# OPT:
## LIGHTWEIGHT SOURCE AUTHENTICATION & PATH VALIDATION

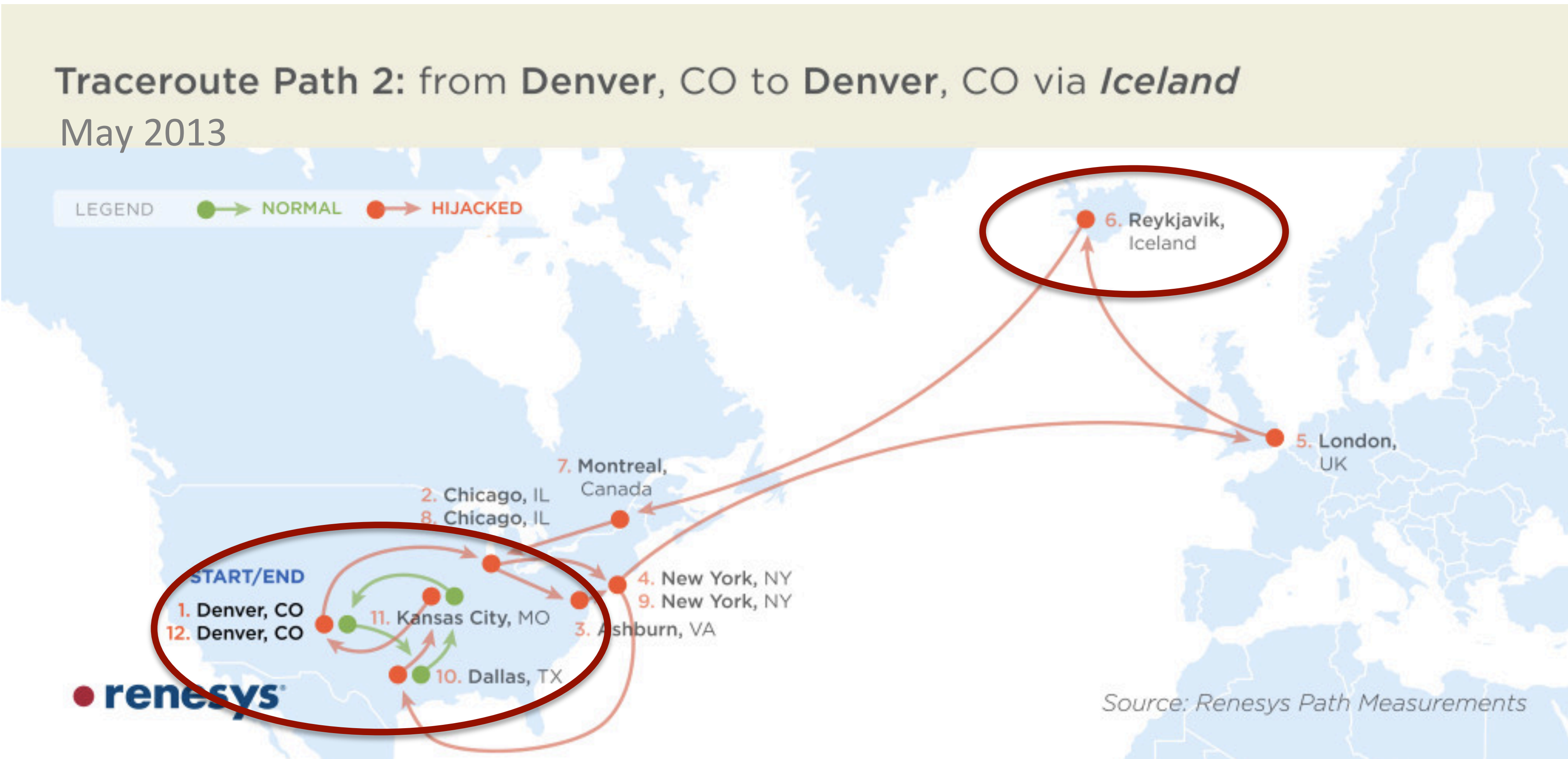
**Tiffany Hyun-Jin Kim**,[1] Cristina Basescu,[2] Limin Jia,[1] Soo Bum Lee,[3] Yih-Chun Hu,[4] and Adrian Perrig[2]

[1]Carnegie Mellon University, [2]ETH Zurich, [3]Qualcomm, [4]Univ. of Illinois at Urbana-Champaign

ACM SIGCOMM, August 20, 2014

# REAL INTERNET PATH MISDIRECTION

- **Limited control of paths → hijacked & redirected**



# POTENTIAL ATTACK SURFACES

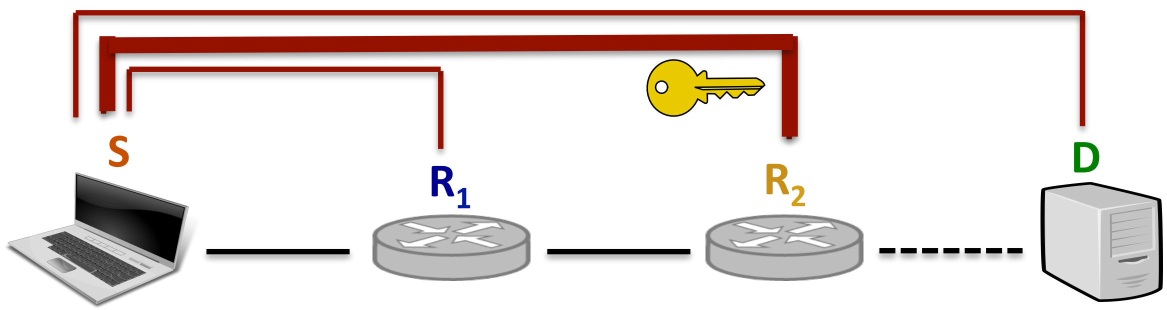
- **Traffic diversion**
  - Attacker eavesdrops any parts of packets (e.g., metadata) with potentially sensitive info

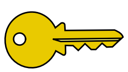
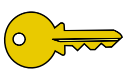
- **Fictitious premium path usage**
  - ISPs use inferior path but charge for premium path
- **Packet injection with spoofed source address**
  - Routers inject extra packets to incriminate source

# CURRENT INTERNET DOESN'T SUPPORT

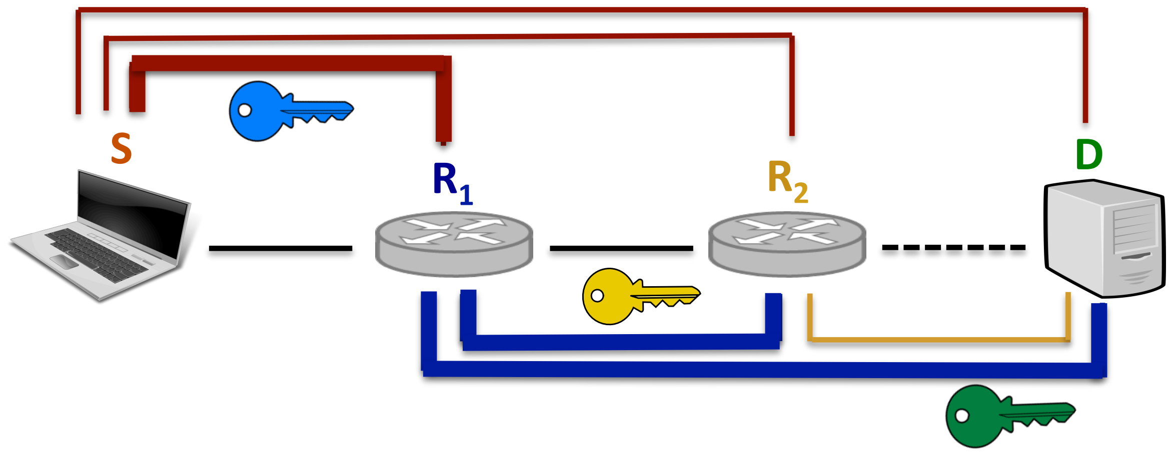
- **Path validation**
  - Client selects an intended path
    - Could be at AS-level or router-level
  - Endhosts check if packet followed intended path in the correct order
- **Source authentication**
  - Routers check the sender of received packet
  - To mitigate address spoofing attacks

# HOW SOURCE CAN BE AUTHENTICATED




- **Use shared secret key with S**
  - $R_2$ shares secret key  with S
  - S creates an authentication field (e.g., MAC) using 
  - Correct MAC can *only* be generated by S

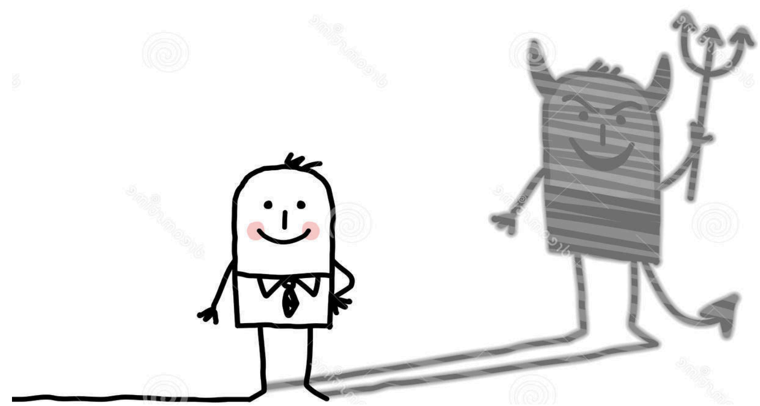
# HOW PATH CAN BE VALIDATED




- **Set up shared secret keys**
  - Using , $R_1$ checks path has been followed so far
  - Using , $R_1$ creates a proof for $R_2$ that it has seen the packet
  - Using , $R_1$ creates a proof for D as well

# COWARD ATTACKS [1]

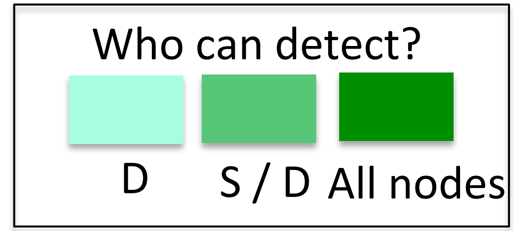
- **Typical source authentication & path validation**
  - Require key setup in advance
- **Attacker's goal is not to get caught**
  - If malicious routers know they are being monitored → *attackers start obeying protocol*

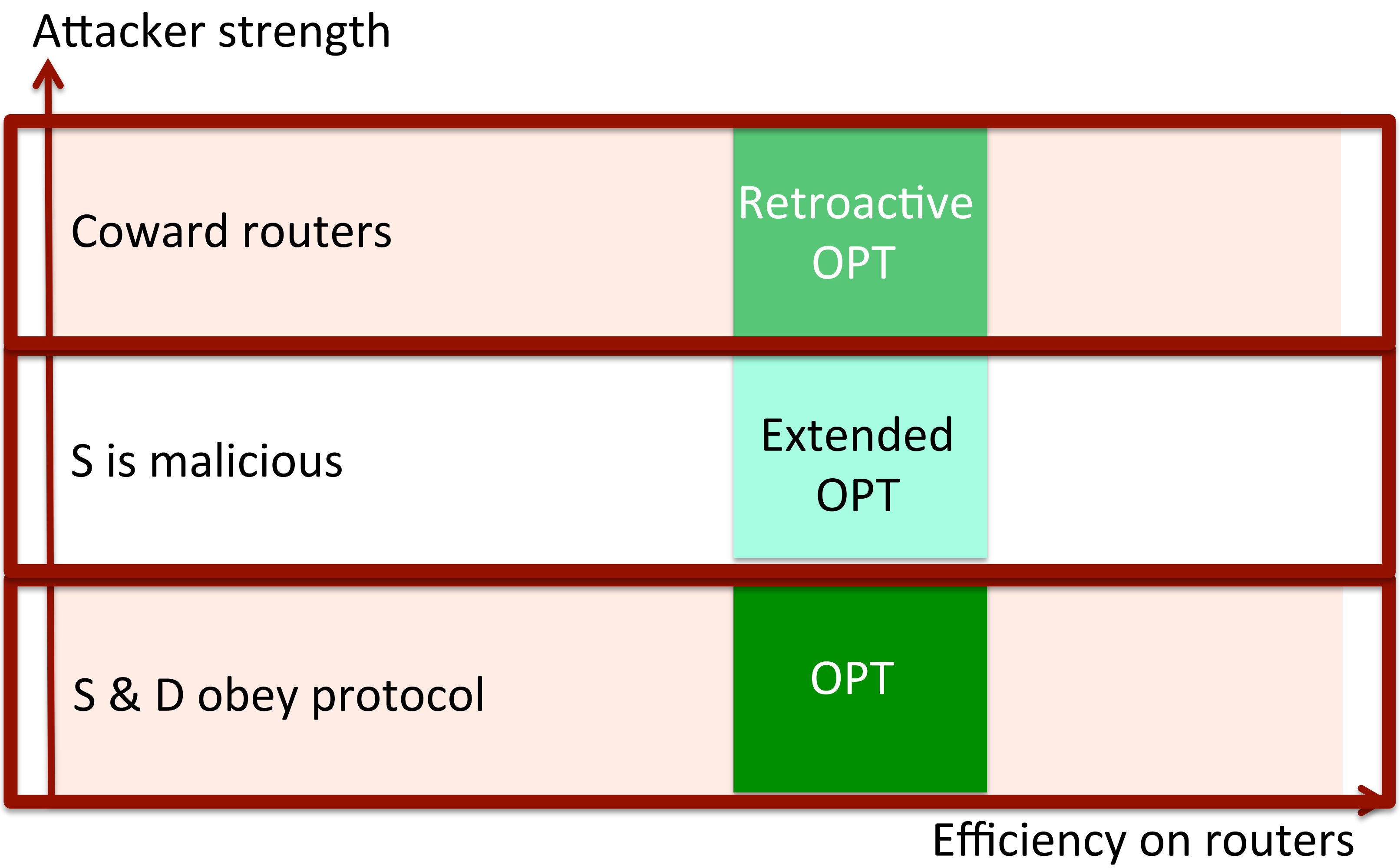


[1] J. Liu et al. Coward Attacks in Vehicular Networks. *Mobile Computing and Communications Review,* 2010.

**Can we design a mechanism for source authentication and path validation that is *practical* for deployment?**

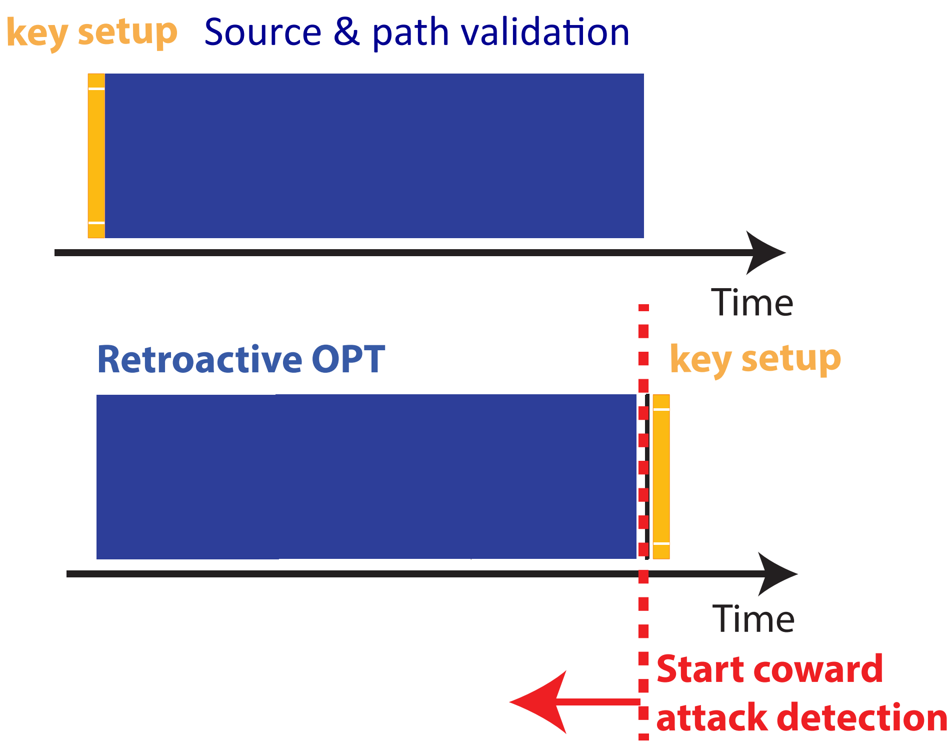
# OUR DESIGN DECISION



# RETROACTIVE-OPT

- ***No key setup*** **before packet forwarding**
  - Only with suspected misbehavior, S and D set up keys for *previous* packets

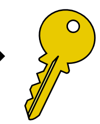


# RETROACTIVE-OPT

- ***No key setup*** **before packet forwarding**
  - Only with suspected misbehavior, S and D set up keys for *previous* packets

- **Routers commit some value during forwarding**
  - Reveal keys used for the commitment later
  - Wrong key or incorrect commitment → misbehavior detected

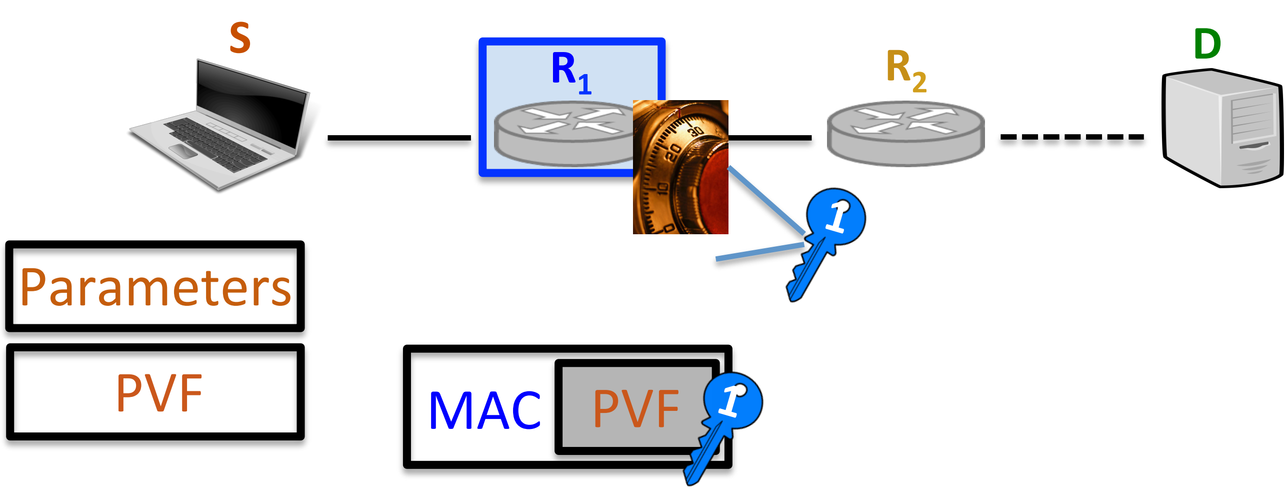
# EFFICIENCY ON ROUTERS

- **Dynamically re-creatable keys on the fly**
  - S selects Parameters that other routers use for key setup
  - Parameters in packet header + *local secret* in memory → 

- ***Constant* crypto computation during forwarding**
  - Independent of *path length*
  - O(1) Message Authentication Code (MAC) operation per packet

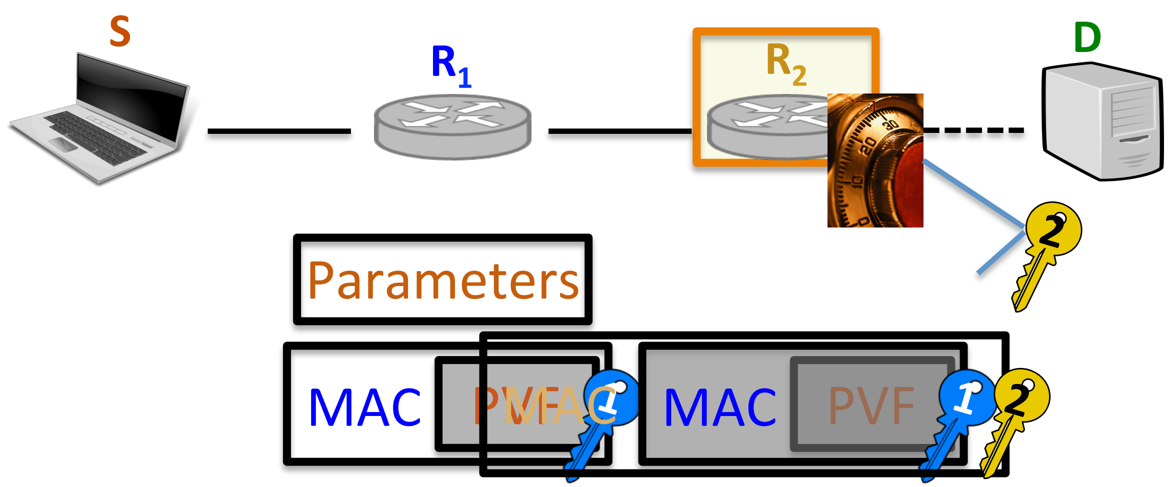
# RETROACTIVE-OPT PROCESS

- **Each OPT downstream node derives a key**
  - Parameters in packet header + local secret in memory
- **Commits PVF with 1 MAC operation**

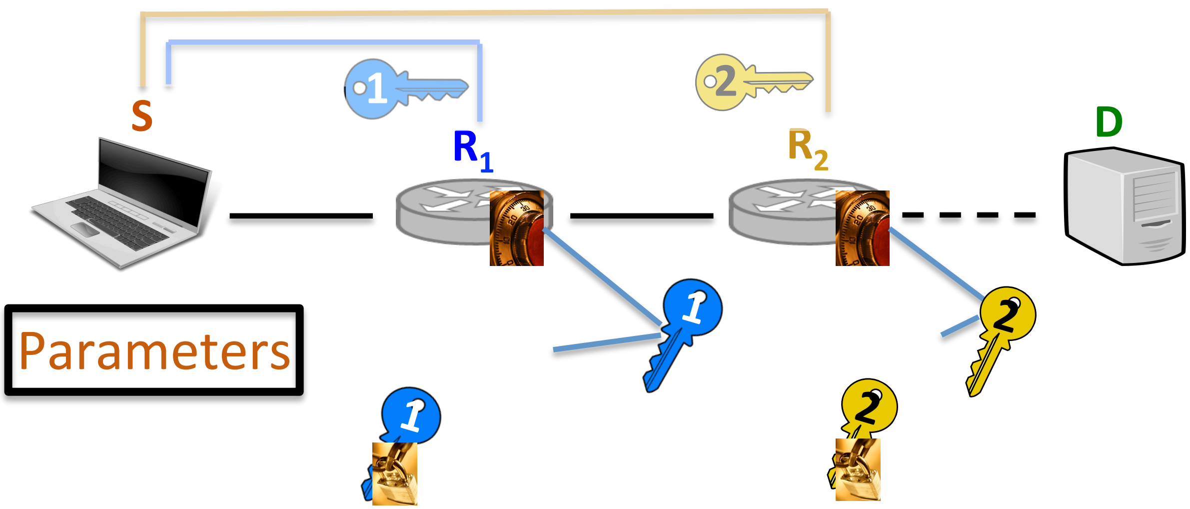


# RETROACTIVE-OPT PROCESS

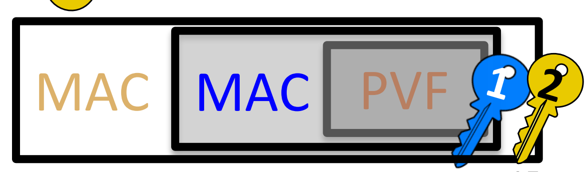
- **Each OPT downstream node derives a key**
  - Parameters in packet header + local secret in memory
- **Commits PVF with 1 MAC operation**



# DYNAMICALLY RECREATABLE KEY




- **Later when S or D wants to validate path for *previous* packets**
  - S forwards Parameters to routers
  - Parameters + single local secret → Router *recomputes* key
  - Forward encrypted & signed keys
  - To detect misbehavior, D recomputes



# LIGHTWEIGHT ON ROUTERS

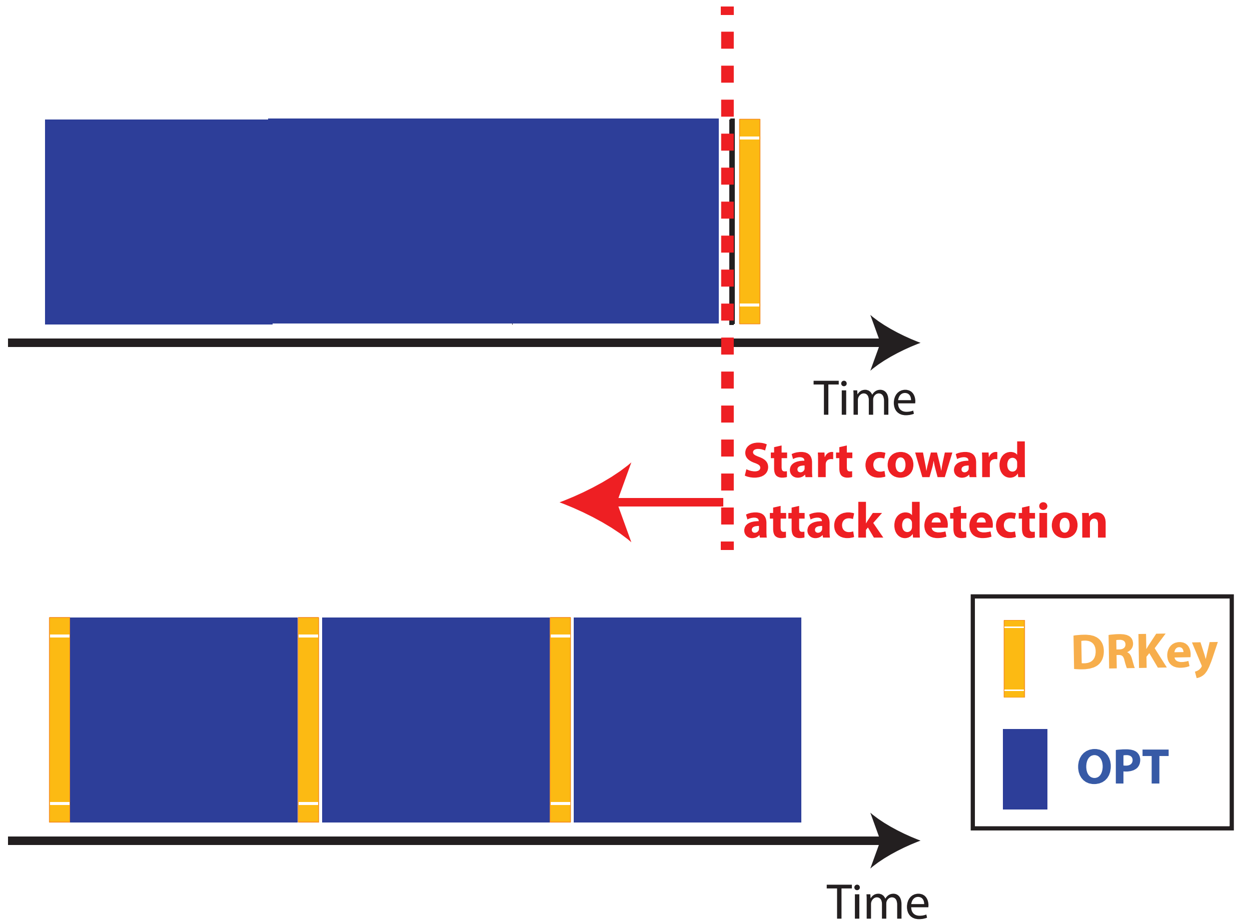
- **Pushes complexity to end hosts**

| | ROUTER | SOURCE / DESTINATION |
|---|---|---|
| **MAC operations** | **O(1)** | O(n) |
| **Storage** | local secret | Parameters MAC MAC PVF |

- **Retroactive-OPT header size *independent* of path length & small**
  - Higher goodput

# OPT VARIATIONS IN PAPER

1. Retroactive-OPT

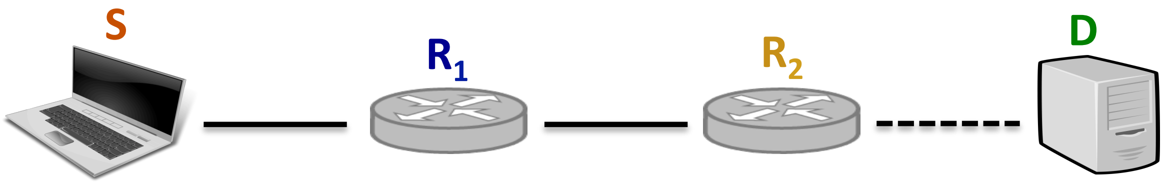



2. OPT
3. Extended-OPT

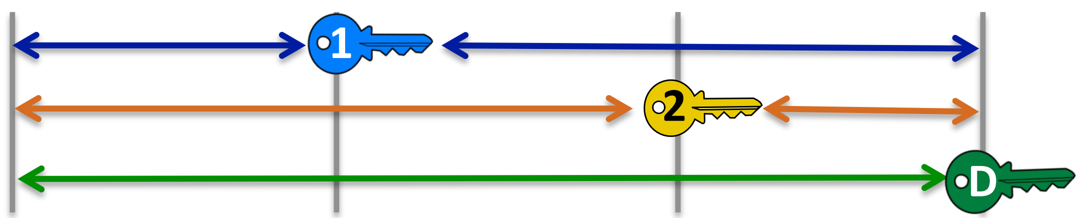
- Keys are set up before protocol starts

# OPT & EXTENDED-OPT OVERVIEW

- **S selects a path to D**



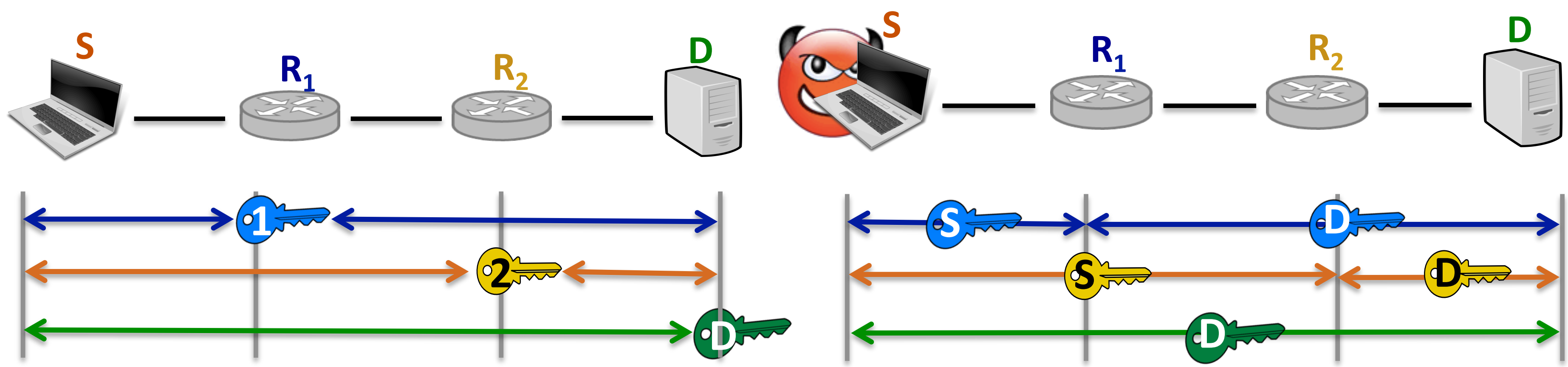

- **Nodes establish shared secret key(s) with S & D**




- **S prepares special fields for each node in the packet header**
  - Helps each router *derive shared key & authenticate source*

- **Each node updates a verification field in the packet header**
  - Helps downstream nodes *validate path*

# 2 OTHER VARIATIONS OF OPT




- **OPT**
  - S & D obey the protocol
  - R shares 1 key with S & D
  - All nodes detect
- **Extended-OPT**
  - S may be malicious
  - R shares 2 keys
  - Destination detects

# CAN OPT DEFEND AGAINST ATTACKS?

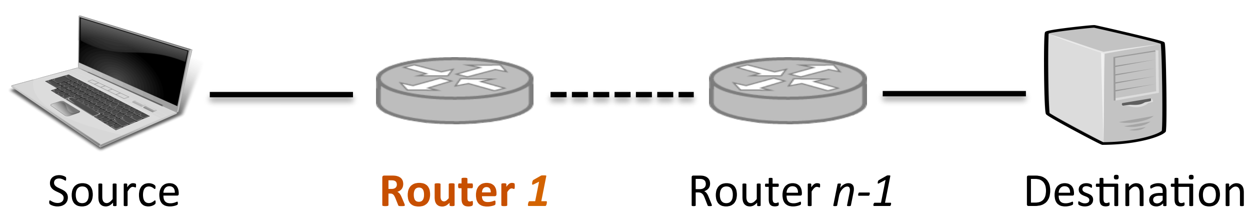
- **Proof-based (mechanized) formal verification [2]**

| ATTACKER | **DEFENSE** |
|---|---|
| Alters packets | Cannot compute *valid PVF* without secret keys |
| Deviates path | Cannot compute *valid PVF* |
| Coward attacks | *Retroactive* version mitigates |
| State-exhaustion DoS attacks | Memory-lookup of *a single value* & *O(1) MAC* operation |
| Collude & redirect packets | *Honest router or destination* drops |

[2] F. Zhang, et al. Mechanized Network Origin and Path Authenticity Proofs. *To appear in CCS 2014.*

# OPT IMPLEMENTATION

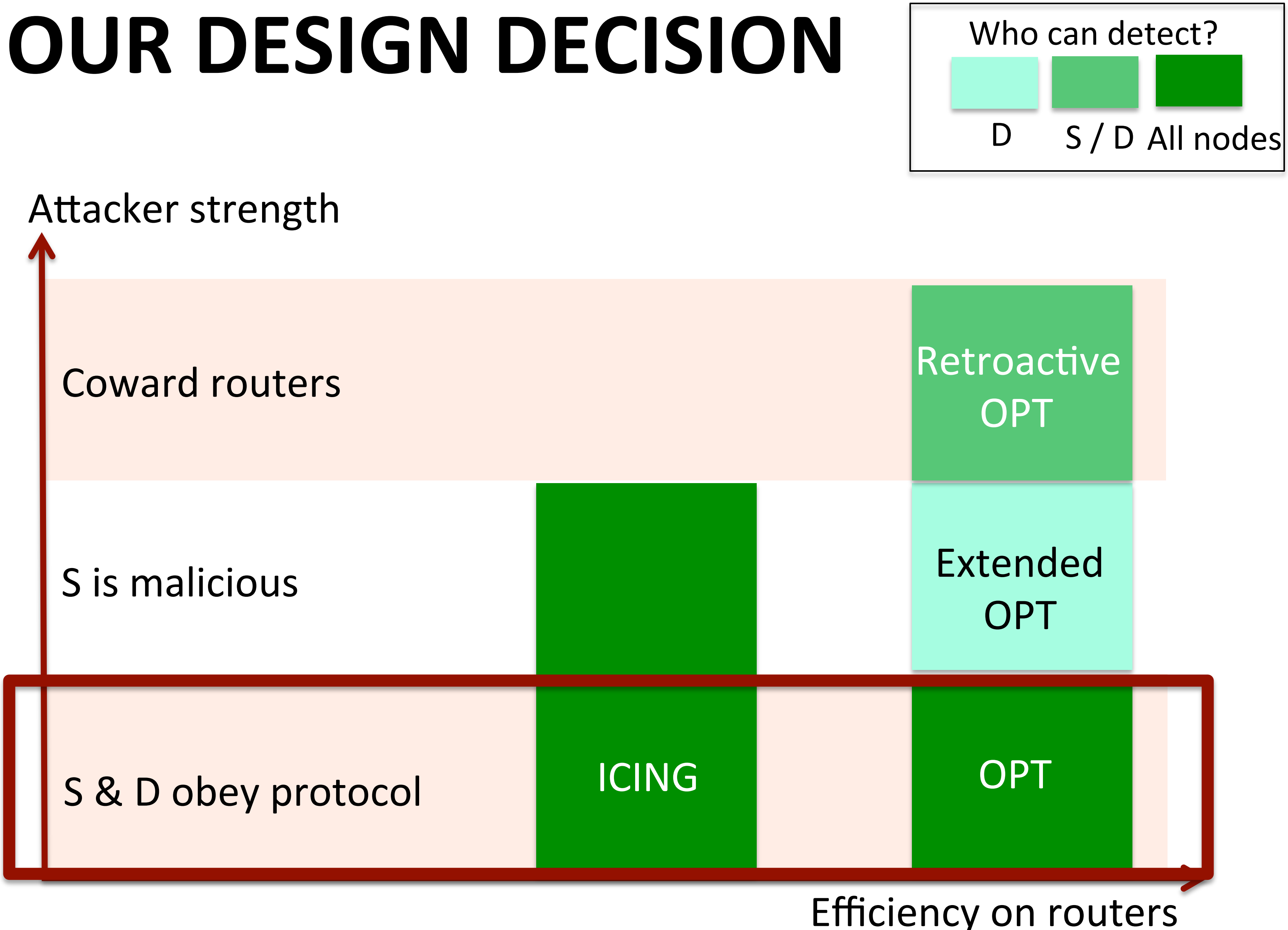
- **Router performance evaluation goals**
    1. Per-packet processing overhead
    2. Scalability w.r.t. path length
- **Compare generic OPT with ICING [3]**
    - Pairwise key-based source authentication & path validation for all nodes




| | |
|---|---|
| Storage overhead | *n-1* pairwise keys (16B each) |
| Computation overhead | *n-1* MAC computations |

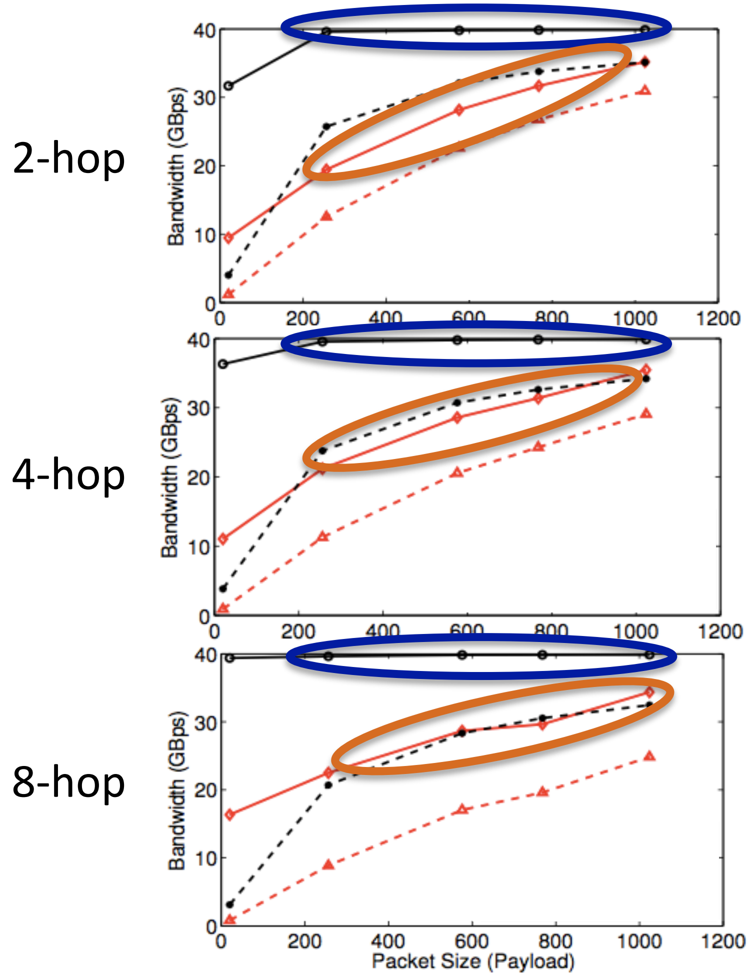
[3] J. Naous et al. Verifying and Enforcing Network Paths with ICING. CoNEXT 2011

# OUR DESIGN DECISION



# OPT THROUGHPUT & GOODPUT

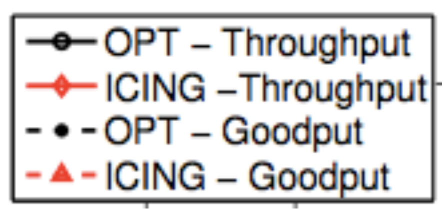
- **Traffic generated for 10 sec at 40 Gbps**




**OPT throughput**

vs.

**ICING throughput**



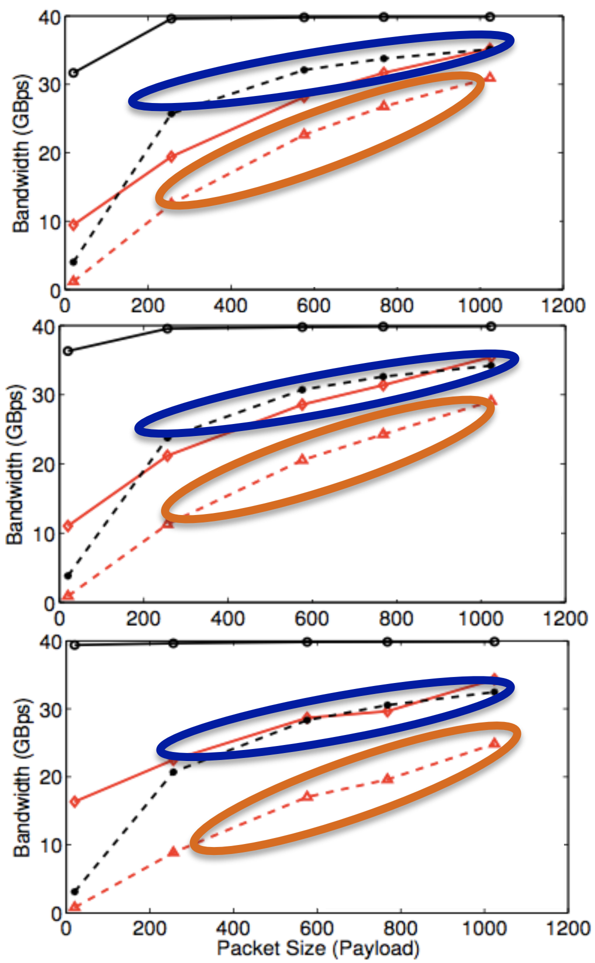
# OPT THROUGHPUT & GOODPUT

- **Traffic generated for 10 sec at 40 Gbps**

2-hop

4-hop

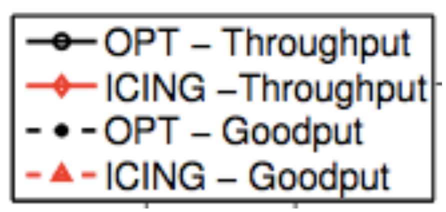
8-hop




**OPT goodput**

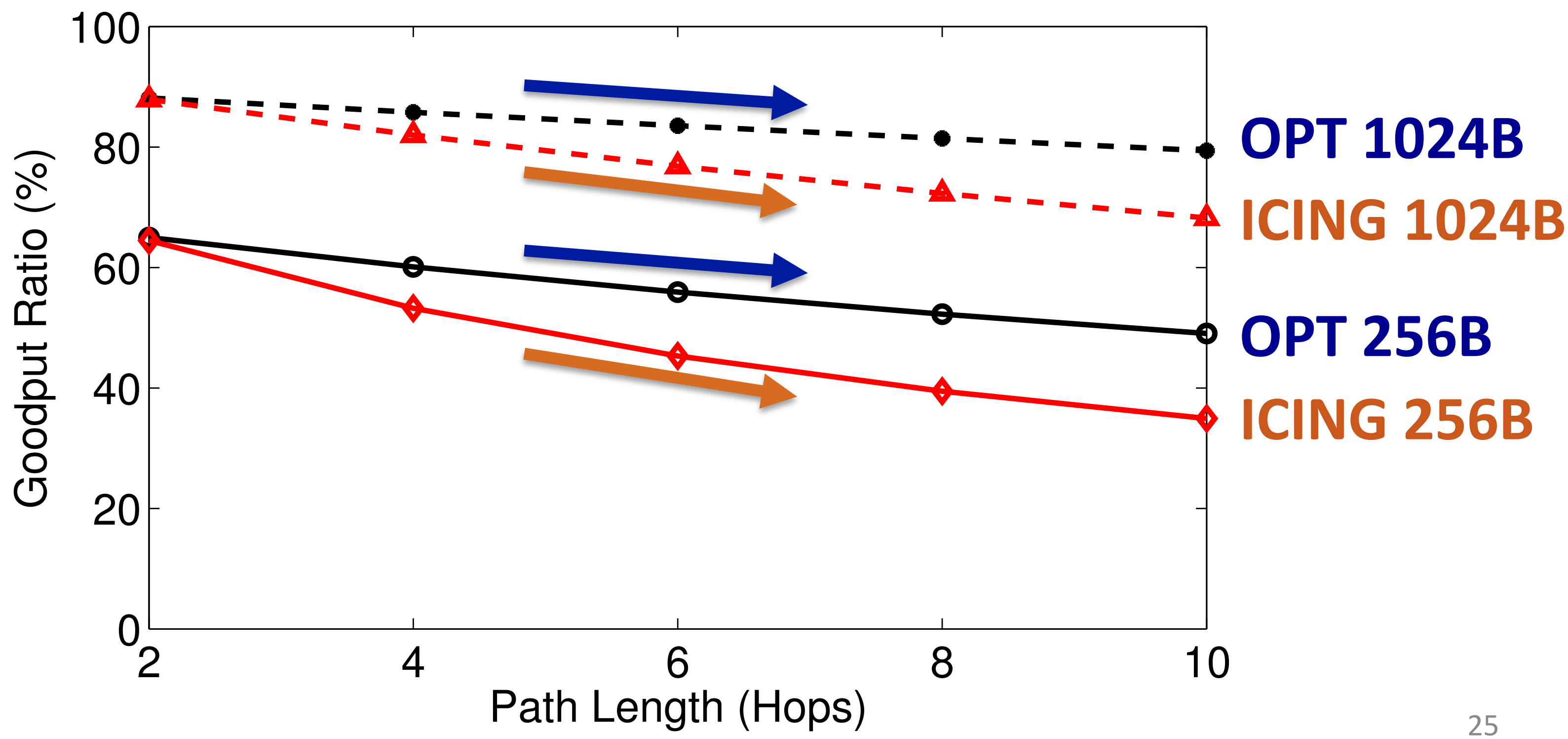
vs.

**ICING goodput**



# OPT PATH LENGTH SCALABILITY

- **Ratio between goodput & throughput**
  - Small (256B) and large (1024B) packets with varying path lengths



# CONCLUSIONS

- **OPT: efficient protocol for source and path validation**
  - Without burdening *routers*
- **OPT achieves performance improvements**
  - Minimal storage & computational overhead on routers
    - *Regardless of path length*
- **Retroactive-OPT to defend against *coward attacks***

## Thank you

***hyunjin@cmu.edu***

Special thanks to: George Danezis, Yue-Hsun Lin, Ratul Mahajan, Raphael Reischuk, XIA team, and anonymous reviewers ☺