PCP Failure Model

Mohammed Boucadair,
Francis Dupont,
Reinaldo Penno
Plan

- Problem statement
- Failure cases
- Synchronization
- GET/NEXT
- Open Questions
Problem Statement

• No explicit dynamic mapping should be lost
• The PCP Server should not have mappings unknown by the PCP Client (*Stale* mappings, in fact a synchronization problem)
Failure Cases (I)

- PCP Client crashes
- PCP Server crashes
- Both PCP Client and Server crash
Failure Cases (2)

• If one crashes, the state (explicit dynamic mapping table) is still available at the other end

• If both crash, the operational requirement is to have stable/persistent storage at either PCP Client or Server

• Easy extension to a chain with PCP Proxies
Synchronization (1)

• The PCP Client creates/renews/refreshes all its explicit dynamic mappings by sending MAP requests: the Client image will be included in the Server image

• It is the standard action when the PCP Server has crashed and reset the Epoch value to zero
Synchronization (2)

- The PCP Client sends a *delete all* MAP request: the Server image is reset to the empty state
- Formally it works but it is sure it is not what users really want...
Synchronization (3)

• Add a new operation which allows the PCP Client to download the PCP Server image
GET/NEXT

- A new OpCode and a new Option
Open Questions (1)

• With more than one PCP Client on a host they can conflict (no way to recognize/select the owner)

• An InterWorking Function without stable storage can’t recover its state after a crash (*stale* mappings become *orphan* mappings)

• Common rejected solution
Open Questions (2)

• And security requirements?

• A CGN MUST NOT lose explicit dynamic (and static) mappings (mapping theft)