PANRG · IETF 126 · VIENNA · JULY 2026

SCION Transport in Tailscale

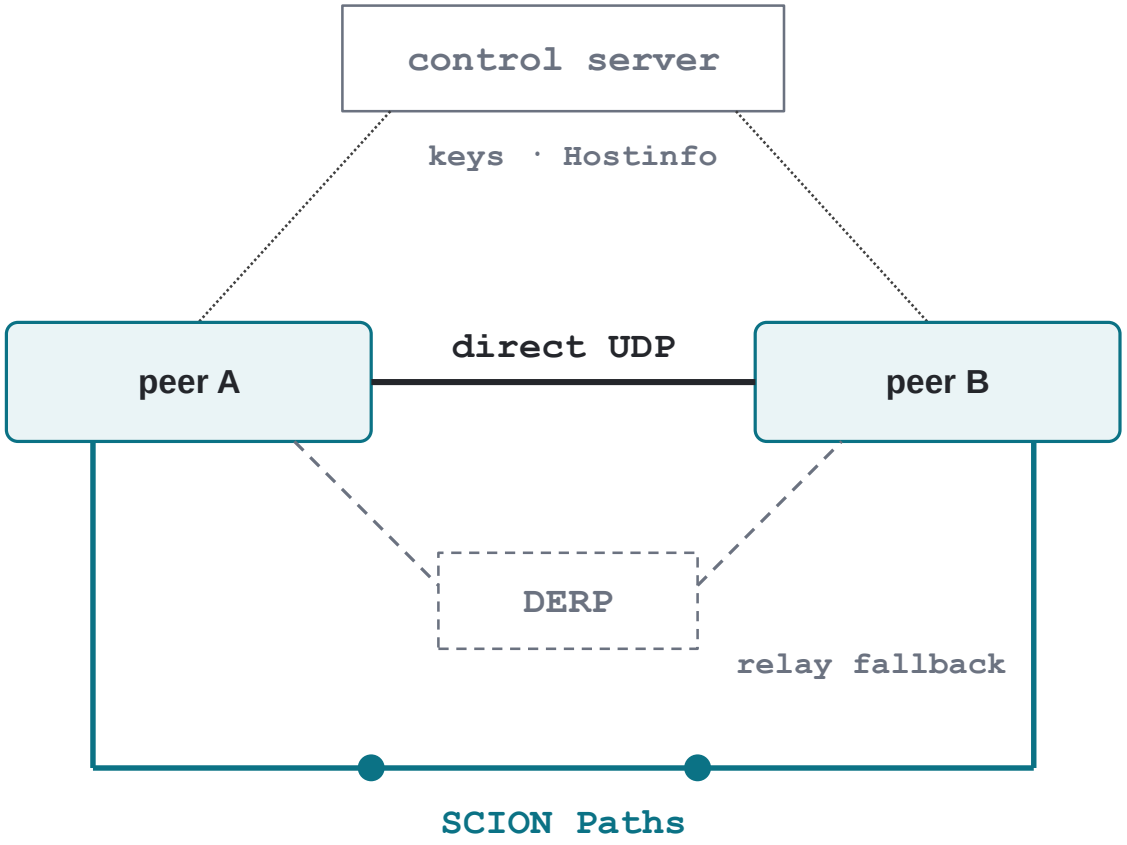
Adding SCION as a transport to a mesh VPN

Tony John
NetSys Lab · Otto von Guericke University Magdeburg

github.com/netsys-lab/tailscale-scion
github.com/netsys-lab/tailscale-android-scion

Tailscale already runs a path selection loop

- Tailscale = WireGuard mesh + Coordination
  - **`magicsock:`** UDP multiplexer per node, multiplexes UDP and DERP
  - **`epAddr:`** candidate endpoint set per peer: direct addresses and DERP
  - **`disco:`** ping/pong measures RTT per candidate
  - **`betterAddr:`** selects **`bestAddr`** per peer based on RTT; DERP relay when nothing direct works
- **Missing: Paths from a PAN**



This work: SCION paths as additional candidates. SCION stack in-process, selector unchanged.

Signaling SCION endpoints through an unmodified control server

- SCION addresses (ISD-AS:[IP]:Port) must be relayed to peers
- The control server only relays service types it knows
- A new **`tailcfg.SCION`** type would be dropped; modifying the control server was out of scope
- Solution: encode it in the **`peerapi4`** service description, which is relayed verbatim

1 advertise
`hi.Services[peerapi4]`

→

2 relay
`server unchanged`

→

3 parse
`scionServiceFromPeer()`

→

4 discover
`async path lookup`

→

5 register
`one epAddr per filtered path`

```
// peerapi4.Description wire format
scion=19-ffaa:1:eba,[10.0.0.1]:30041
     ISD-AS        underlay addr : port
```

Zero control-plane changes.
Works with Tailscale and Headscale servers as they are. Old peers ignore the field.

Path selection: three problems that come with many paths

Each SCION path is its own epAddr. It competes with UDP and DERP in the selector.
Three problems remain:

1 **Paths are filtered before they are measured**

Probing every path is too expensive. Candidates: sort by hop count, keep the 5 most distinct.
The path set is fixed before the first RTT is measured.

2 **Chasing the fastest path causes flapping**

Per-path latency is the median of an 8-slot ring.
A switch needs 20 % or 2 ms improvement, checked every 1 s.
Three timeouts demote a path. Probe runs round robin.

3 **Latency is the only property in the score**

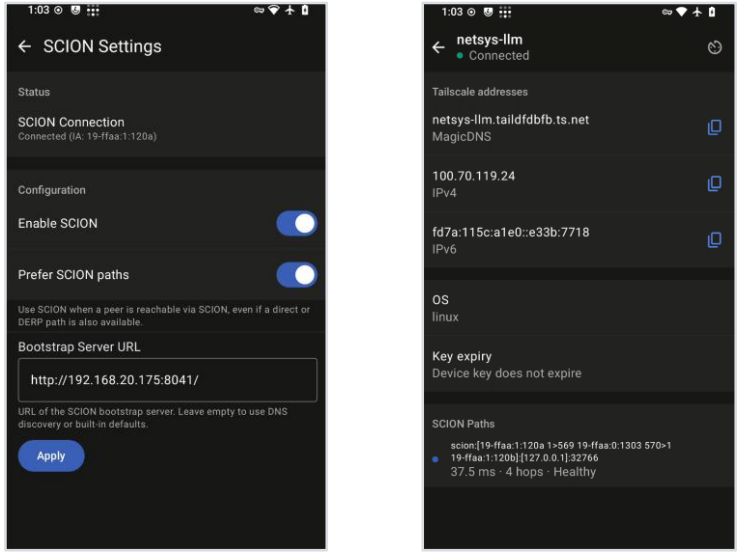
The selector ranks by RTT and address type, nothing else.

Running it

- Linux binaries and an Android APK, or build from source
- Config: topology.json + TRCs, or specify `TS_SCION_BOOTSTRAP_URL`
- SCION failure falls back to UDP or DERP, transparently.

```
$ curl --unix-socket …/tailscaled.sock …/localapi/v0/scion-status
{ "Connected": true, "LocalIA": "19-ffaa:1:eba" }
$ tailscale status
100.70.119.24 … active; direct scion:[19-ffaa:1:120a 1>1 19-
ffaa:1:120b]:[127.0.0.1]:32766
```

SCION path shown inline in peer status



Android: SCION settings and live per-peer path status

Findings and open questions

FINDINGS

1 **Endpoints with selection loops already exist at scale, but, sized for a handful of paths**

Mesh VPNs, ICE, MPTCP and QUIC, all assume a few candidates per peer. A path-aware network offers hundreds (Avg ~500 for SCION)

2 **Signaling was the easy part**

Existing metadata channels carry the path information. Zero control-plane changes

3 **Open problems are almost all path selection, probing and measurement**

How to filter and how many paths to consider, how often to probe them, how to trust noisy RTTs

OPEN QUESTIONS

- Which path properties (RFC 9473) justify their measurement cost in an endpoint selector?
- Probing under a budget: full-mesh cost grows with peers × paths. ID-INT[1] could help reduce cost. More on-going/prior works?
- Path preference is a per-device user toggle today. Should it sit with the user or the tailnet operator, and at what granularity?

Try it: `github.com/netsys-lab/tailscale-scion` · `github.com/netsys-lab/tailscale-android-scion` · feedback and issues welcome

1. Schulz, L. C., & Hausheer, D. (2024, October). ID-INT: secure inter-domain in-band telemetry. In *2024 20th International Conference on Network and Service Management (CNSM)* (pp. 1-9). IEEE.