LISP+Wireguard

Alejandro Barcia, Albert López, Jordi Paillissé, Albert Cabellos

LISP WG Interim Meeting

May 2020

Intro

- Motivation
    - Rethink LISP security architecture
    - Focus on popular use cases only
        - No need to provide security for all the use-cases
    - Main inspiration: Wireguard
- In this talk
    - What is Wireguard?
    - LISP+Wireguard
    - Implementation & Performance analysis
    - Discussion

What is Wireguard?

What is Wireguard?

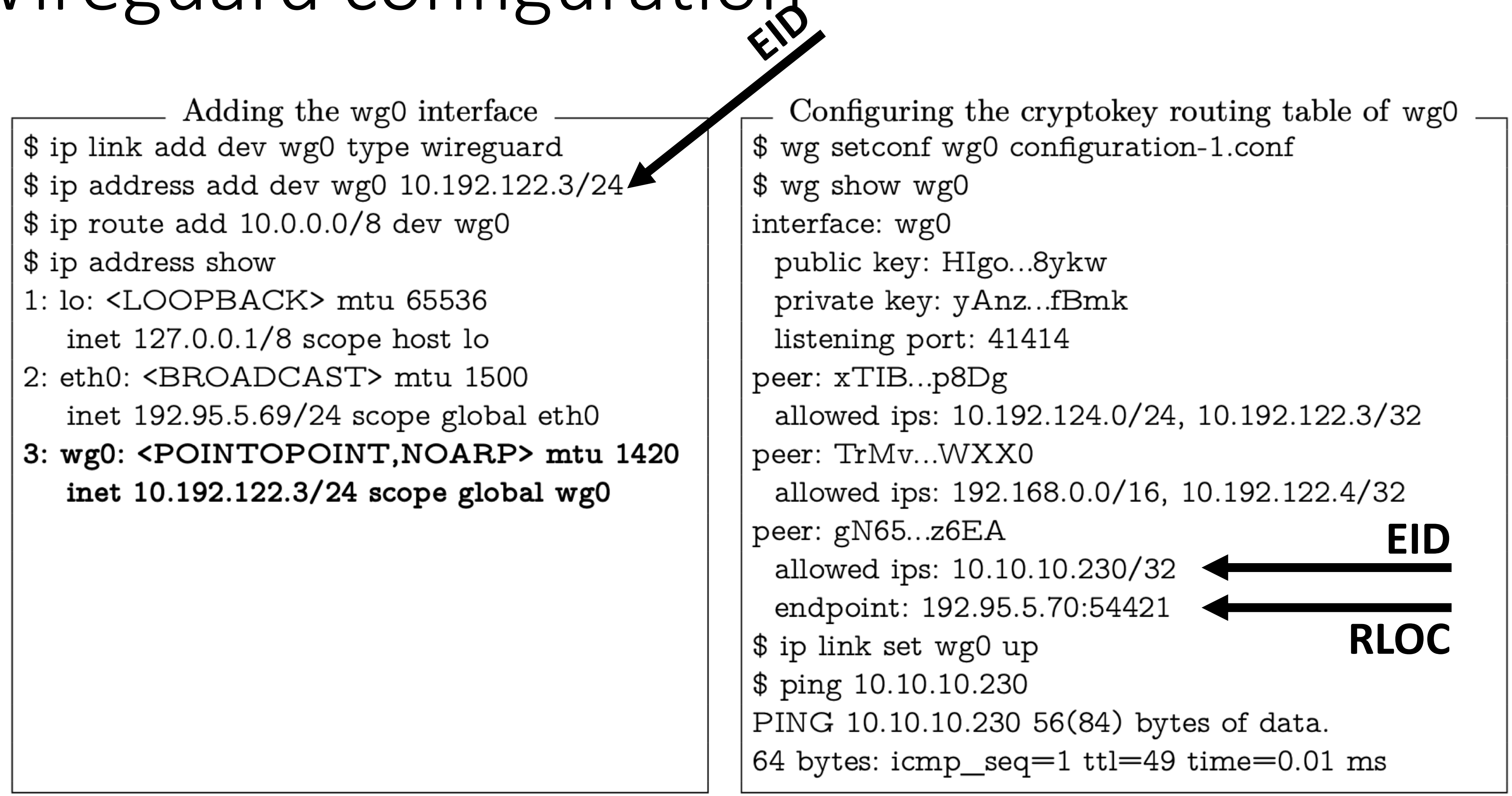
WIREGUARD®

FAST, MODERN, SECURE VPN TUNNEL

- Wireguard is a secure network tunnel (VPN)
- Merged in the Linux Kernel (≥ 5.6)
- Wireguard Design Principles
  - Traditional solution is IPSec+IKEv2
  - Large choices of cyphersuites and key exchange mechanisms
  - Separated exchange layer form encrypted transport
  - This results in complex code, hard to perform security audits and prone to misconfiguration
- Wireguard aims to tradeoff flexibility for simplicity

Jason A. Donenfeld “WireGuard: Next Generation Kernel Network Tunnel “https://www.wireguard.com/papers/wireguard.pdf

Wireguard configuration



Adding the wg0 interface

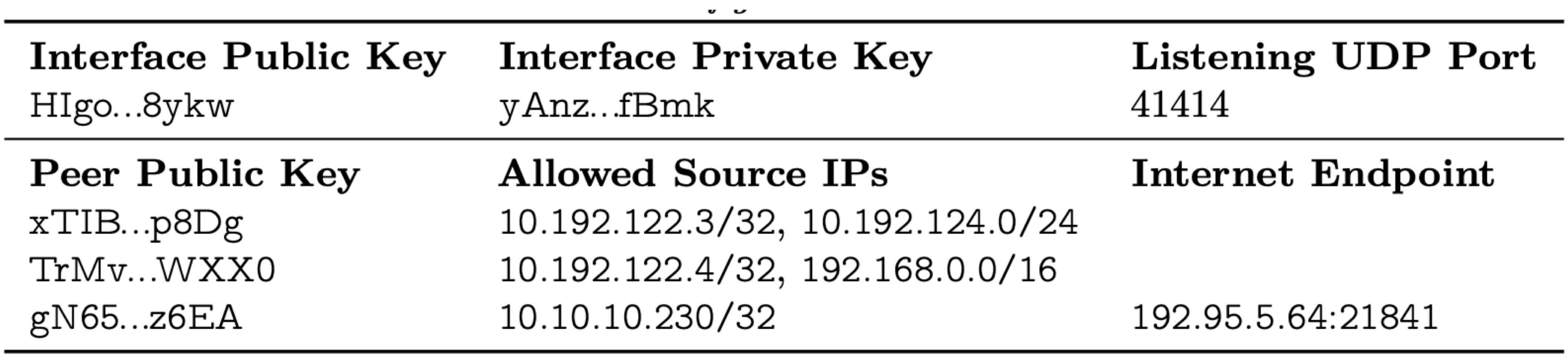
```
$ ip link add dev wg0 type wireguard
$ ip address add dev wg0 10.192.122.3/24
$ ip route add 10.0.0.0/8 dev wg0
$ ip address show
1: lo: <LOOPBACK> mtu 65536
    inet 127.0.0.1/8 scope host lo
2: eth0: <BROADCAST> mtu 1500
    inet 192.95.5.69/24 scope global eth0
3: wg0: <POINTOPOINT,NOARP> mtu 1420
    inet 10.192.122.3/24 scope global wg0
```

Configuring the cryptokey routing table of wg0

```
$ wg setconf wg0 configuration-1.conf
$ wg show wg0
interface: wg0
  public key: HIgo...8ykw
  private key: yAnz...fBmk
  listening port: 41414
peer: xTIB...p8Dg
  allowed ips: 10.192.124.0/24, 10.192.122.3/32
peer: TrMv...WXX0
  allowed ips: 192.168.0.0/16, 10.192.122.4/32
peer: gN65...z6EA
  allowed ips: 10.10.10.230/32
  endpoint: 192.95.5.70:54421
$ ip link set wg0 up
$ ping 10.10.10.230
PING 10.10.10.230 56(84) bytes of data.
64 bytes: icmp_seq=1 ttl=49 time=0.01 ms
```

Jason A. Donenfeld "WireGuard: Next Generation Kernel Network Tunnel "https://www.wireguard.com/papers/wireguard.pdf

Wireguard cryptokey routing



Interface Public Key	**Interface Private Key**	**Listening UDP Port**
HIgo...8ykw	yAnz...fBmk	41414
Peer Public Key	**Allowed Source IPs**	**Internet Endpoint**
xTIB...p8Dg	10.192.122.3/32, 10.192.124.0/24	
TrMv...WXX0	10.192.122.4/32, 192.168.0.0/16	
gN65...z6EA	10.10.10.230/32	192.95.5.64:21841

Jason A. Donenfeld “WireGuard: Next Generation Kernel Network Tunnel “https://www.wireguard.com/papers/wireguard.pdf

What is Wireguard?

WIREGUARD®

FAST, MODERN, SECURE VPN TUNNEL

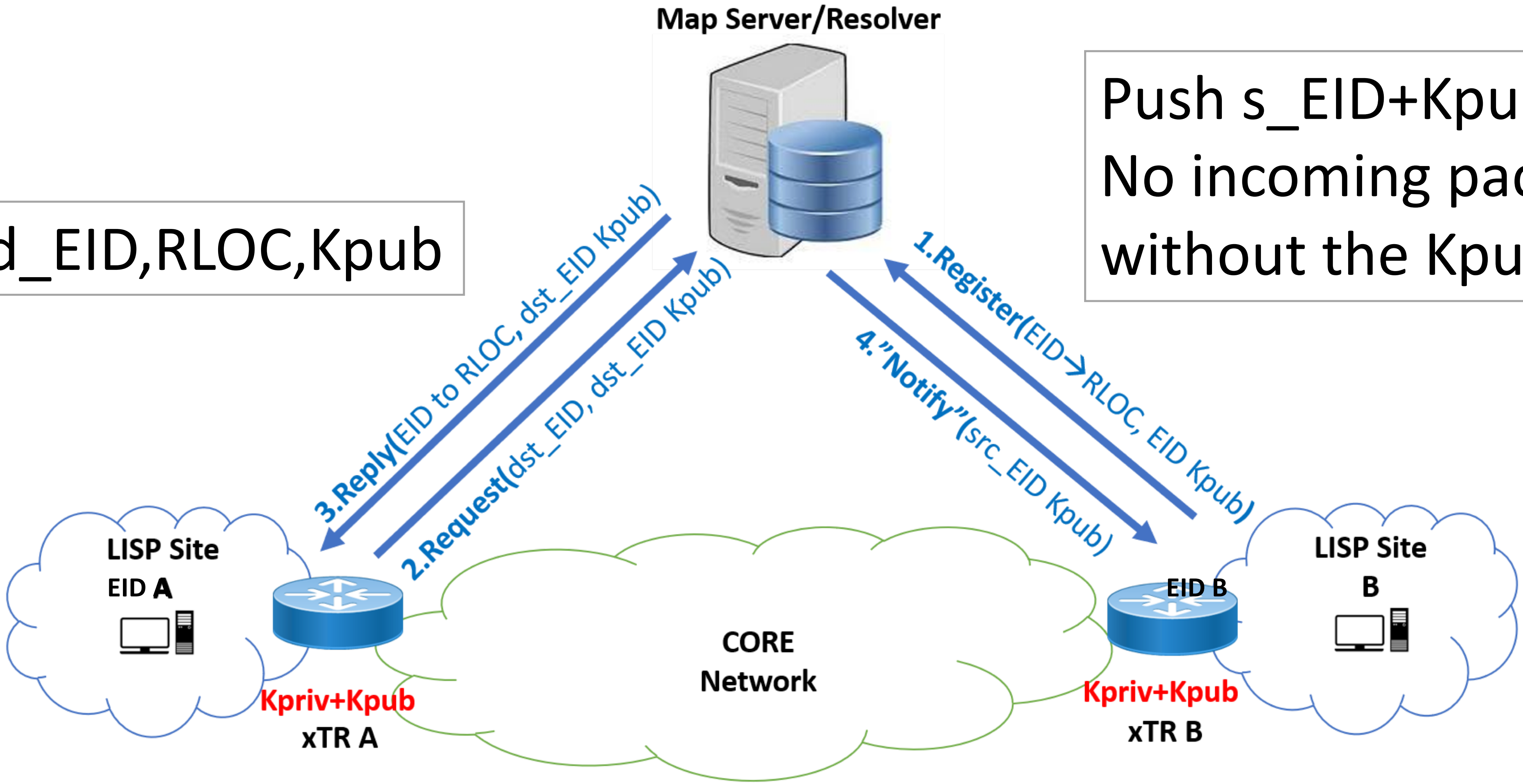
- Key distribution equivalent to OpenSSH → Out-of-band exchange of static public keys between peers
- Wireguard lacks cyper and protocol agility, only supports a set of cyphersuits.
  - No cypersuite negotation
  - All nodes need to be (software) updated to support new ones
- Session key exchanges, connections, disconnections, reconnections, discovery, and so forth happen behind the scenes transparently
- **Wireguard natively supports layer 3 mobility**
  - No need to notify peers (e.g, SMR) about new location or rendevouz server (e.g., Home Agent)

LISP+Wireguard

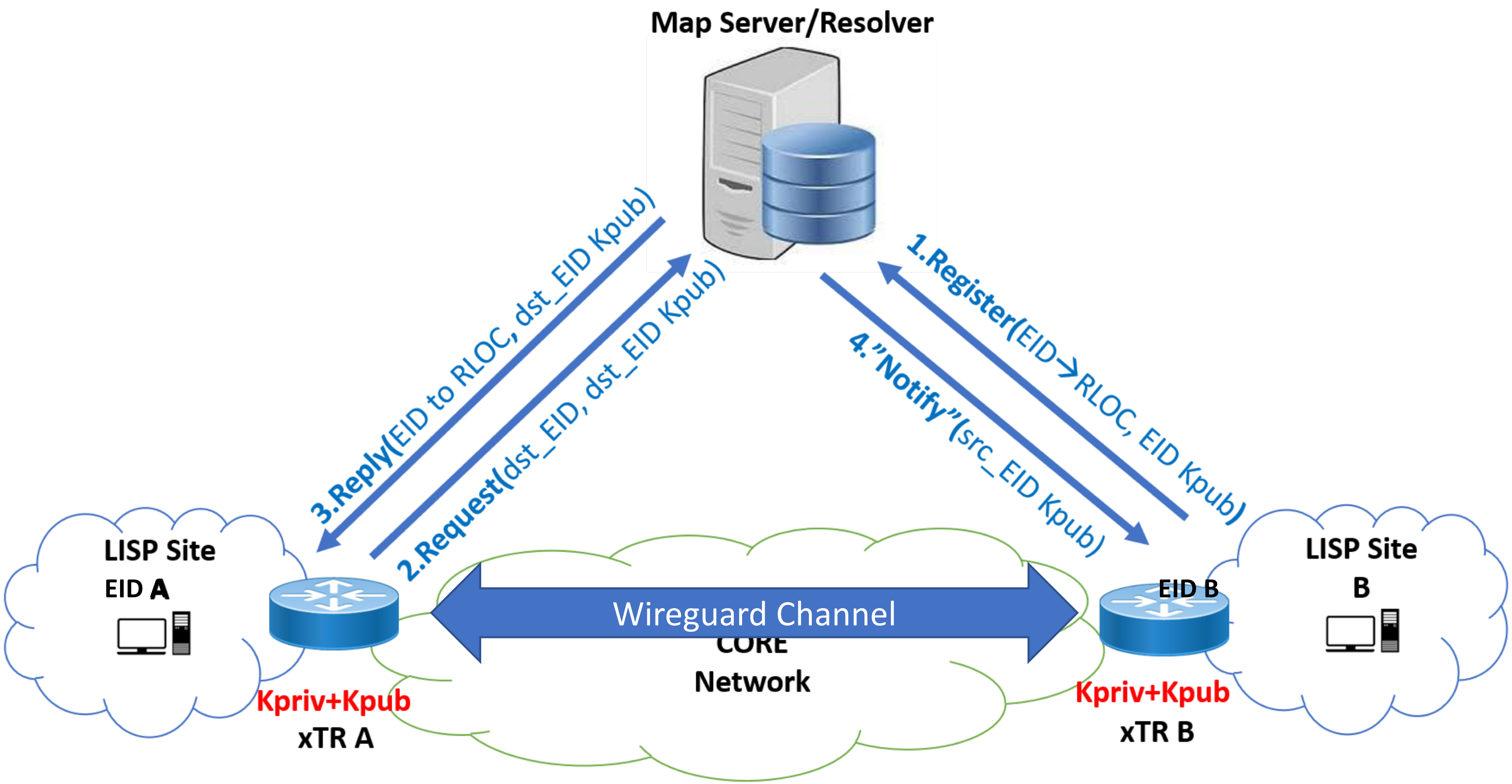
LISP+Wireguard

Pull d_EID,RLOC,Kpub

Push s_EID+Kpub
No incoming packets
without the Kpub

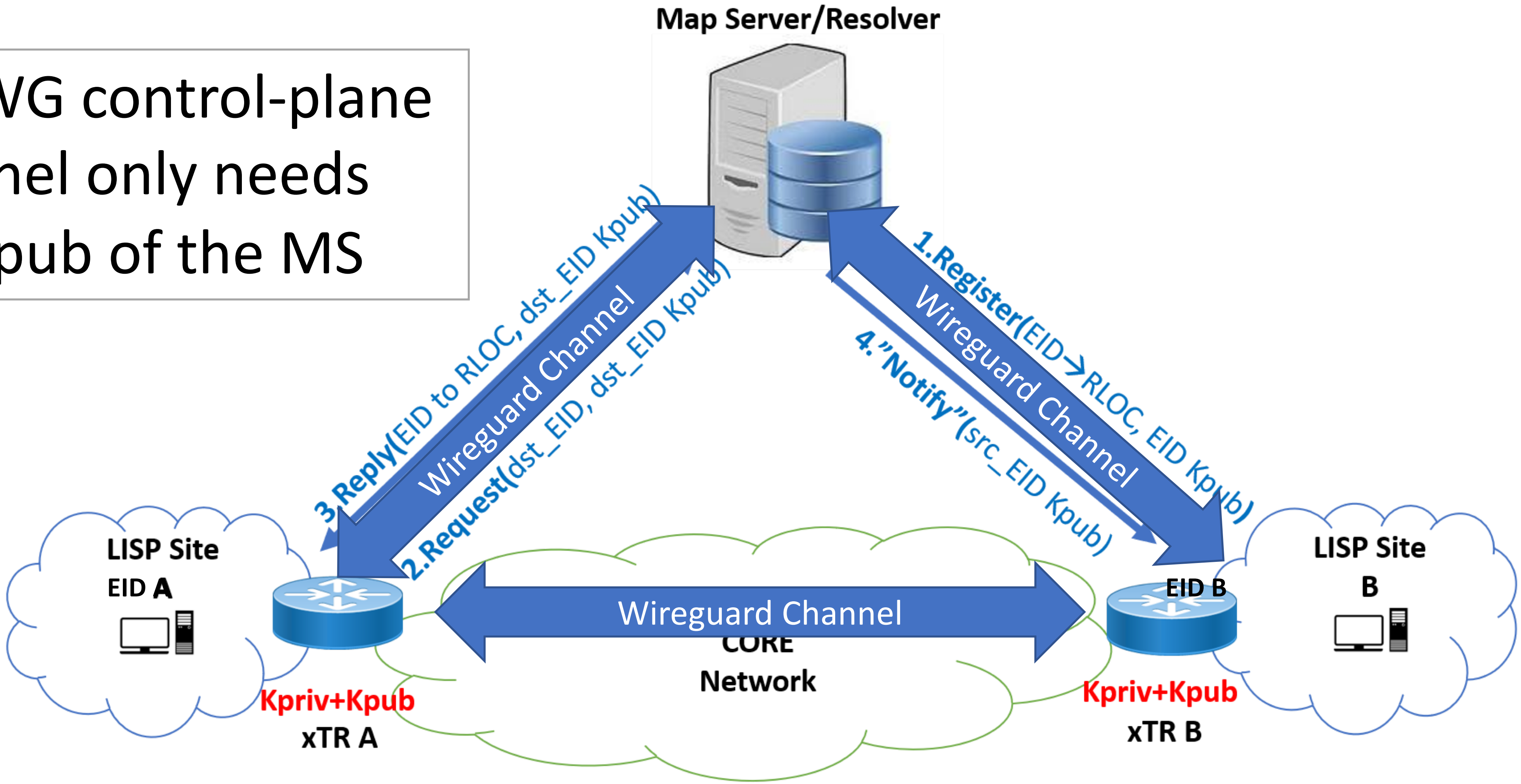


LISP+Wireguard



LISP+Wireguard

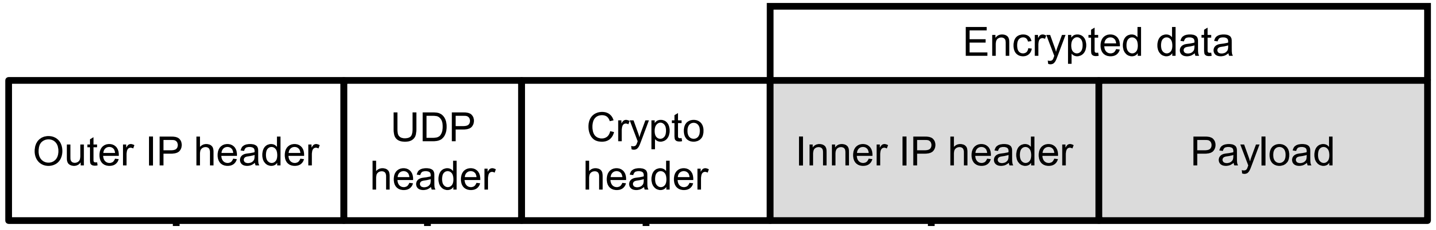
The WG control-plane Channel only needs the Kpub of the MS



Implementation & Performance Analysis

Prototype

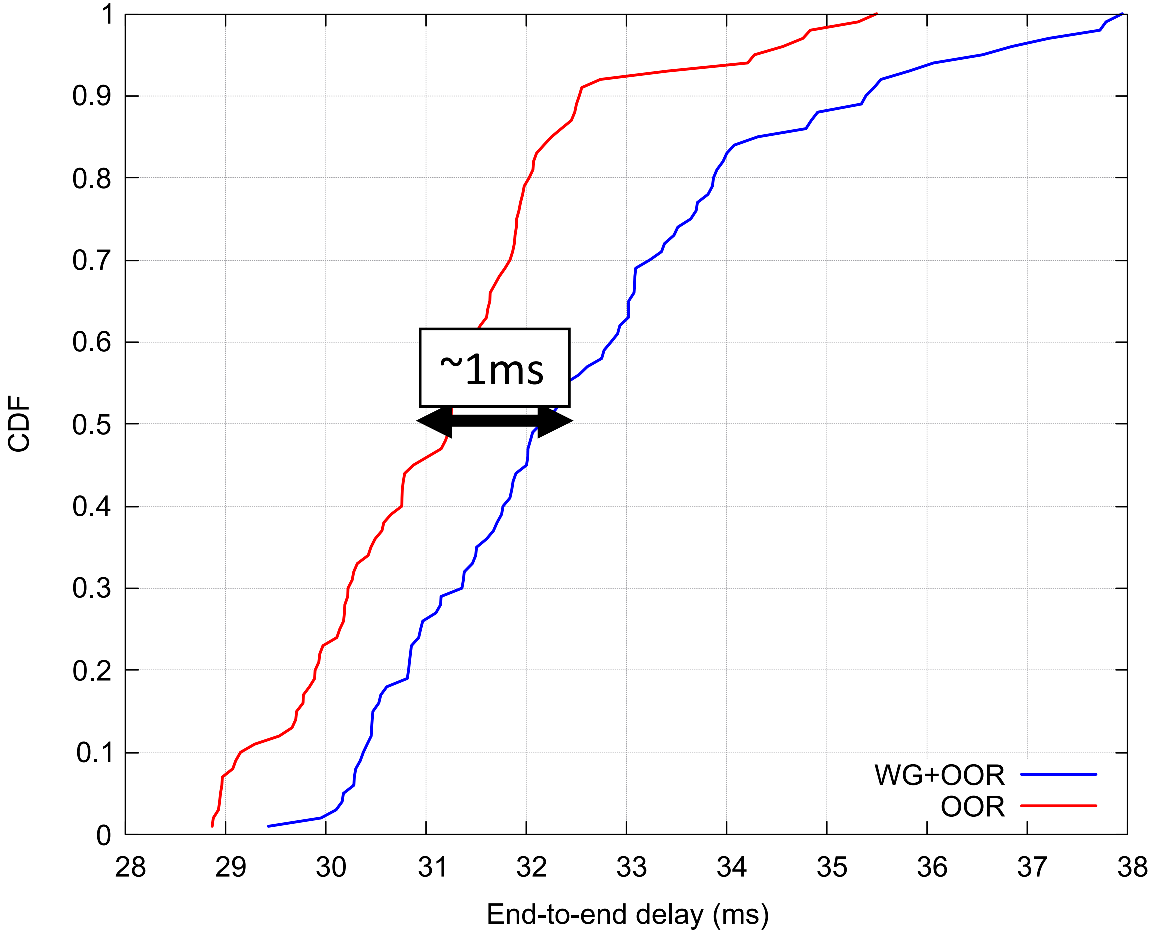
- Prototyped and opensourced using Open Overlay Router [1]
- We configure the wg0 interface using WG API
- Mappings are only needed for the first connection
  - Afterwards, WG takes care of new EID-to-RLOC mappings for that peer
- No modifications to Wireguard kernel module
  - No multihoming
  - No IID
  - No distributed MS



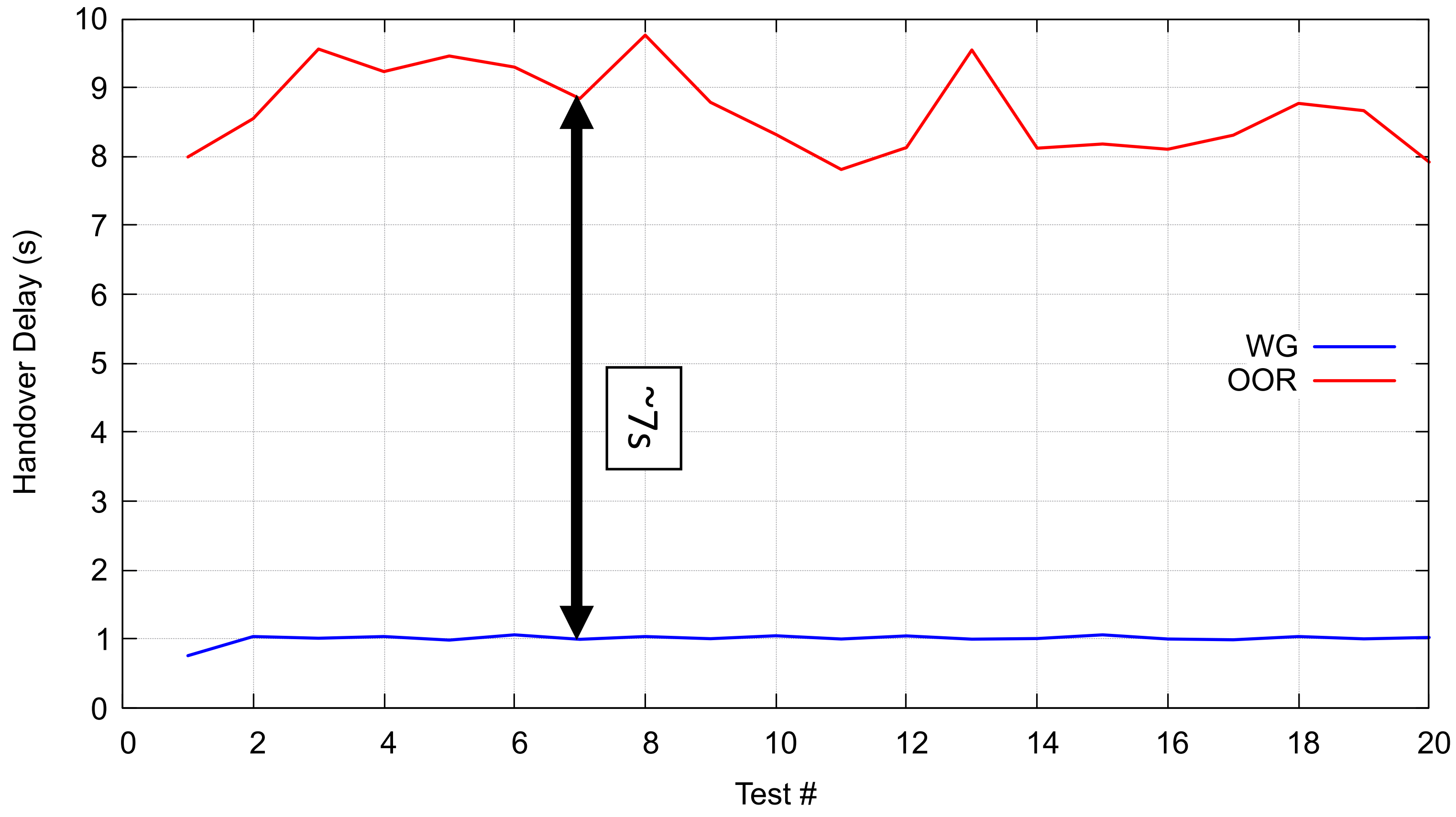
[1] https://openoverlayrouter.org/

End-to-End Latency

- Caches empty, latency of the first packet



Handover latency



- No SMR
- No RTR
- No control-plane
- No RLOC-probing
- Data-packets are authenticated with the Kpub

Discussion

Discussion

- This work represents two things:
  - A LISP security architecture assuming a single MS deployment
  - A control-plane for Wireguard
- How to support multi-homing?
- How to support IID?
- How to support distributed Mapping System?
- What can we learn from WG design principles?