ADNCP = ?

Markus Stenberg
DNCP in a nutshell (1/2)

State (shared)

.. an ordered set of TLVs..

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>

<network hash>
DNCP in a nutshell (2/2)

1a. network state (Trickled multicast)

Node 1

2. request node data

Node 2

1b. network state (Trickled unicast)

3. node data
DNCP

• Generalized from HNCP, which was intentionally limited and simple

• Single area link-state database, i.e.
  • all nodes share the same data
  • .. which includes connectivity graph (required to minimize traffic in stable state; another option would be e.g. TTL-based scheme)

• Trickle based synchronization (multi/unicast)

• Merkle tree (network hash -> node hash -> node data)

• DNCP leaves some details to DNCP profiles
  • transport, node identifier length/derivation/collision handling, hash..
DNCP scalability issues

1. Single area

2. If UDP transport is being used, packet size sets low, hard limits
   - RFC2460 => ‘few’ nodes with UDP transport (1500 octets) - even with small node identifier/node data hash
   - Similarly, single node can only advertise ~1400 octets of data
ADNCP = (Autonomic) ..

- Extends DNCP with (zero/little-config) areas => Addresses scaling issue #1
- TLS-only for unicast, UDP for multicast => Addresses scaling issue #2
- PKI-only = Compatible with the bootstrap draft
- Objective-specific specifications can add point-to-point exchanges which can leverage the existing secure TLV-based transport for
  - discovery, negotiation, synchronization
- Provides some guidance on how to do distributed, graph-based
  - discovery, negotiation, synchronization, intent distribution
- (And multi-party algorithms in general; Prefix Assignment provided as an example)
Anima + DNCP != ADNCP?

DNCP only in ‘core’ (1/2)

• Single DNCP area in the "core" of the network

• Something bit more request-reply-ish on the edges

• Should scale reasonably well given limited number of core nodes
Anima + DNCP != ADNCP?

Per-objective DNCP (2/2)

• The DNCP network(s) may share secure transport with the non-DNCP methods (IPsec, TLS, something else)

• The networks should be small and be isolated from each other (in terms of DNCP state)
Questions?