TC “Tunnel Configuration”

Problem Statement

Thomas Narten, IBM
Alain Durand, ???

Problem Statement

- IPv6 in the core, Access network (e.g. DSLAM) at the edge is not IPv6 capable.
- Question: how to set up an IPv6 over IPv4 tunnel between a home site and the core?
  - Need to agree on tunnel properties
    - Authentication
    - MTU
    - Encapsulation
  - Need to find out the end point (orthogonal issue)
    - Statically configured?
    - Automatically discovered?

ISP assumptions

- Core networks has been upgraded to v6
- ISP has obtained v6 address space from registries
- Access networks are not yet upgraded
- ISP does not (yet) offer v6 native to “all” customer
- ISPs has POPs throughout the world
  - Customer should use the tunnel end point “close to” the POP it is connected to.
  - Tunnel end point may change as the user roams within the ISP

Customer network assumption

- IPv4 only connection to ISP
- Dual stack node to set up the tunnel, possible scenarios:
  - a single node, directly attached to the ISP access network
  - a router, directly attached to the ISP access network
  - a node or router, behind an unmodifiable IPv4-only customer owned NA

Why negotiate?

- Need to work on all kind of access media
- It is not possible to define a one-size-fit-all
- Possible parameters to negotiate:
  - Authentication may or may not be required
  - Encryption of tunnel content may or may not be required
  - OAM, e.g. Keep alive, NAT state refresh,...
  - Encapsulation type (IP in IP, GRE,...)
  - MTU may need to vary

End Point Discovery

- Orthogonal issue to tunnel config
- Manual input of a DNS name / IP adress solves the problem from a tunnel config perspective
- Automatic discovery is “difficult” in the general case (see presentation later in the agenda)
- Other applications (SMTP, SIP,...) do not provide automatic discovery.
- If it is a generic problem, why solve it in this particular WG?

Tunnel encapsulation (v6 over v4 case)

- IPv6 over IPv4 (not NAT friendly)
- IPv6 over UDP over IPv4
  - Demultiplexing based on IP address/UDP port
  - May not scale due to NAT issues (beyond 64k)
  - Should encapsulation support 100k+ sessions per tunnel end point?
- Specialized encapsulation:
  - Allow tunnel end point to specify a demux key (e.g. à la L2TP or IPsec SPI)

Open Discussion

- Should this WG do something specific for the scenario described in the problem statement or something more generic that handles different types of encapsulation?
- Can we use off-the-shelf tunneling encapsulation?
  - IPv6/UDP/IPv4 by default?
- Are existing off-the-shelf solutions adequate for the task?
- Should this WG address the tunnel end point discovery part of the problem?