15/08...\$15.00

DOI: <http://dx.doi.org/10.1145/2785956.2787497>

part of the network. As a network operator, you suspect a denial-of-service attack (DoS), but cannot know for sure without inspecting the traffic as it could also be a flash crowd. Your goal is therefore to: (i) isolate the flows destined to these IP addresses, (ii) direct them to a scrubber connected between B and C , in order to “clean” them if needed, and (iii) reduce congestion by load-balancing the traffic on unused links, like (B, E) .

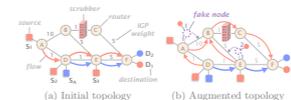


Figure 1: Fibbing can steer the initial forwarding paths (see (a)) for D_1 through a scrubber by adding fake nodes and links (see (b)).

Performing this routine task is very difficult in traditional networks. First, since the middlebox and the destinations are not adjacent to each other, the configuration of multiple devices needs to change. Also, since intra-domain routing is typically based on shortest path algorithms, modifying the routing configuration is likely to impact many other flows not involved in the attack. In Fig. 1a, any attempt to reroute flows to D_1 would also reroute flows to D_2 since they home to the same router. Advertising D_1 from the middlebox would attract the right traffic, but would not necessarily alleviate the congestion, because *all* D_1 traffic would traverse (and congest) path (A, D, E, B) , leaving (A, B) unused. Well-known Traffic-Engineering (TE) protocols (e.g., MPLS RSVP-TE [1]) could help. Unfortunately, since D_1 traffic enters the network from multiple points, many tunnels (three, on A , D , and E , in our tiny example) would need to be configured and signaled. This increases both control-plane and data-plane overhead.

Software Defined Networking (SDN) could easily solve the problem as it enables centralized and direct control of the forwarding behavior. However, moving away from distributed routing protocols comes at a cost. In-

Networks often rely on forwarding policies,
especially enterprise and campus networks

Policies are often defined
by different people

Customer relationship
administrator

Company network
administrator

Forwarding policy

Customer relationship
administrator

Company network
administrator

Forwarding policy

Customer relationship
administrator

Only marketing employee can use
a CRM application, using port 7000.
Traffic must go via a Load-Balancer first.

Company network
administrator

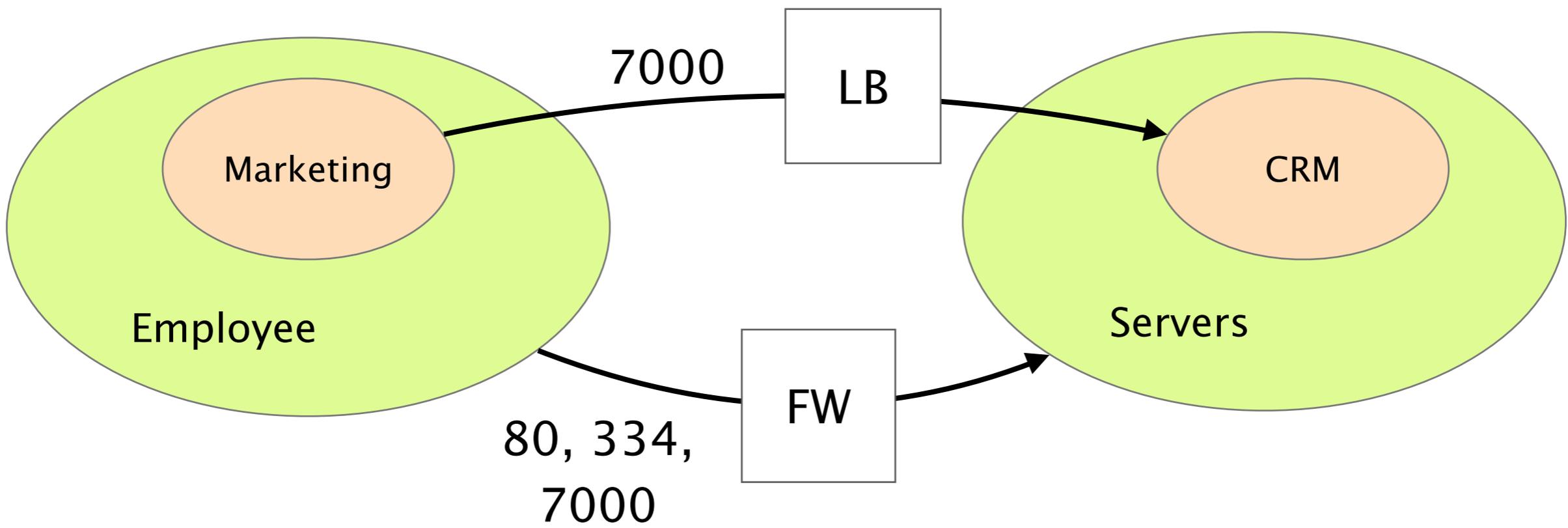
Forwarding policy

Customer relationship
administrator

Only marketing employee can use
a CRM application, using port 7000.
Traffic must go via a Load-Balancer first.

Company network
administrator

Any employee can only access servers
using port 80, 334 and 7000.
All traffic must go via a Firewall first.



Customer relationship
administrator

Only marketing employee can use
a CRM application, using port 7000.
Traffic must go via a Load-Balancer first.

Company network
administrator

Any employee can only access servers
using port 80, 334 and 7000.
All traffic must go via a Firewall first.

What about marketing employees' traffic to the CRM?

Customer relationship administrator

Only marketing employee can use a CRM application, using port 7000. Traffic must go via a Load-Balancer first.

Company network administrator

Any employee can only access servers using port 80, 334 and 7000. All traffic must go via a Firewall first.

It must go through a LB

Customer relationship
administrator

Only marketing employee can use
a CRM application, using port 7000.

Traffic must go via a Load-Balancer first.

Company network
administrator

Any employee can only access servers
using port 80, 334 and 7000.

All traffic must go via a Firewall first.

It must go through a LB *and* a Firewall

Customer relationship
administrator

Only marketing employee can use
a CRM application, using port 7000.

Traffic must go via a Load-Balancer first.

Company network
administrator

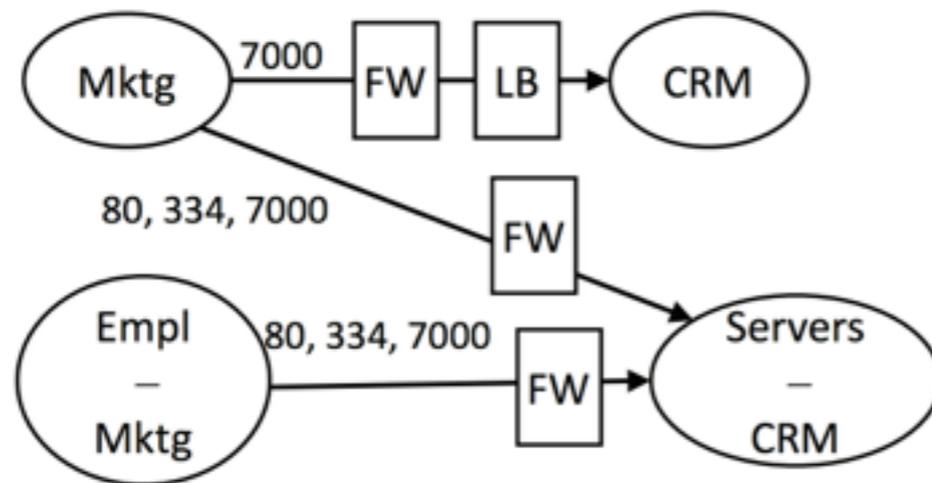
Any employee can only access servers
using port 80, 334 and 7000.

All traffic must go via a Firewall first.

Composing different policies is tricky
as we must reason on the **joint intent**

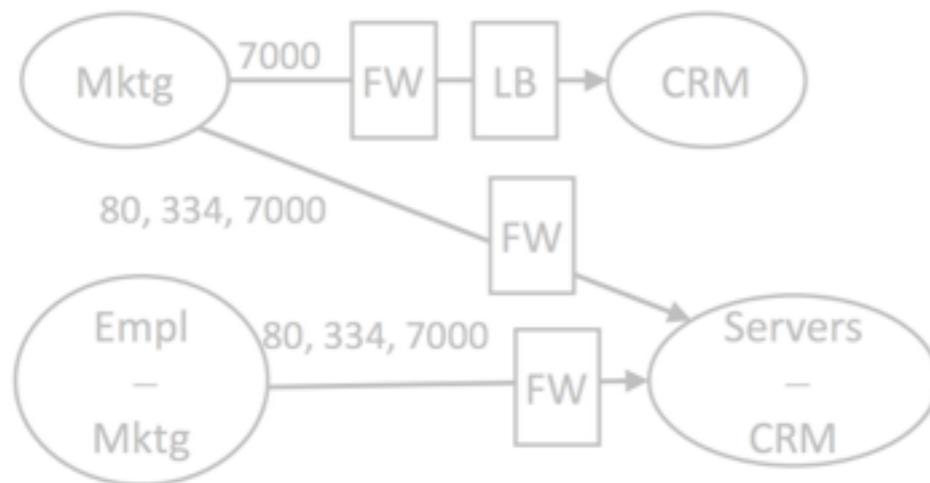
PGA uses a graph abstraction to specify policies

PGA high-level policy



PGA uses a graph abstraction to specify policies and automatically composes and compiles them

PGA high-level policy



PGA framework

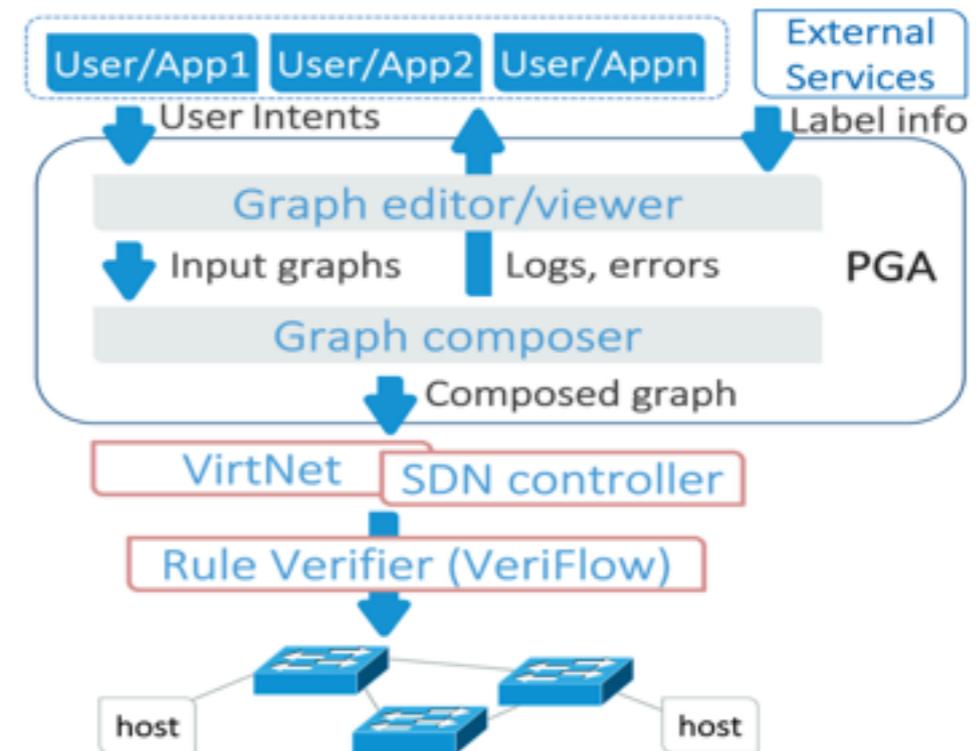


Figure 2: PGA system architecture.

SDN track @SIGCOMM'15

BwE: Flexible, Hierarchical Bandwidth Allocation for WAN Distributed Computing

Alok Kur
Nikhil Kasin
Björn Ca

A Declarative and Expressive Approach to Control Forwarding Paths in Carrier-Grade Networks

Renaud Hartert^{*}, Stefano Vissicchio^{*}, Pierre Schaus^{*}, Olivier Bonaventure^{*},
Clarence Filisfilis[†], Thomas Telkamp[†], Pierre Francois[‡]

^{*} Université catholique de Louvain [†] Cisco Systems, Inc. [‡] IMDEA Networks Institute
^{*} firstname.lastname@uclouvain.be [†] {cfilifil,thtelkam}@cisco.com [‡] pierre.francois@imdea.org

ABSTRACT

WAN bandwidth requirements are economically infeasible to satisfy. It is important to allocate bandwidth efficiently and based on traffic conditions. For example, to receive a service such as an allocation, favoring allocation in favor of service that is the ideal basis for design and implementation. BwE supports: (i) setting prioritized bandwidth reservation that is an arbitrary configuration and delegation of bandwidth, (ii) accounting for bandwidth conditions, (iii) an engineered network to override (perhaps) bandwidth conditions. BwE has delimitation and simple utilization and simple.

CCS Concepts

•Networks → Network architectures; Traffic engineering algorithms; Network management; Routing protocols; •Theory of computation → Constraint and logic programming;

Keywords

SDN; traffic engineering; service chaining; segment routing; MPLS; ISP; optimization
^{*}R. Hartert is a research fellow of F.R.S.-FNRS, and S. Vissicchio is a postdoctoral researcher of F.R.S.-FNRS.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom
© 2015 ACM. ISBN 978-1-4503-3542-3/15/08...\$15.00
DOI: http://dx.doi.org/10.1145/2785956.2787495

ABSTRACT

SDN simplifies network management by relying on declarativity (high-level interface) and expressiveness (network flexibility). We propose a solution to support those features while preserving high robustness and scalability as needed in carrier-grade networks. Our solution is based on (i) a two-layer architecture separating connectivity and optimization tasks; and (ii) a centralized optimizer called DEFO, which translates high-level goals expressed almost in natural language into compliant network configurations. Our evaluation on real and synthetic topologies shows that DEFO improves the state of the art by (i) achieving better trade-offs for classic goals covered by previous works, (ii) supporting a larger set of goals (refined traffic engineering and service chaining), and (iii) optimizing large ISP networks in few seconds. We also quantify the gains of our implementation, running Segment Routing on top of IS-IS, over possible alternatives (RSVP-TE and OpenFlow).

CCS Concepts

•Networks → Network architectures; Traffic engineering algorithms; Network management; Routing protocols; •Theory of computation → Constraint and logic programming;

Keywords

SDN; traffic engineering; service chaining; segment routing; MPLS; ISP; optimization

^{*}R. Hartert is a research fellow of F.R.S.-FNRS, and S. Vissicchio is a postdoctoral researcher of F.R.S.-FNRS.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom
© 2015 ACM. ISBN 978-1-4503-3542-3/15/08...\$15.00
DOI: http://dx.doi.org/10.1145/2785956.2787495

1. INTRODUCTION

By promising to overcome major problems of traditional per-device network management (e.g., see [1]), centralized architectures enabled by protocols like OpenFlow [2] and segment routing [3] are attracting huge interest from both researchers and operators. Two features are key to this success: declarativity and expressiveness. The former improves manageability, promoting abstract and high-level interfaces to configuration. The latter enables flexibility of network behavior, e.g., in terms of packet forwarding and modification.

Unfortunately, prior works on Software Defined Networking (SDN) do not cover carrier-grade networks, i.e., geographically-distributed networks with hundreds of nodes like Internet Service Provider (ISP) ones. Those networks have special needs: Beyond manageability and flexibility, ISP operators also have to guarantee huge scalability (e.g., to support all the Internet prefixes at tens of Points of Presence) and preserve network performance upon failures (e.g., to comply with Service Level Agreements). Moreover, the large scale and geographical distribution of those networks exacerbates SDN challenges, like controller reactivity, controller-to-switch communication and equipment upgrade. Consequently, SDN solutions targeting campuses [2], enterprises [4] and data-centers (DCs) [5], cannot be easily ported to carrier-grade networks. Even approaches designed for wide area and inter-DC networks [6, 7, 8] do not fit. Indeed, they assume that (i) the scale of the network (e.g., number of devices and geographical distances) is small, (ii) scalability and robustness play a more limited role (e.g., because of the small number of destinations [6]), and (iii) the SDN controller may apply some control over traffic sources (e.g., [7]).

Nevertheless, carrier-grade networks would also benefit from an SDN-like approach. Currently, network management (i) relies on protocols with practical limitations, either in terms of expressiveness (as for link-state IGPs, constrained by the adopted shortest-path routing model) or of scalability and overhead (like for MPLS RSVP-TE, based on per-path tunnel signaling); and

PGA: Using Graphs to Express and Automatically Reconcile Network Policies

Chaitan Prakash^{1*}, Jeongkeun Lee¹, Yoshio Turner^{2*}, Joon-Myung Kang¹, Aditya Akella³,
Sujata Banerjee⁴, Charles Clark⁵, Yadi Ma¹, Puneet Sharma¹, Ying Zhang¹
¹University of Wisconsin-Madison, ¹HP Labs, ²Banyan, ³HP Networking

ABSTRACT

Software Defined Networking (SDN) and cloud automation enable a large number of diverse parties (network operators, application admins, tenants/end-users) and control programs (SDN Apps, network services) to generate network policies independently and dynamically. Yet existing policy abstractions and frameworks do not support natural expression and automatic composition of high-level policies from diverse sources. We tackle the open problem of automatic, correct and fast composition of multiple independently specified network policies. We first develop a high-level Policy Graph Abstraction (PGA) that allows network policies to be expressed simply and independently, and leverage the graph structure to detect and resolve policy conflicts efficiently. Besides supporting ACL policies, PGA also models and composes service chaining policies, i.e., the sequence of middleboxes to be traversed, by merging multiple service chain requirements into conflict-free composed chains. Our system validation using a large enterprise network policy dataset demonstrates practical composition times even for very large inputs, with only sub-millisecond runtime latencies.

CCS Concepts

•Networks → Programming interfaces; Network management; Middle boxes / network appliances; Network domains; Network manageability; Programmable networks; Data center networks;

Keywords

Policy graphs; Software-Defined Networks

^{*}This work was performed while at HP Labs.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom
© 2015 ACM. ISBN 978-1-4503-3542-3/15/08...\$15.00
DOI: http://dx.doi.org/10.1145/2785956.2787506

1. INTRODUCTION

Computer networks, be they ISPs, enterprise, datacenter, campus or home networks, are governed by high-level policies derived from network-wide requirements. These network policies primarily relate to connectivity, security and performance, and dictate who can have access to what network resources. Further, policies can be static or dynamic (e.g., triggered). Traditionally, network admins translate high level network policies into low level network configuration commands and implement them on network devices, such as switches, routers and specialized network middleboxes (e.g., firewalls, proxies, etc.). The process is largely manual, often internalized by experienced network admins over time. In large organizations, multiple policy sub-domains exist (e.g., server admins, network engineers, DNS admins, different departments) that set their own policies to be applied to the network components they own or manage. Admins and users who share a network have to manually coordinate with each other and check that the growing set of policies do not conflict and match their individually planned high level policies when deployed together.

Given this current status of distributed network policy management, policy changes take a long time to plan and implement (often days to weeks) as careful semi-manual checking with all the relevant policy sub-domains is essential to maintain correctness and consistency. Even so, problems are typically detected only at runtime when users unexpectedly lose connectivity, security holes are exploited, or applications experience performance degradation.

And the situation can get worse as we progress towards more automated network infrastructures, where the number of entities that generate policies independently and dynamically will increase manifold. Examples include SDN applications in enterprise networks, tenants/users of virtualized cloud infrastructures, and Network Functions Virtualization (NFV) environments, details in §2.1.

In all of these settings, it would be ideal to eagerly and automatically detect and resolve conflicts between individual policies, and compose them into a coherent conflict-free policy set, well before the policies are deployed on the physical infrastructure. Further, having a high level policy abstraction and decoupling the policy specification from the underlying physical infrastructure would significantly reduce the burden

Central Control Over Distributed Routing

http://fibbing.net

Stefano Vissicchio^{*}, Olivier Tilmans^{*}, Laurent Vanbever[†], Jennifer Rexford[‡]

^{*} Université catholique de Louvain, [†] ETH Zurich, [‡] Princeton University
^{*} name.surname@uclouvain.be, [†]lvnbever@ethz.ch, [‡]jrex@cs.princeton.edu

ABSTRACT

Centralizing routing decisions offers tremendous flexibility, but sacrifices the robustness of distributed protocols. In this paper, we present *Fibbing*, an architecture that achieves both flexibility and robustness through central control over distributed routing. *Fibbing* introduces fake nodes and links into an underlying link-state routing protocol, so that routers compute their own forwarding tables based on the augmented topology. *Fibbing* is expressive, and readily supports flexible load balancing, traffic engineering, and backup routes. Based on high-level forwarding requirements, the *Fibbing* controller computes a compact augmented topology and injects the fake components through standard routing-protocol messages. *Fibbing* works with any unmodified routers speaking OSPF. Our experiments also show that it can scale to large networks with many forwarding requirements, introduces minimal overhead, and quickly reacts to network and controller failures.

CCS Concepts

•Networks → Routing protocols; Network architectures; Programmable networks; Network management;

Keywords

Fibbing; SDN; link-state routing

1. INTRODUCTION

Consider a large IP network with hundreds of devices, including the components shown in Fig. 1a. A set of IP addresses (D_1) see a sudden surge of traffic, from multiple entry points (A , D , and E), that congests a

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom
© 2015 ACM. ISBN 978-1-4503-3542-3/15/08...\$15.00
DOI: http://dx.doi.org/10.1145/2785956.2787497

part of the network. As a network operator, you suspect a denial-of-service attack (DoS), but cannot know for sure without inspecting the traffic as it could also be a flash crowd. Your goal is therefore to: (i) isolate the flows destined to these IP addresses, (ii) direct them to a scrubber connected between B and C , in order to “clean” them if needed, and (iii) reduce congestion by load-balancing the traffic on unused links, like (B , E).

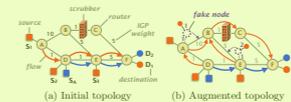


Figure 1: Fibbing can steer the initial forwarding paths (see (a)) for D_1 through a scrubber by adding fake nodes and links (see (b)).

Performing this routine task is very difficult in traditional networks. First, since the middlebox and the destinations are not adjacent to each other, the configuration of multiple devices needs to change. Also, since intra-domain routing is typically based on shortest path algorithms, modifying the routing configuration is likely to impact many other flows not involved in the attack. In Fig. 1a, any attempt to reroute flows to D_1 would also reroute flows to D_2 since they home to the same router. Advertising D_1 from the middlebox would attract the right traffic, but would not necessarily alleviate the congestion, because all D_1 traffic would traverse (and congest) path (A , D , E , B), leaving (A , B) unused. Well-known Traffic-Engineering (TE) protocols (e.g., MPLS RSVP-TE [1]) could help. Unfortunately, since D_1 traffic enters the network from multiple points, many tunnels (three, on A , D , and E , in our tiny example) would need to be configured and signaled. This increases both control-plane and data-plane overhead.

Software Defined Networking (SDN) could easily solve the problem as it enables centralized and direct control of the forwarding behavior. However, moving away from distributed routing protocols comes at a cost. In-

programmability

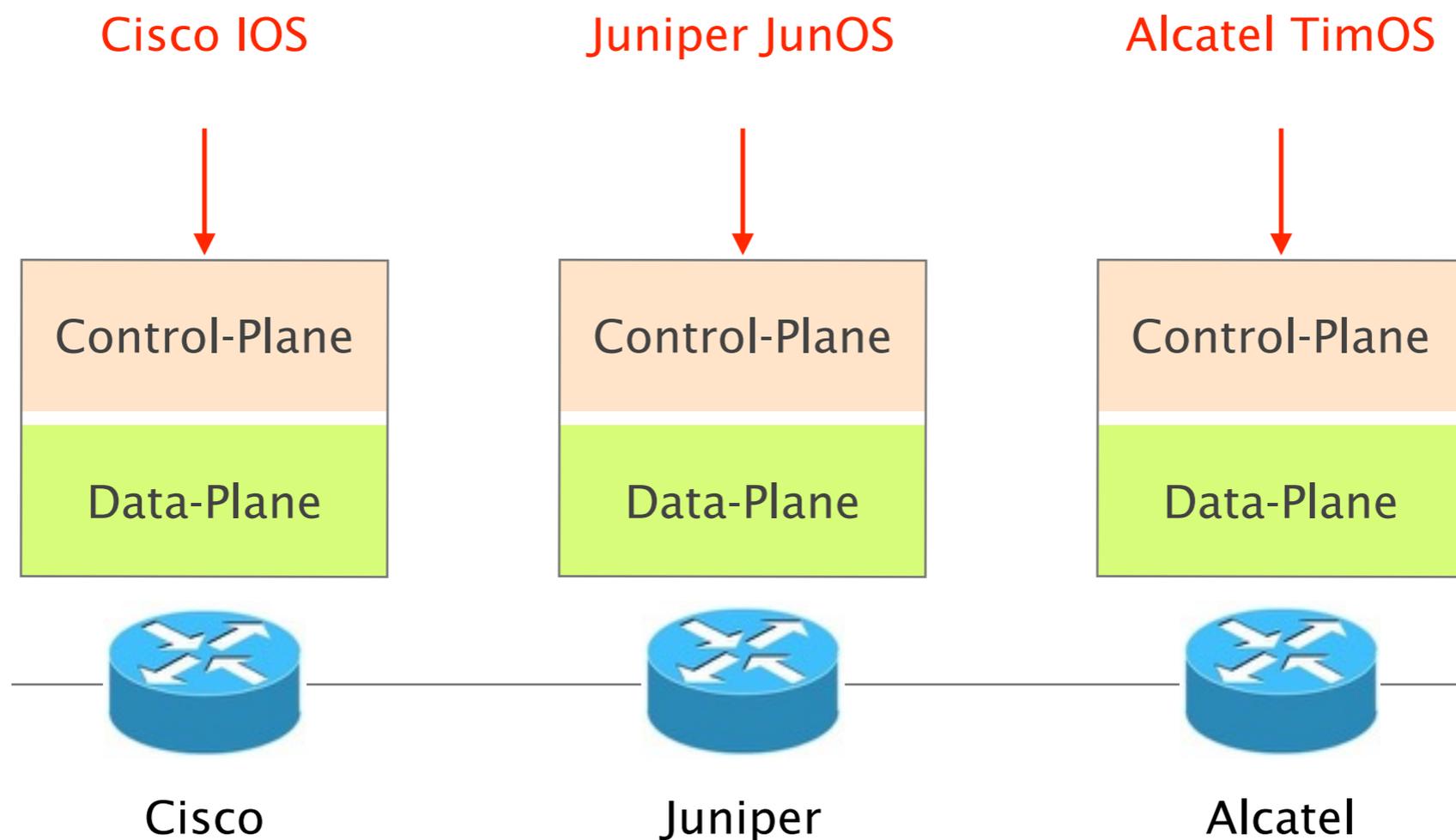
SDN is great, but we need compatible devices
(which aren't deployed in most networks)

Wouldn't it be great to program
an **existing network** “à la SDN”?

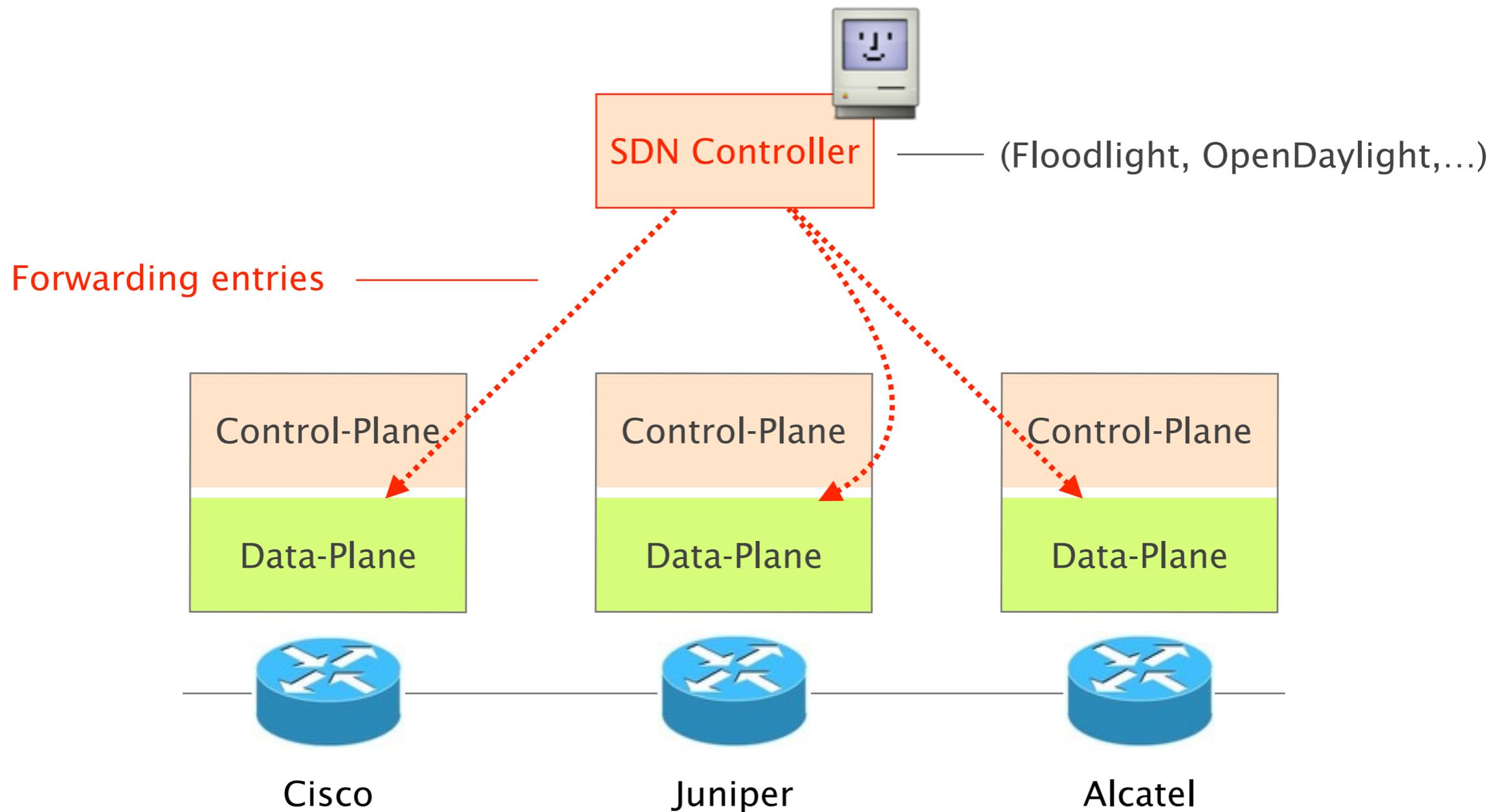
Wouldn't it be great to program
an existing network "à la SDN"?

what does it mean?

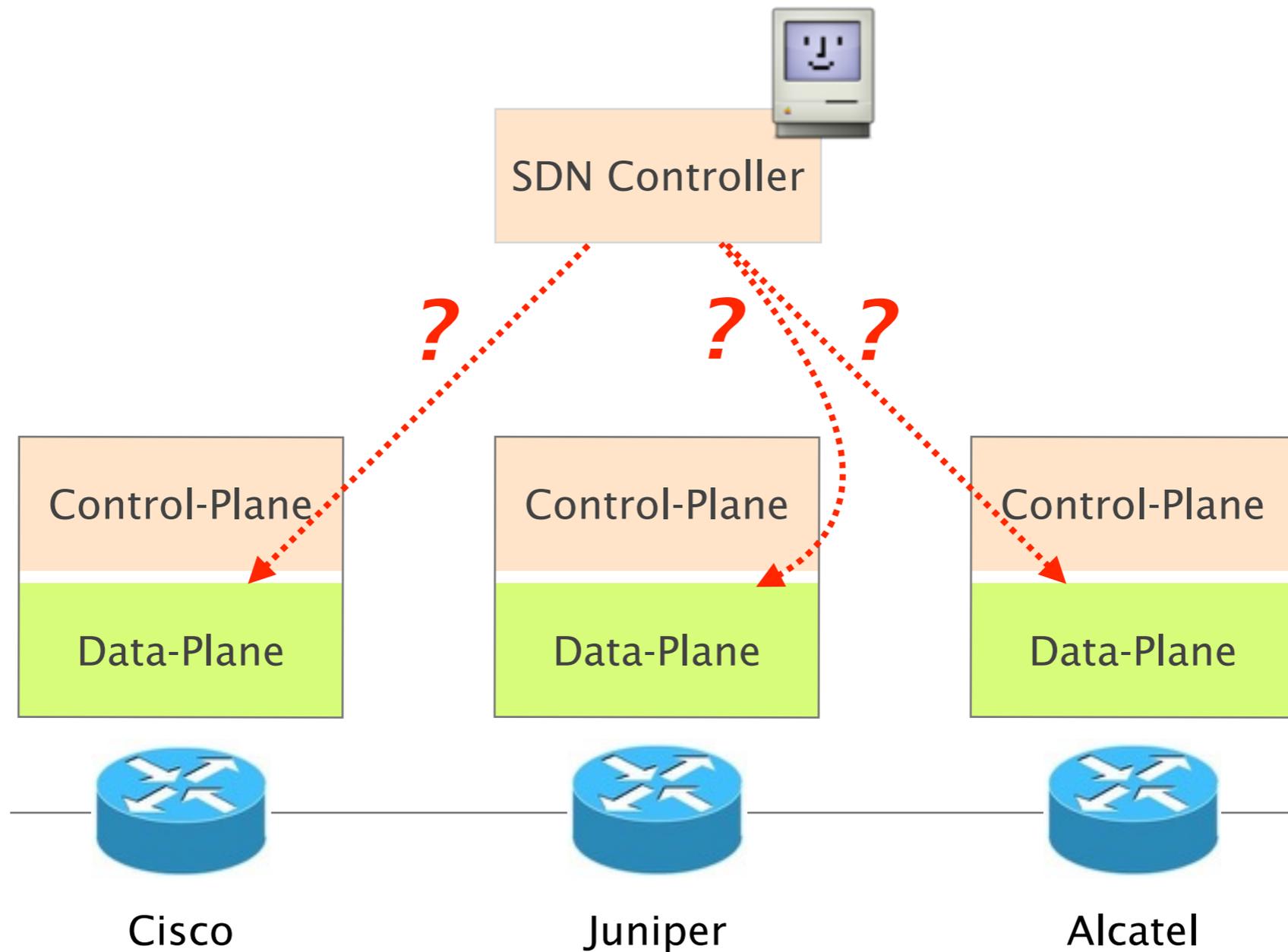
Instead of **configuring** a network using configuration “languages”...



...program it from a central SDN controller



For that, we need an API
that *any* router can understand



Routing protocols are perfect candidates to act as such API

- messages are standardized
all routers speak the same language
- behaviors are well-defined
e.g., shortest-path routing
- implementations are widely available
nearly all routers support OSPF

Fibbing

Fibbing

= lying

Fibbing

to **control** router's forwarding table

Given a set of forwarding entries
to install network-wide

Given a set of forwarding entries
to install network-wide,

Fibbing generates **fake routing messages**
which trick routers into computing the
appropriate forwarding entries.

Given a set of forwarding entries
to install network-wide,

Fibbing generates fake routing messages
which trick routers into computing the
appropriate forwarding entries.

In a way that is **scalable and robust**

SDN track @SIGCOMM'15

BwE: Flexible, Hierarchical Bandwidth Allocation for WAN Distributed Computing

Alok Kur
Nikhil Kasin
Björn Ca

A Declarative and Expressive Approach to Control Forwarding Paths in Carrier-Grade Networks

Renaud Hartert^{*}, Stefano Vissicchio^{*}, Pierre Schaus^{*}, Olivier Bonaventure^{*},
Clarence Filisfilis[†], Thomas Telkamp[†], Pierre Francois[‡]

^{*} Université catholique de Louvain [†] Cisco Systems, Inc. [‡] IMDEA Networks Institute
^{*} firstname.lastname@uclouvain.be [†] {cifsfil,thtelkam}@cisco.com [‡] pierre.francois@irdea.org

ABSTRACT
WAN bandwidth requirements are economically infeasible. It is important to allocate bandwidth efficiently and based on traffic. For example, service to receive it such an allocation, favoring allocation reservation that is the ideal basis for design and implementation. BwE supports (i) setting prioritized bandwidth reservation, (ii) an arbitrary circuit and delegation archy, all accounting for network conditions, (iii) an engineered network to override (perhaps) bandwidth allocations. BwE has del utilization and simple.

CCS Concepts
•Networks → Network architectures; Traffic engineering algorithms; Network management; Routing protocols; •Theory of computation → Constraint and logic programming;

Keywords
SDN; traffic engineering; service chaining; segment routing; MPLS; ISP; optimization
^{*}R. Hartert is a research fellow of F.R.S.-FNRS, and S. Vissicchio is a postdoctoral researcher of F.R.S.-FNRS.

CCS Concepts
•Networks → Network architectures; Traffic engineering algorithms; Network management; Routing protocols; •Theory of computation → Constraint and logic programming;

Keywords
SDN; traffic engineering; service chaining; segment routing; MPLS; ISP; optimization
^{*}R. Hartert is a research fellow of F.R.S.-FNRS, and S. Vissicchio is a postdoctoral researcher of F.R.S.-FNRS.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom
© 2015 ACM. ISBN 978-1-4503-3515-0... \$15.00
DOI: http://dx.doi.org/10.1145/2785956.2787495

PGA: Using Graphs to Express and Automatically Reconcile Network Policies

Chaitan Prakash^{1,2*}, Jeongkeun Lee¹, Yoshio Turner^{2,3*}, Joon-Myung Kang¹, Aditya Akella⁴,
Sujata Banerjee¹, Charles Clark¹, Yadi Ma¹, Puneet Sharma¹, Ying Zhang¹
¹University of Wisconsin-Madison, ¹HP Labs, ²Banyan, ³HP Networking

ABSTRACT

Software Defined Networking (SDN) and cloud automation enable a large number of diverse parties (network operators, application admins, tenants/end-users) and control programs (SDN Apps, network services) to generate network policies independently and dynamically. Yet existing policy abstractions and frameworks do not support natural expression and automatic composition of high-level policies from diverse sources. We tackle the open problem of automatic, correct and fast composition of multiple independently specified network policies. We first develop a high-level Policy Graph Abstraction (PGA) that allows network policies to be expressed simply and independently, and leverage the graph structure to detect and resolve policy conflicts efficiently. Besides supporting ACL policies, PGA also models and composes service chaining policies, i.e., the sequence of middleboxes to be traversed, by merging multiple service chain requirements into conflict-free composed chains. Our system validation using a large enterprise network policy dataset demonstrates practical composition times even for very large inputs, with only sub-millisecond runtime latencies.

CCS Concepts

•Networks → Programming interfaces; Network management; Middle boxes / network appliances; Network domains; Network manageability; Programmable networks; Data center networks;

Keywords

Policy graphs; Software-Defined Networks

^{*}This work was performed while at HP Labs.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom
© 2015 ACM. ISBN 978-1-4503-3515-0... \$15.00
DOI: http://dx.doi.org/10.1145/2785956.2787506

1. INTRODUCTION

Computer networks, be they ISPs, enterprise, datacenter, campus or home networks, are governed by high-level policies derived from network-wide requirements. These network policies primarily relate to connectivity, security and performance, and dictate who can have access to what network resources. Further, policies can be static or dynamic (e.g., triggered). Traditionally, network admins translate high level network policies into low level network configuration commands and implement them on network devices, such as switches, routers and specialized network middleboxes (e.g., firewalls, proxies, etc.). The process is largely manual, often internalized by experienced network admins over time. In large organizations, multiple policy sub-domains exist (e.g., server admins, network engineers, DNS admins, different departments) that set their own policies to be applied to the network components they own or manage. Admins and users who share a network have to manually coordinate with each other and check that the growing set of policies do not conflict and match their individually planned high level policies when deployed together.

Given this current status of distributed network policy management, policy changes take a long time to plan and implement (often days to weeks) as careful semi-manual checking with all the relevant policy sub-domains is essential to maintain correctness and consistency. Even so, problems are typically detected only at runtime when users unexpectedly lose connectivity, security holes are exploited, or applications experience performance degradation.

And the situation can get worse as we progress towards more automated network infrastructures, where the number of entities that generate policies independently and dynamically will increase manifold. Examples include SDN applications in enterprise networks, tenants/users of virtualized cloud infrastructures, and Network Functions Virtualization (NFV) environments, details in §2.1.

In all of these settings, it would be ideal to eagerly and automatically detect and resolve conflicts between individual policies, and compose them into a coherent conflict-free policy set, well before the policies are deployed on the physical infrastructure. Further, having a high level policy abstraction and decoupling the policy specification from the underlying physical infrastructure would significantly reduce the burden

Central Control Over Distributed Routing

http://fibbing.net

Stefano Vissicchio^{*}, Olivier Tilmans^{*}, Laurent Vanbever[†], Jennifer Rexford[‡]
^{*} Université catholique de Louvain, [†] ETH Zurich, [‡] Princeton University
^{*} name.surname@uclouvain.be, lvanbever@ethz.ch, jrex@cs.princeton.edu

ABSTRACT

Centralizing routing decisions offers tremendous flexibility, but sacrifices the robustness of distributed protocols. In this paper, we present *Fibbing*, an architecture that achieves both flexibility and robustness through central control over distributed routing. *Fibbing* introduces fake nodes and links into an underlying link-state routing protocol, so that routers compute their own forwarding tables based on the augmented topology. *Fibbing* is expressive, and readily supports flexible load balancing, traffic engineering, and backup routes. Based on high-level forwarding requirements, the *Fibbing* controller computes a compact augmented topology and injects the fake components through standard routing-protocol messages. *Fibbing* works with any unmodified routers speaking OSPF. Our experiments also show that it can scale to large networks with many forwarding requirements, introduces minimal overhead, and quickly reacts to network and controller failures.

CCS Concepts

•Networks → Routing protocols; Network architectures; Programmable networks; Network management;

Keywords

Fibbing; SDN; link-state routing

1. INTRODUCTION

Consider a large IP network with hundreds of devices, including the components shown in Fig. 1a. A set of IP addresses (D_1) see a sudden surge of traffic, from multiple entry points (A , D , and E), that congests a

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom
© 2015 ACM. ISBN 978-1-4503-3515-0... \$15.00
DOI: http://dx.doi.org/10.1145/2785956.2787497

part of the network. As a network operator, you suspect a denial-of-service attack (DoS), but cannot know for sure without inspecting the traffic as it could also be a flash crowd. Your goal is therefore to: (i) isolate the flows destined to these IP addresses, (ii) direct them to a scrubber connected between B and C , in order to “clean” them if needed, and (iii) reduce congestion by load-balancing the traffic on unused links, like (B , E).

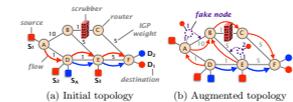


Figure 1: Fibbing can steer the initial forwarding paths (see (a)) for D_1 through a scrubber by adding fake nodes and links (see (b)).

Performing this routine task is very difficult in traditional networks. First, since the middlebox and the destinations are not adjacent to each other, the configuration of multiple devices needs to change. Also, since intra-domain routing is typically based on shortest path algorithms, modifying the routing configuration is likely to impact many other flows not involved in the attack. In Fig. 1a, any attempt to reroute flows to D_1 would also reroute flows to D_2 since they home to the same router. Advertising D_1 from the middlebox would attract the right traffic, but would not necessarily alleviate the congestion, because all D_1 traffic would traverse (and congest) path (A , D , E , B), leaving (A , B) unused. Well-known Traffic-Engineering (TE) protocols (e.g., MPLS RSVP-TE [1]) could help. Unfortunately, since D_1 traffic enters the network from multiple points, many tunnels (three, on A , D , and E , in our tiny example) would need to be configured and signaled. This increases both control-plane and data-plane overhead.

Software Defined Networking (SDN) could easily solve the problem as it enables centralized and direct control of the forwarding behavior. However, moving away from distributed routing protocols comes at a cost. In-

bandwidth management

network policies

programmability

One more thing...

P4: Programming Protocol-Independent Packet Processors

Pat Bosshart[†], Dan Daly^{*}, Glen Gibb[†], Martin Izzard[†], Nick McKeown[‡], Jennifer Rexford^{**}, Cole Schlesinger^{**}, Dan Talayco[†], Amin Vahdat[‡], George Varghese[§], David Walker^{**}
[†]Barefoot Networks ^{*}Intel [‡]Stanford University ^{**}Princeton University [§]Google [§]Microsoft Research

ABSTRACT

P4 is a high-level language for programming protocol-independent packet processors. P4 works in conjunction with SDN control protocols like OpenFlow. In its current form, OpenFlow explicitly specifies protocol headers on which it operates. This set has grown from 12 to 41 fields in a few years, increasing the complexity of the specification while still not providing the flexibility to add new headers. In this paper we propose P4 as a strawman proposal for how OpenFlow should evolve in the future. We have three goals: (1) Reconfigurability in the field: Programmers should be able to change the way switches process packets once they are deployed. (2) Protocol independence: Switches should not be tied to any specific network protocols. (3) Target independence: Programmers should be able to describe packet-processing functionality independently of the specifics of the underlying hardware. As an example, we describe how to use P4 to configure a switch to add a new hierarchical label.

1. INTRODUCTION

Software-Defined Networking (SDN) gives operators programmatic control over their networks. In SDN, the control plane is physically separate from the forwarding plane, and one control plane controls multiple forwarding devices. While forwarding devices could be programmed in many ways, having a common, open, vendor-agnostic interface (like OpenFlow) enables a control plane to control forwarding devices from different hardware and software vendors.

Version	Date	Header Fields
OF 1.0	Dec 2009	12 fields (Ethernet, TCP/IPv4)
OF 1.1	Feb 2011	15 fields (MPLS, inter-table metadata)
OF 1.2	Dec 2011	36 fields (ARP, ICMP, IPv6, etc.)
OF 1.3	Jun 2012	40 fields
OF 1.4	Oct 2013	41 fields

Table 1: Fields recognized by the OpenFlow standard

The OpenFlow interface started simple, with the abstraction of a single table of rules that could match packets on a dozen header fields (e.g., MAC addresses, IP addresses, protocol, TCP/UDP port numbers, etc.). Over the past five years, the specification has grown increasingly more complicated (see Table 1), with many more header fields and

multiple stages of rule tables, to allow switches to expose more of their capabilities to the controller.

The proliferation of new header fields shows no signs of stopping. For example, data-center network operators increasingly want to apply new forms of packet encapsulation (e.g., NVGRE, VXLAN, and STT), for which they resort to deploying software switches that are easier to extend with new functionality. Rather than repeatedly extending the OpenFlow specification, we argue that future switches should support flexible mechanisms for parsing packets and matching header fields, allowing controller applications to leverage these capabilities through a common, open interface (i.e., a new “OpenFlow 2.0” API). Such a general, extensible approach would be simpler, more elegant, and more future-proof than today’s OpenFlow 1.x standard.

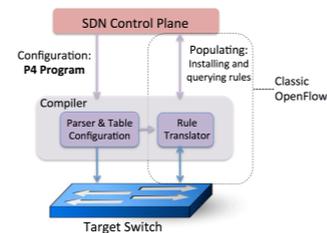


Figure 1: P4 is a language to configure switches.

Recent chip designs demonstrate that such flexibility can be achieved in custom ASICs at terabit speeds [1, 2, 3]. Programming this new generation of switch chips is far from easy. Each chip has its own low-level interface, akin to microcode programming. In this paper, we sketch the design of a higher-level language for Programming Protocol-independent Packet Processors (P4). Figure 1 shows the relationship between P4—used to configure a switch, telling it how packets are to be processed—and existing APIs (such as OpenFlow) that are designed to populate the forwarding tables in fixed function switches. P4 raises the level of abstraction for programming the network, and can serve as a

Learn how to:

- adapt the forwarding logic of a SDN device
- define your very own OpenFlow protocol!

Best of CCR session, Thu 20

Software Defined Networks

SIGCOMM'15 Topic Preview



Laurent Vanbever

www.vanbever.eu

Wishing you every success
in your future SDN research!