

# PCP Failure Scenarios

**draft-boucadair-pcp-failure**

IETF 82-Taipei, November 2011

M. Boucadair, F. Dupont, R. Penno

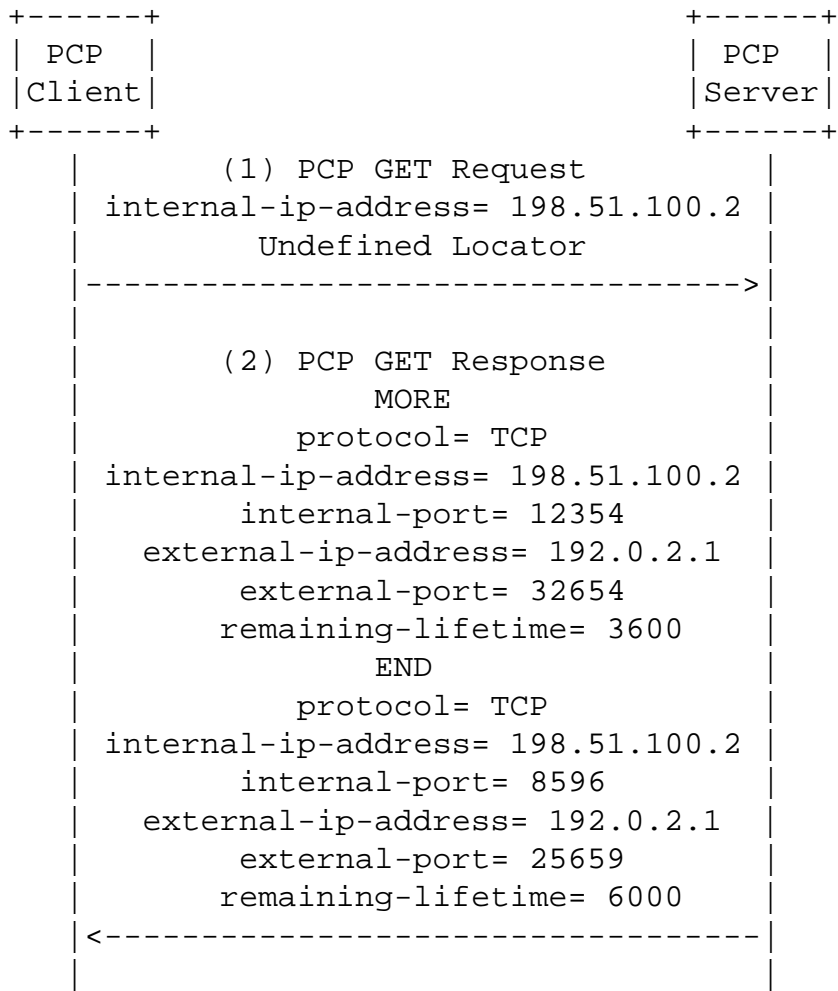
# Scope

- Document PCP failure scenarios:
  - PCP Client crash
  - Application crash
  - PCP Server failures
    - Discussion on PCP Server redundancy mode: (1) State Redundancy is Enabled, (2) Cold-Standby without State Redundancy and (3) Anycast Redundancy Mode
  - Change of the IP address of the CPE WAN I/F
    - e.g., how an IPv4 host connected to a DS-Lite CPE is aware that a new IPv6 address is used by the B4?
  - Host failure
  - Change of Internal IP address (3<sup>rd</sup> party case)
- Some failure modes may lead to stale mappings and therefore burn out per-user quota
  - Access to the service may be impacted
- Document a mechanism for state synchronization purposes between client and server

# State Synchronization Procedure

1. One element (i.e., PCP Client/host/application, PCP Server, PCP Proxy, PCP IWF) of the chain is **REQUIRED** to use stable storage
2. If the PCP Client (resp., the PCP Server) crashes and restarts, it synchronizes with the PCP Server (resp., the PCP Client)
3. If both crash, then one has to use stable storage and we fall back in the previous case as soon as we know which one (the Epoch value provides this information)
4. PCP Server  $\Rightarrow$  PCP Client non-disruptive synchronization requires a **GET/NEXT** mechanism to retrieve the state from the PCP Server;
  - Without this mechanism the only way to put the PCP Server in a known state is for the PCP Client to send a `delete all request`, a clearly disruptive operation
5. PCP Client  $\Rightarrow$  PCP Server synchronization is done by a re-create or refresh of the state
  - The PCP Client **MAY** retrieve the PCP Server state in order to prevent stale explicit dynamic mappings

# GET/NEXT Flow Example



# Next Steps

- Comments are welcome
- WG adoption?