Why do standards matter?

The beauty about standards is that there are more than one to choose from….

How many engineers does it take to plug a laptop into a socket?

Based on a true story…

1st Attempt

Laptop plug

Wrong shape

Hotel room socket

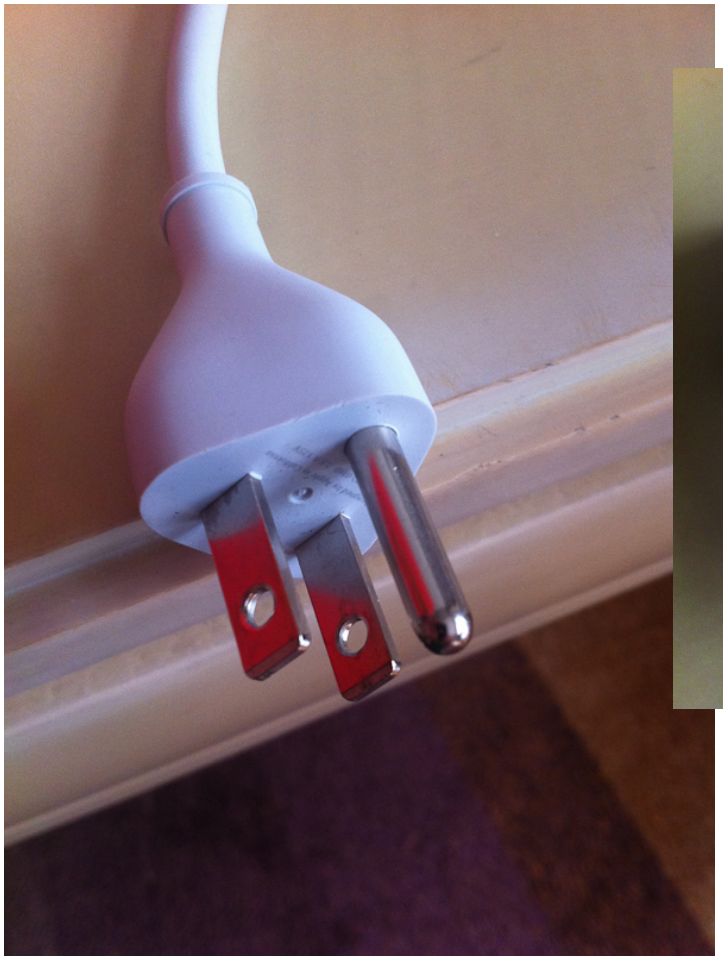
2nd Attempt

Laptop plug

No ground

“multi-protocol” socket

3rd Attempt



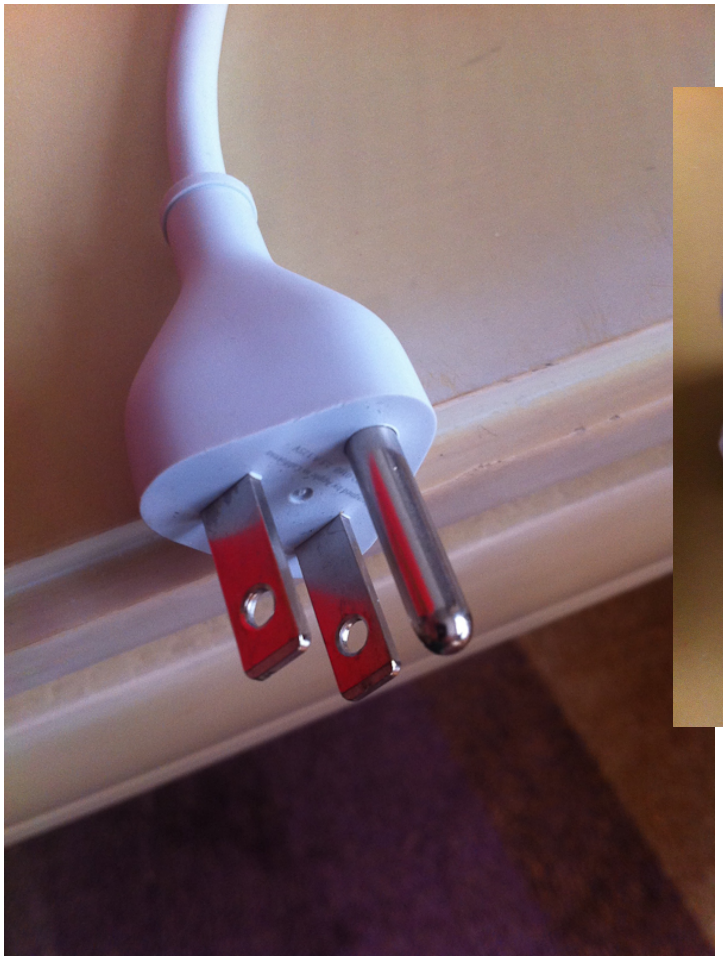
Laptop plug

AU adaptor

Ground safety prong missing

“multi-protocol” socket

4th Attempt



Laptop plug

UK adaptor

Wrong shape

“multi-protocol” socket

5th Attempt

Laptop plug

US adaptor

"Hot" prong too wide

"multi-protocol" socket

6th Attempt

Laptop plug

Victory!

“multi-protocol” socket v2.0

LET’S NOT DO THAT AGAIN…

Taking stock from the Softwire Interim meeting

Charting the “stateless” problem space

Don’t say stateless…

- **3 characteristics**
  A. Per-flow NAT binding on CPE vs CGN
  B. Per-subscriber mapping
     1 Hub & Spokes, on PRR
       (static or learned)
     2 Hub & Spokes, on PRR
       (derived from address mapping rules)
     3 Mesh, on CPE
       (derived from address mapping rules)
  C. Translation vs Encapsulation

A) per-flow NAT bindings

- On CPE
- Do not over-optimize port allocation mechanisms

➔ If you need flexibility on port distribution, use a CGN

AGREED

B1) per-subscriber state on **PRR** (static or learned)

- High level:
  - **Scattered IPv4 address space**
  - **Do not overload CPE with a large number of mapping rules**
  - Hub & spoke model

- No use of algorithmic mapping rules
  - neither on CPE nor on PRR
- **Per user state on PRR**
- Derivative of DS-Lite/4over6 using DHCP or PCP port distribution

B2) per-subscriber mappings on **PRR** (derived from address mapping)

- Same as B1), **but without per-user state**

- Use address & port mapping algorithmic rules as “forwarding function” on PRR

B3) per-subscriber mappings on **CPE** (derived from address mapping)

- Mesh model
- Use of provisioning and forwarding algorithmic rules on CPE
- Consensus to converge toward a unified address & port mapping algorithm

➔ Can the unified address & port mapping algorithm be specified to include B2 case?

  - No rules on CPE:
    send all traffic to AFTR tunnel endpoint

C) Translate vs Encap

- Technically, mostly the same
  - Minor differences: overhead vs loss of information

- Operational perspective
  - **Discussion** on what is easier to do for on-path processing (QoS/ACL/…):
    - Look at IPv6-translated headers in middle points
    - Look at encapsulated IPv4 headers in middle points
    - Look at decapsulated IPv4 headers in end points only

Way Forward

Way Forward, part 1:

Way Forward, part 1:

- Design team to propose a unified address & port
  - Design team to propose a unified address & port mapping algorithm
  - Target: PS document
    - Applicable to both H&S or Mesh and Encapsulation or Translate
- Define unified DHCPv6 options for the above
  - Define unified DHCPv6 options for the above
  - Target: PS document
    - Applicable to both H&S or Mesh and Encapsulation or

Way Forward, part 2

Way Forward, part 2

- One or multiple approaches?
  - Impact on interoperability
- One or multiple approaches?
  - What to implement on CPEs ?
  - (Std track, all the RFCs
- If we adopt, do they all have the same status?

Since the Interim

- Formed MAP design team (chaired by Ole)
  - 2 drafts: MAPing & DHCP option
- Lots of discussions
  - SD-NAT __ -U,…)
  - SD-NAT