Protocol Virtualization through Dynamic Network Stacks

J. Heuschkel#, I. Schweizer#, T. Zimmermann*, K. Wehrle*, M. Mühlhäuser#

# Telecooperation, Computer Science, TU Darmstadt
* Chair of Communication and Distributed Systems, RWTH Aachen
Agenda

- Motivation
- Possible Applications
- Architecture
- Evaluation and Open Challenges
- Conclusion
Motivation

Application
- Transport
- Network
- Data Link
- Physical
Possible Applications

- End to End SDN
  
  Enabled through DyNS
Possible Applications

- Protocol transformation
- Just in time support for new protocols
- Transparent encryption

Use DCCP

Use encryption
Possible Applications

- Multipath / Multihome
  - Different network interface without specialized protocols
  - Different stacks per interface
- Seamless interface switching

Diagram:
- Nodes connected with lines showing network connections
- Labels: UDP/IPv6, TCP/IPv4
Key Ideas

- **Dynamic Network Stack (DyNS):**
  - Decoupling of network protocols and applications
    - Protocol Virtualization
  - Extend the power of SDN to the end nodes
  - Create and manage different network stacks per application
  - Decide best stack at runtime
  - Enables switching between stacks at runtime
Architecture

Any application (e.g. Dropbox)

Virtual network interface

Network stacks

Optional protocol analysis

Contains requirements and constrains
- from developer
- from traffic engineering

Allowed better control from core network

Any application (e.g. Dropbox)

Virtual network interface

Network stacks

Optional protocol analysis

Contains requirements and constrains
- from developer
- from traffic engineering

Allowed better control from core network
Architecture

Start

receive packet

parser reads packet

parser extracts meta information and payload

give information to manager

is new app?

need new protocol?

need new NW interface?

create new stack

send packet

End
Decision

- Local:
  - Protocol information through Parser
  - App Requirements through App DB

- Centralized:
  - Central optimization (SDN) through Management Interface
Evaluation

- Tested in a single core VM
- 500 - 300k packets processed per second (avg 100k)
- Up to 10.1 Mbit/s
Evaluation

- Protocol switch always possible
- No performance lost while switching
Open Challenges

- Finalize the idea and show that it’s possible
  - Seamless interface switching
  - Support for receiving direction
  - Traffic engineering
- Performance!
  - Hardware-Acceleration? (FPGA / GPGPU)
Conclusion

- Architecture for decoupling network protocols from applications (Protocol Virtualization)
- Brings the idea of SDN to end nodes
- Enables adaptive networks for the complete path
- Useable in existing applications

We want to achieve in the near future:
- Complete the presented architecture idea
- Implement it for end- and infrastructure-nodes
- Enable good performance
The End

Thank You!

...and so, in conclusion, the proposed method...

THANK GOODNESS, ALMOST OVER... HOPEFULLY I DIDN'T BORE THEM TO TEARS.

...thank you, you've been a great audience...

OK, THE OBLIGATORY CALL FOR QUESTIONS AND I AM DONE...

any ques...

...tions?

OR NOT

www.phdcomics.com
Nunes et al.
Software-Defined-Networking-Enabled Capacity Sharing in User-Centric Networks
IEEE Communications Magazine 09/2014
Multistack

Honda et al
Rekindling Network Protocol Innovation with User-Level Stacks
SIGCOMM 04/2014