

Discovering Provisioning Domain Names and Data

draft-ietf-intarea-provisioning-domains-00

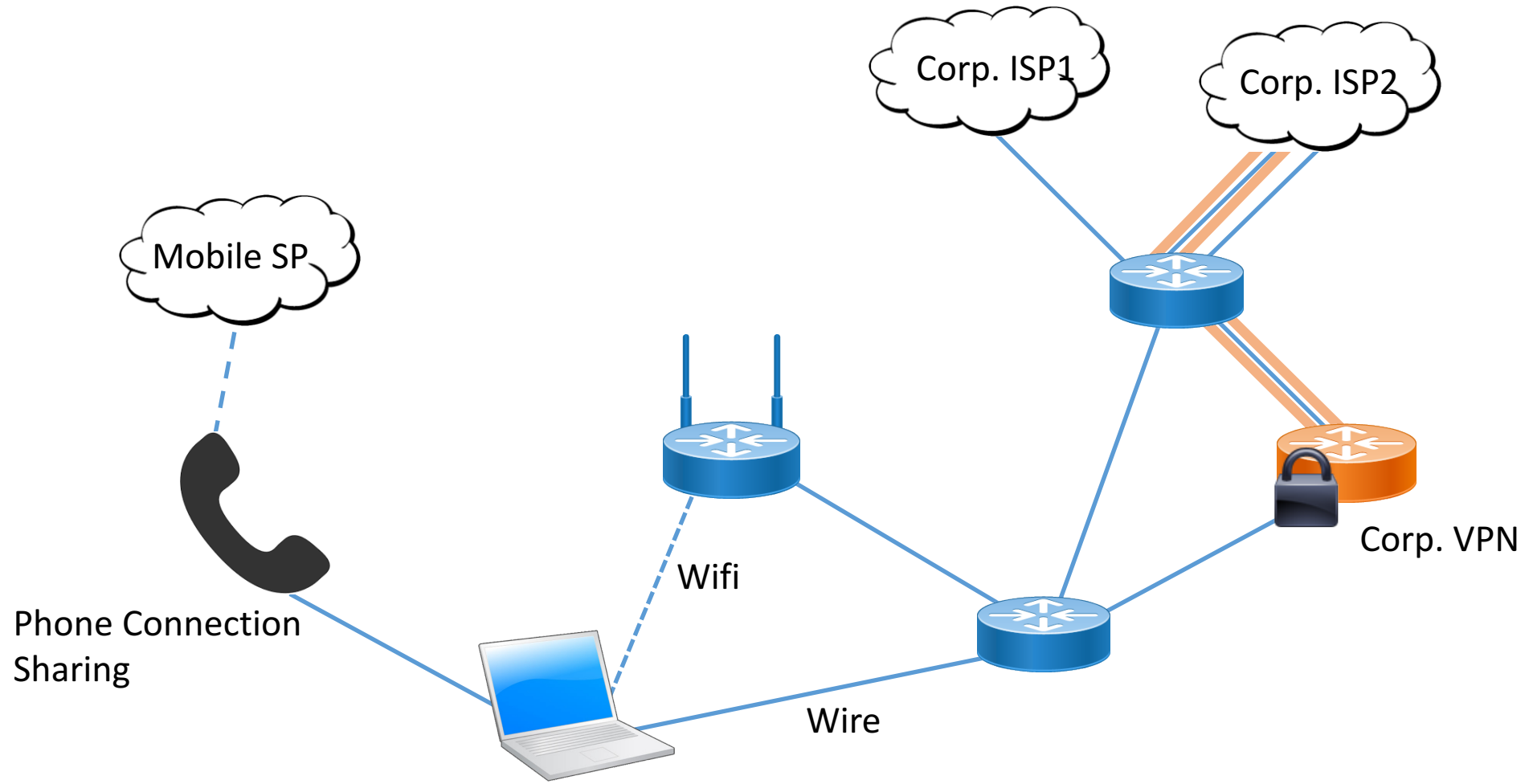
P. Pfister, **E. Vyncke**, T. Pauly, D. Schinazi, M. Keane

PANRG requests

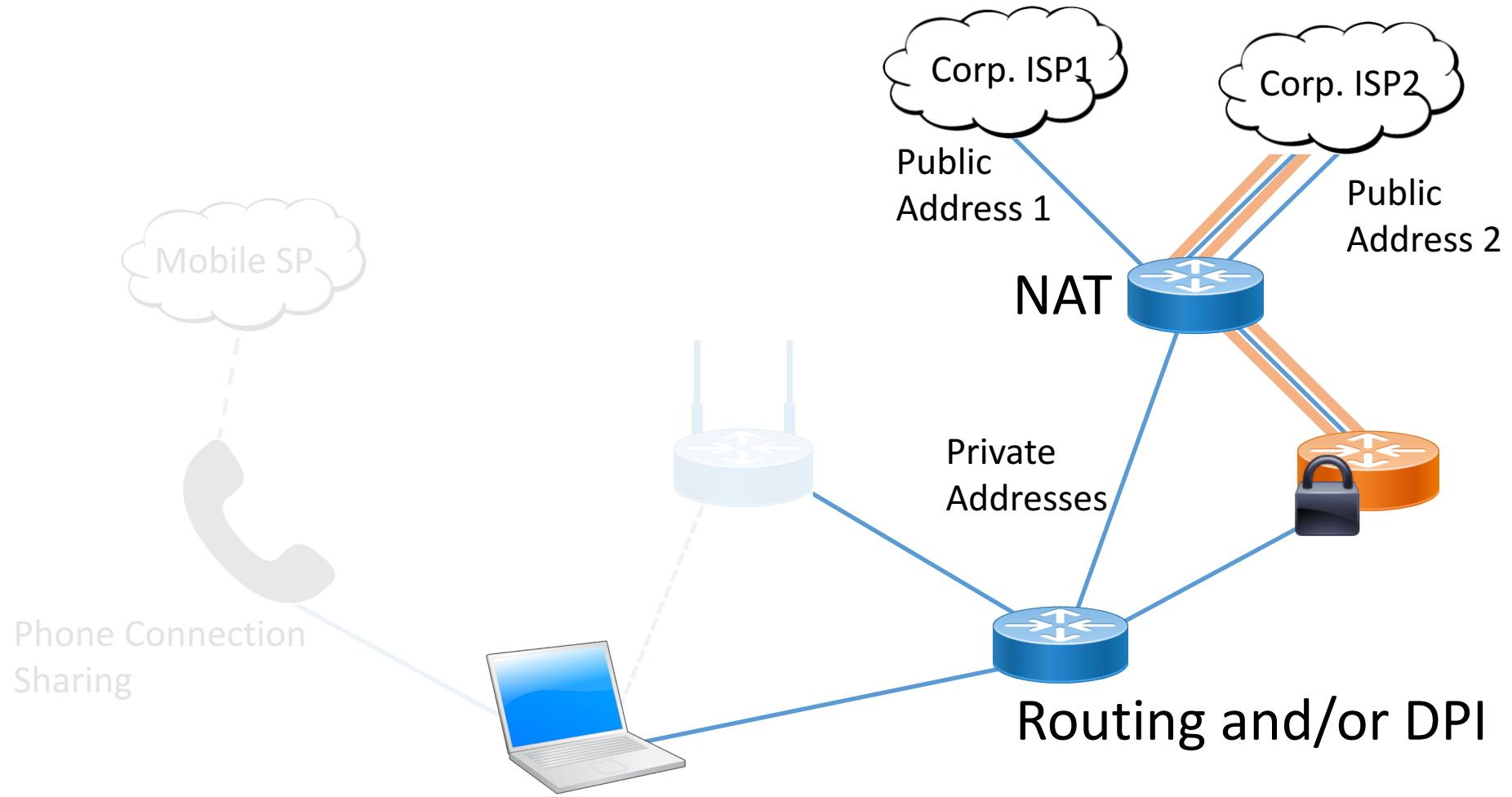
- I believe the PVD work is perfectly in scope of the RG as it covers bringing path awareness to transport and application layer protocols. It would be highly appreciated if someone could present at the PANRG session about PVDs.
- Multipath protocols taking advantage of this mobile connectivity begin to show us a way forward, though: if endpoints cannot control the path, at least they can determine the properties of the path by choosing among paths available to them. This research group aims to support research in bringing path awareness to transport and application layer protocols, and to bring research in this space to the attention of the Internet engineering and protocol design community.

Hosts and networks are multi-homed

Just a few examples...



Multi-Homing, the legacy way...

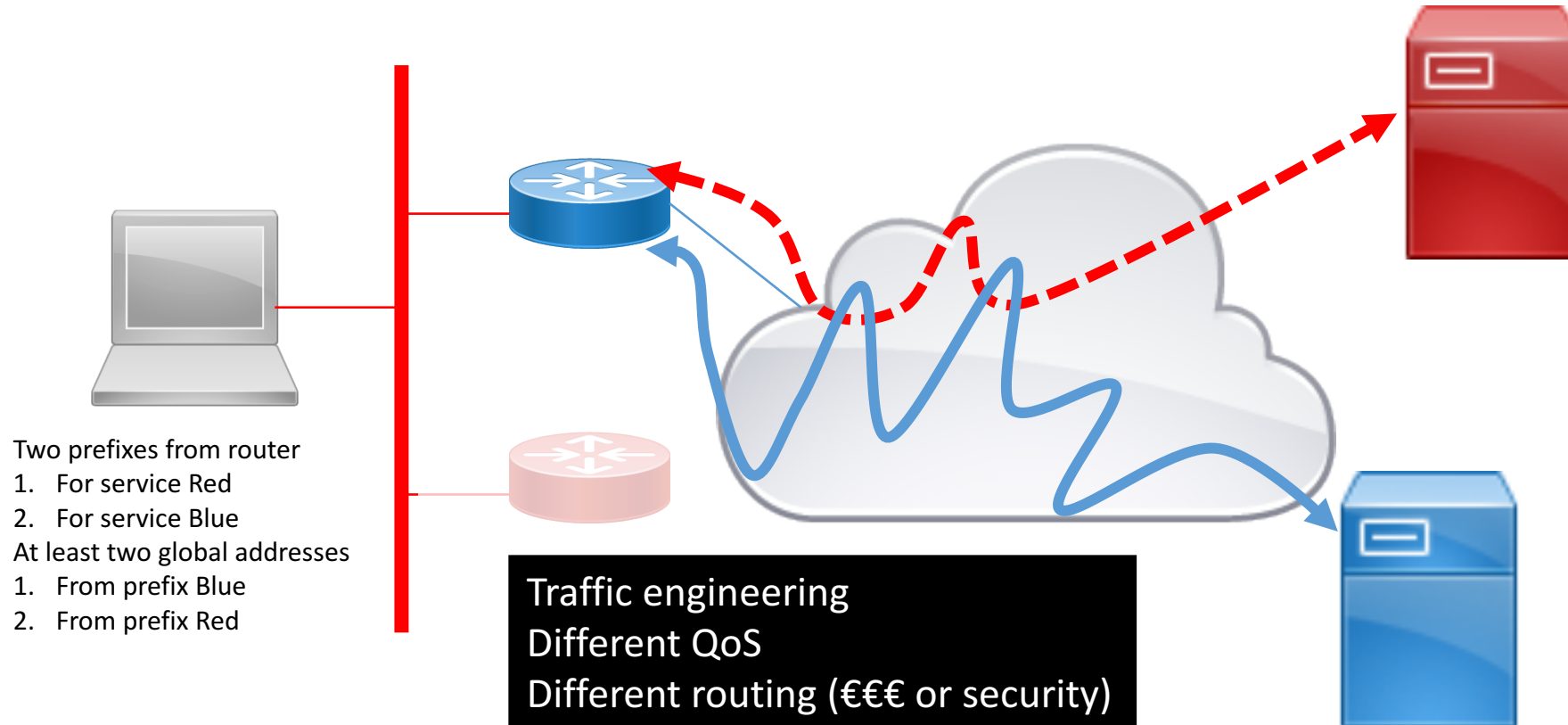


Bundling IP address & DNS resolver

Multihoming and CDNs

- Name lookups for resources stored on CDNs give different answers depending on the network connection
- Host on homenet may look up name using resolver from provider A, then connect to CDN using provider B
- This will generate support requests
- What to do?

Service Selection



The purpose of this draft is to:

1. Identify Provisioning Domains (PvDs).

[RFC7556] Provisioning Domains (PvDs) are consistent sets of network properties that can be implicit, or advertised explicitly.

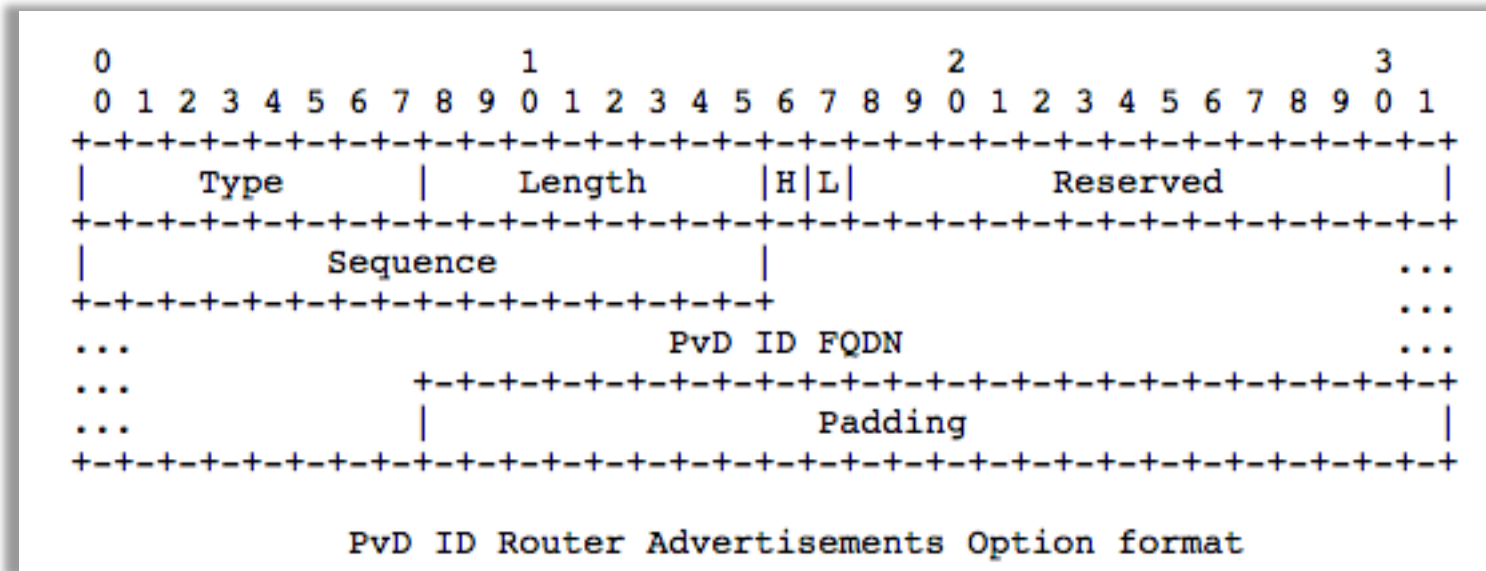
Differentiate provisioning domains by using FQDN identifiers.

2. Give PvD Additional Information.

Name, characteristics, captive portal, etc...

Step 1: Identify PvDs

With the PvD ID Router Advertisement Option

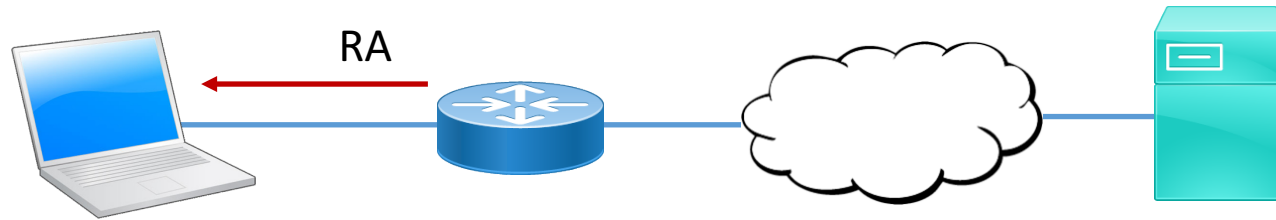


- At most **one occurrence** in each RA.
- **PvD ID** is an **FQDN** associated with options in the RA.
- **H bit** to indicate **Additional Information is available with HTTPS**.
- **L bit** to indicate the **PvD has DHCPv4 on the link**.
- Seq. number used for **push-based refresh**.

Step 1b: Identifying PvD (Cont.)

- Information in a RA without PvD ID is linked to an implicit PvD (identified by interface & link-local address of router)
- Option in RA can change of PvD when they are received in a RA with a different PvD ID
- DHCPv6 information **MUST** be associated to a PvD ID received on the same interface from the same link-local address

Step 2: Get the PvD Additional Data

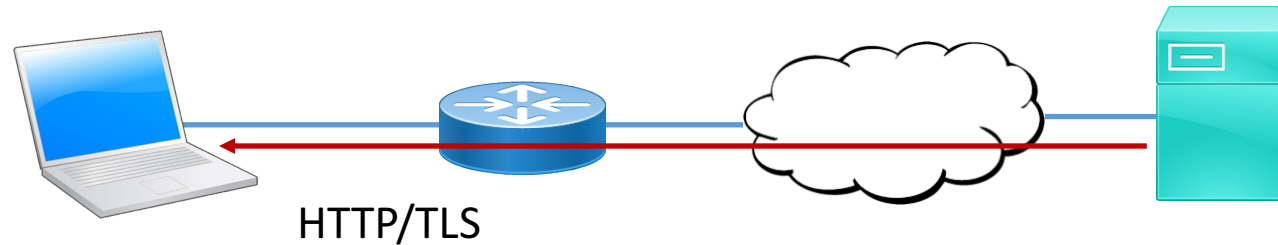


When the H bit is set:

GET <https://<pvd-id>/.well-known/pvd>

Using network configuration (source address, default route, DNS, etc...) **associated with the received PvD.**

Step 2: Get the PvD Additional Data



When the H bit is set:

GET https://<pvd-id>/.well-known/pvd

Using network configuration (source address, default route, DNS, etc...)
associated with the received PvD.

Step 2: Get the PvD Additional Data

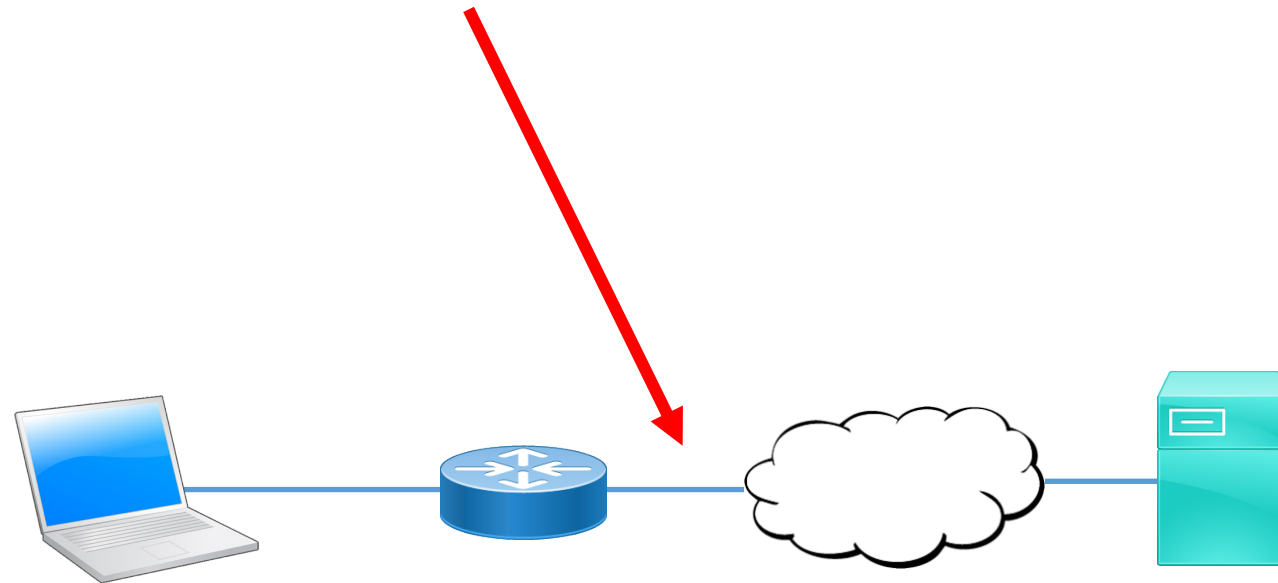
```
{
  "name": "Foo Wireless",
  "expires": "2017-07-23T06:00:00Z",
  "prefixes" : ["2001:db8:1::/48", "2001:db8:4::/48"],
  "localizedName": "Foo-Hôtel à Paris Wifi",
  "dnsZones": ["example.com", "sub.example.com"];
  "characteristics": {
    "maxThroughput": { "down":200000, "up": 50000 },
    "minLatency": { "down": 0.1, "up": 1 }
  }
}
```

Some other examples (see also <https://smart.mpvd.io/.well-known/pvd>) :

```
noInternet : true,
metered : true,
captivePortalURL : "https://captive.org/foo.html"
```

Step 2: Additional Data Describing the Network

- Cost of the network access
- Performance of the first uplink (ADSL, FTTH, ...)
- Captive portal
- Walled garden
- ...



Implementation status

Linux - <https://github.com/IPv6-mPvD>

- pvdd: A Daemon to manage PvD IDs and Additional Data
- Linux Kernel patch for RA processing
- iproute tool patch to display PvD IDs
- Wireshark dissector
- RADVD and ODHCPD sending PvD ID

Implemented in on commercial vendor router

neat

A New, Evolutive API and Transport-Layer Architecture for the Internet: <https://www.neat-project.org/>

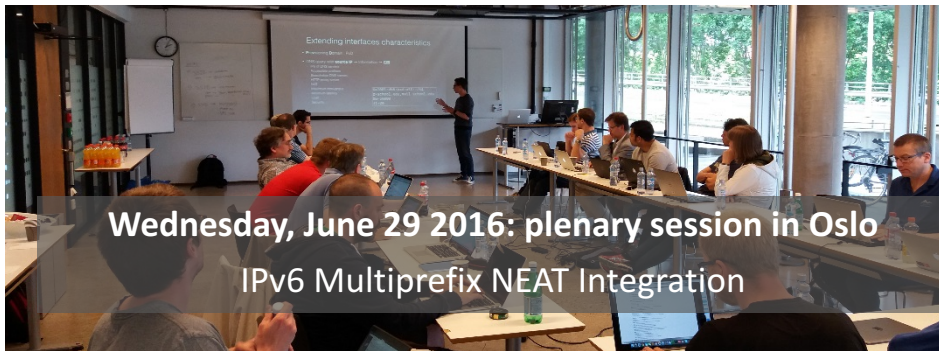
European H-2020 project











10 partners (Cisco, Mozilla, EMC, Celerway...)

Provisioning Domain (information about a prefix) via DNS [draft-stenberg-mif-mpvd-dns-00](#) (old)

Integration to NEAT code: <https://github.com/NEAT-project/neat/pull/80>

Asking the user to choose with relevant criteria and simple UI



LTE (ORANGE)			
	 2 mn	\$ 0.5 GB \$0	 3%
VPN OVER LTE (ORANGE)			
	 6 mn	\$ 0.5 GB \$0	 4%
Wi-Fi (OSLO HOSTEL Wi-Fi)			
	 11 mn		 1%