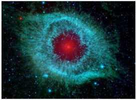
Verifiable Network Paths for the *Nebula* Data Plane

Antonio R. Nicolosi
nicolosi@cs.stevens.edu
Stevens Institute of Technology

Based on joint work with J.Naous (MIT), M. Walfish, M. Miller, A.Seehra (UT-Austin), and D.Mazières (Stanford)

Outline

- Project *Nebula*
- *Nebula* Control/Data Plane (NVENT/NDP)
- Path Verification in NDP: Mechanism Details



Project *Nebula*

Nebula—Motivation: Trustworthy Cloud Computing

- Realizing olden-golden ‘computing utility’
- Why didn’t it happen in the 60’s?
  - Computing technology (HW / OS / SW); HCI; Networking
- Today: Lots of progress, but still inadequate n/w
  - ✓ Pervasive, mobile, broadband connectivity
  - ✗ Five 9’s availability / reliability
  - ✗ In general, assurances other than raw reachability
- And tomorrow?
  - ☞ Future-proofing via extensibility / evolvability

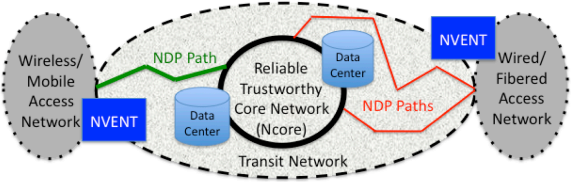
The *Nebula* Vision

Make cloud computing trustworthy

Elaborating a bit:

Provide secure, highly available, and robust communication services to critical applications in the emerging cloud and mobile environment

Overview of the *Nebula* Architecture



Three components:

- NCore: *Nebula* Core network
- NVENT: *Nebula* Virtual & Extensible Networking Techniques
- NDP: *Nebula* Data Plane

Enabling the *Nebula* Vision

Secure, highly available, and robust communication

- Ncore, NVENT, and NDP tackle above challenge from *complimentary* and *redundant* angles
- *E.g.,* availability and robustness
  - NCore *tolerates failures* of core routers
  - NVENT+NDP enable *path diversity*

NVENT+NDP

Q: How do NVENT and NDP enable path diversity?

- NVENT allows parties to express routing preferences and retrieve suitable paths
  - *E.g.*, “Need ≥ 3 node-disjoint paths from *A* to *B*”
- NDP constrains the network paths that data packets actually take

NVENT+NDP ‘thesis’

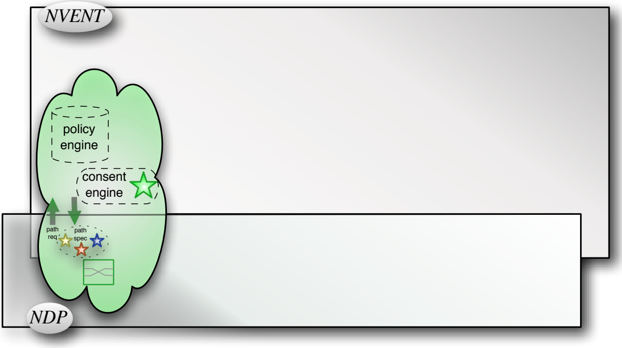
Policy Routing + Path Verification together provide meaningful assurances about network traffic

NVENT/NDP Interface

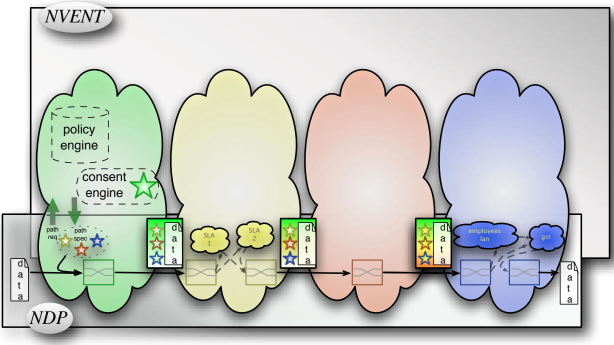
Main principles:

- Separate *decision-making* from *enforcement*
  - Policy decisions in (evolvable) control plane
  - Enforcement in high-speed data plane
- Establish n/w paths prior to communication
  - ☞ Crucially, only negligible state overhead at forwarders

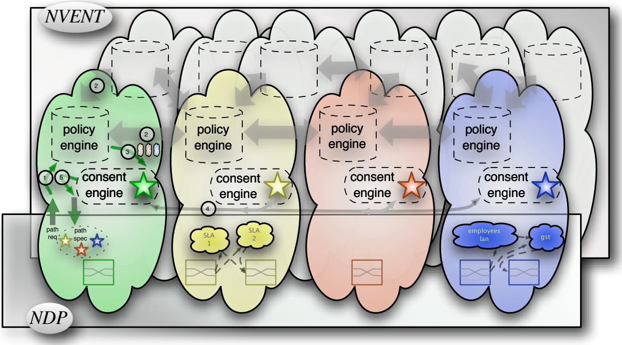
NVENT/NDP Interface (cont'd)



NDP Forwarding: Overview



Outline of NVENT Routing



NDP Forwarding: Main Challenge

Path Verification

Assume an adversarial, decentralized, and high-speed environment. How can a forwarder verify, upon arrival of a packet, that the packet followed an approved network path?

Our approach

1. Path Consent: Before communication, all nodes on path approve its usage (based on policy)
2. Path Compliance: On pkt ingress, can ascertain that path is approved, and pkt is following path

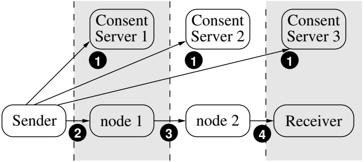
Path Verification in NDP

- Map *path consent* and *path compliance* to cryptographic tokens (MAC's):
  - PoC: *Proof of Consent*
  - PoP: *Proof of Path*
- PoCs minted in control plane (*consent engines*) and checked in data plane
  - Based on symmetric keys shared within a realm (AS)
- PoPs minted by upstream forwarders and checked by downstream forwarders
  - Based on symmetric keys derived via NIDH and SCNs

Naming in NDP

- NDP realms use self-certifying names (SCNs)
  - Realm name is a (short) PK, generated by node itself
  - ☞ No need for a central naming authority
- NDP nodes use non-interactive Diffie-Hellman (NIDH) to establish pairwise PoP keys $k_{i,j}$'s
  - Node in realm N_i uses its realm's secret key to derive shared key $k_{i,j}$ simply from realm N_j's *name*
- Realm names are 'multiplexed' using tags
  - Opaque identifiers whose meaning is local to realm
  - *E.g,* specific actions to perform on packet upon arrival
  - 'Generalized' MPLS label of sort

Path Verification in NDP (cont'd)



P	N_0	N_1	N_2	N_3
V_1	$A_1 \oplus \mathrm{PoP}_{0,1}$			
V_2	$A_2 \oplus \mathrm{PoP}_{0,2}$			
V_3	$A_3 \oplus \mathrm{PoP}_{0,3}$			
	Payload			

2

N_0	N_1	N_2	N_3
$A_1 \oplus \mathrm{PoP}_{0,1}$			
$A_2 \oplus \mathrm{PoP}_{0,2} \oplus \mathrm{PoP}_{1,2}$			
$A_3 \oplus \mathrm{PoP}_{0,3} \oplus \mathrm{PoP}_{1,3} \oplus \mathrm{PoP}_{2,3}$			
Payload			

4

NDP Header

- Two main parts: path P and verifiers V_j's
- Sender (N_0) initializes V_j's with PoCs and PoPs
- Each N_i checks its verifier (V_i) and updates downstream verifiers (V_j for $j > i$)
  - Checking V_i ensures both path consent (via PoC) and interim path compliance (via the PoPs)
  - Updating PoPs in V_j ($j > i$) "tells" N_j that packet has gone through N_i (enabling N_j to check compliance)

Path Verification in NDP: Costs

- Space overhead: $\approx 20\%$
  - Average header: ≈ 250 bytes
  - Average packet size: $\approx 1,300$ bytes
- Hardware cost: $\approx 2\times$ IP router
  - Gate count on NetFPGA: IP $8.7M$, NDP-like $13.4M$
  - NDP-forwarding good-put: $\approx 80\%$ of IP

Summary

- *Nebula*'s vision: Trustworthy cloud computing
- Evolvability and assurance in NVENT+NDP
- Securing n/w forwarding w/ verifiable paths

Caveats / Open Problems

- *Path compliance* doesn't protect pkt's future
  - Feasible to encrypt/decrypt at each hop (*i.e.,* ON)?
- *P. compliance* can't prove where pkt *didn't* go
  - Preventing surreptitious tunneling by nodes on path?
- Cheaper verification via probabilistic checking?
  - Or are NDP assurances all-or-nothing?
- Withholding consent and net-neutrality
  - Is transparency enough to foster consumer choice?
- Privacy implications of full paths in headers

Acknowledgments

- *Nebula* is supported by the NSF under its *Future Internet Architecture* program
  - All opinions reported are those of the author and do not necessarily reflect the views of the NSF

Thank You!

Questions?

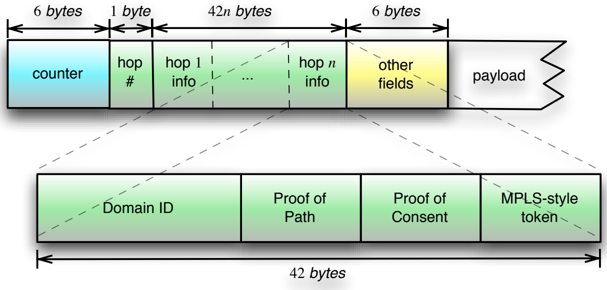
STEVENS INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY

The *Nebula* Team

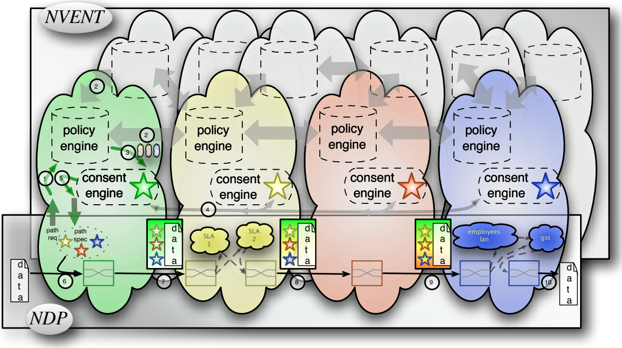
Researcher	Expertise	NEBULA Focus
Tom Anderson	Distributed Systems, Architecture	NCore
Ken Birman*	Reliable Distributed Systems	All
Matthew Caesar	Reliable Distributed Systems	NCore
Douglas Comer*	Architecture, Protocols	All
Chase Cotton	Reliable Routers	NCore
Michael Freedman	Security, Distributed Systems	NVENT
Andreas Haeberlen	Architecture	NVENT
Zack Ives	Distributed Databases	NVENT
Arvind Krishnamurthy	Distributed Systems	NCore
William Lehr	Economics, Architecture	Economics
Boon Thau Loo	Protocol Verification, Security	NVENT
David Mazieres	Security	NDP
Antonio Nicolosi	Cryptography	NDP
Jonathan Smith*	Architecture, Security	All
Ion Stoica	Architecture	NDP
Robbert van Renesse	Reliable Distributed Systems	NVENT
Michael Walfish	Network Architecture	NDP
Hakim Weatherspoon	Architecture, Reliable Routers	NCore
Christopher Yoo	Regulation	Regulation

The *Nebula* Team

NDP Header


6 bytes
1 byte
42n bytes
6 bytes
counter
hop #
hop 1 info
...
hop n info
other fields
payload
Domain ID
Proof of Path
Proof of Consent
MPLS-style token
42 bytes

NVENT+NDP


NVENT
policy
engine
consent
engine
policy
engine
consent
engine
policy
engine
consent
engine
policy
engine
consent
engine
path
req
path
spec
SLA
1
SLA
2
employees
lan
gst
data
NDP