PolKA: Polynomial Key-based Architecture for Source Routing

Cristina Dominicini, Rafael Guimarães, Diego Mafioletti, Magnos Martinello, Moises R. N. Ribeiro, Rodolfo Villaça, Frédéric Loui, Jordi Ortiz, Frank Slyne, Marco Ruffini, and Eoin Kenny

1Federal Institute of Espírito Santo, 2Federal University of Espírito Santo, 3RENATER, 4University of Murcia, 5Trinity College Dublin, and 6HEANET

Contact: cristina.dominicini@ifes.edu.br
Motivation

- **SDN and Programmable Network Devices:**
  - Innovation and custom protocols.

- **Challenge:** How to select paths and load-balance between them to adapt to variable workloads?
  - **Common solution:** encode multiple paths in core nodes as forwarding table entries, and allow the edge to select among them.

- **Problems:**
  - Large number of states → Management burden
  - Restricted capacity of switch tables → Traffic engineering cannot exploit all paths
  - Latency for path setup
Source Routing (SR)

- A source specifies all forwarding nodes in the path.
- A route label is added to the packet header.
- **Traditional way: List-based SR (LSR)**
  - The path is defined as a stack of output ports.
- **Limitations:**
  - State in the packet:
    - Each node performs a pop on the stack.
    - Rewrite operation.
  - No implicit way of representing multiple paths.
Problem: Is it possible to define a fully stateless SR approach?
- No packet rewrite, No tables
- … and offer support for complex use cases...

- Source Routing based on a arithmetic operation
- Residue Number System (RNS) and Chinese Remainder Theorem (CRT)
- Emulated proof-of-concept in Mininet

**ONDM 2021: “Deploying PolKA Source Routing in P4 Switches”**
- Deployment in the GEANT P4 Lab testbed with Tofino switches
- PoC of PolKA in real-world environment
PolKA: Architecture

Service Layer
- Network Functions
- Network Applications
- Network Tools

Control and Orchestration Layer
- Topology Knowledge
- Path Computation
- Polynomial Generation
- Manipulation of tables

Physical Infrastructure Layer
- P4-based Data Plane

Motivation  Problem  Proposal  Prototype  Evaluation  Conclusions
The forwarding uses a mod operation (remainder of division):

\[
\text{portID} = \langle \text{routeID} \rangle_{\text{nodeID}}
\]

- P4 language does not support the mod operation.
- **Solution:** reuse CRC hardware (Cyclic Redundancy Check)
  - The Tofino Native Architecture (TNA) supports custom CRC polynomials.
How does PolKA work?

- In a network configuration phase, the **Controller** assigns irreducible polynomials to core switches (**nodeIDs**).
- Port labels are represented as binary polynomials (**portIDs**).
How does PolKA work?

- The **Controller** chooses a **path** for a specific flow (proactively or reactively):
  - A set of switches: \{0011, 0111, 1011\}
  - and their output ports: \{1, 10, 110\}
How does PolKA work?

- The **Controller** calculates the *route IDs* using the polynomial **Chinese Remainder Theorem**.

\[ s_1(t) = t + 1 = 11 \]
\[ s_2(t) = t^2 + t + 1 = 111 \]
\[ s_3(t) = t^3 + t + 1 = 1011 \]

**node ID polynomials**

\[ o_1(t) = 1 \]
\[ o_2(t) = t = 10 \]
\[ o_3(t) = t^2 + t = 110 \]

**port ID polynomials**

Calculate route ID with CRT

\[ t^4 \equiv 1 \mod (t + 1) \]
\[ t^4 \equiv t \mod (t^2 + t + 1) \]
\[ t^4 \equiv (t^2 + t) \mod (t^3 + t + 1) \]
\[ t^4 = 10000 \]
How does PolKA work?

- The **Controller** installs **flow entries** at the edges to add/remove **routeIDs**.

\[ R = 10000 \]
How does PolKA work?

- When packets arrive, an action at ingress embeds `routeID` into the packets.
How does PolKA work?

- Forwarding using mod operation: $<10000>_{0011} = 1 \rightarrow$ output port
- No packet rewrite! No tables!
How does PolKA work?

- Forwarding using mod operation: \(<10000>_{0111} = 10 \rightarrow \text{output port}\)
- No packet rewrite! No tables!
How does PolKA work?

- Forwarding using mod operation: $<10000>_{1011} = 110 \rightarrow$ output port
- No packet rewrite! No tables!
How does PolKA work?

- Finally, an action at edge egress node removes *routeID*. 
How does PolKA work?

- Packet is delivered to the application in a transparent manner.
GÉANT P4 Lab Testbed

- RARE project: [https://wiki.geant.org/display/RARE](https://wiki.geant.org/display/RARE)
- Testbed with Intel/Tofino Barefoot P4 Switches
Throughput & Forwarding Latency:

- PolKA matches the performance of traditional L2 table-based forwarding and LSR approaches.
Agile Path Reconfiguration:
- SDN Controller changes a single flow entry at H1: path is reconfigured from shortest to longest path.
Future Works

- We are integrating PolKA in RARE repository for experimenters.
  - Extension of control planes functionalities.
- We are preparing deployment guidelines for production use cases.
- This proposal was one of the recipients of the 2021 Google Research Scholar Award.
- We are also exploring PolKA properties for innovative applications.
  - Security and Fast Failure Reaction exploring RNS properties.
  - Multipath Routing.
  - ...
Future Works: Multipath Routing

- Extension: the *portid* coefficients represent the transmitting state of the ports instead of port labels.
Future Works

- **Polynomial representation**
  - Polynomials of higher orders for Multi-layer Networks and Slicing

- **Use of multiple keys**
  - Protection paths
  - QoS

- **Source Routing**
  - Service Function Chaining
  - Save TCAM for hybrid operation with table-based approaches
  - Agile Path Reconfiguration
Thank you!

Cristina Klippel Dominicini

cristina.dominicini@ifes.edu.br