ANIMA
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2015

M. Stenberg

March 5, 2015

Autonomic Distributed Node Consensus Protocol
draft-stenberg-anima-adncp-00

Abstract

This document describes the Autonomic Distributed Node Consensus Protocol (ADNCP), a profile of Distributed Node Consensus Protocol (DNCP) for autonomic networking.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 6, 2015.

Table of Contents

1. Introduction

DNCP [I-D.ietf-homenet-dncp] provides a single-area link state database for arbitrary use. ADNCP extends DNCP in several ways and makes it implementable by defining a profile.

ADNCP allows for several types of point-to-point exchanges that match typical autonomic operations. The shared state within ADNCP itself is used to also facilitate some autonomic operations. Whether point-to-point or multi-party algorithms are used is left up to the specification of particular objectives.

To provide for better scalability than the base DNCP, ADNCP also defines (optionally zero-configuration) multi-area system.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Terminology

Reader is assumed to be familiar with the autonomic networking terminology described in [I-D.irtf-nmrg-autonomic-network-definitions] and [I-D.ietf-homenet-dncp].

(ADNCP) area: A set of ADNCP running nodes that are directly connected using a set of DNCP connections. In other words, DNCP network. They share a link state database, and may also have some other data from other areas but no actual topology of the other areas.

(ADNCP) network: A set of connected ADNCP areas.

area owner: The ADNCP node with the highest Node Identifier within the ADNCP area.

connection owner: Either ADNCP node with the highest Node Identifier on a multicast-capable link the connection maps to, or the unicast "server" node that other nodes connect.

per-area: Applicable to the nodes in a particular area.

area-wide: Distribution scope in which content is made available to nodes in only one area.

per-net: Applies to the whole (ADNCP) network.

net-wide: Distribution scope in which content is made available to nodes in all areas.

4. DNCP Profile

ADNCP is defined as a profile of DNCP [I-D.ietf-homenet-dncp] with the following parameters:

- ADNCP uses UDP datagrams on port ADNCP-UDP-PORT as a multicast transport over IPv6 using group All-ADNCP-Nodes-6, or IPv4 using group All-ADNCP-Nodes-4. TLS [RFC5246] on port ADNCP-TCP-PORT is used for unicast transport. Non-secure unicast transport MUST NOT be used and therefore is not defined at all. In a typical case, multicast transport SHOULD be link-local scoped, although other scopes MAY be also used and supported if multicast routing is available.

- ADNCP operates over either unicast connections, or over multicast-capable interfaces. Therefore the value encoded in the DNCP Connection Identifier is left up to the implementation.

- ADNCP nodes MUST support the X.509 PKI-based trust method, and MAY support the DNCP Certificate Based Trust Consensus method.

- ADNCP nodes MUST use the leading 128 bits of SHA256 [RFC6234] as DNCP non-cryptographic hash function H(x).

- ADNCP uses 128-bit node identifiers (DNCP_NODE_IDENTIFIER_LENGTH = 128). A node implementing ADNCP MUST generate their node identifier by applying the SHA256 to their public key. If the node receives a Node State TLV with the same node identifier and a higher update sequence number multiple times, an error SHOULD be made visible to an administrator.

- ADNCP nodes MUST NOT send multicast Long Network State messages, and received ones MUST be ignored

- ADNCP nodes use the following Trickle parameters:

  * k SHOULD be 1, given the timer reset on data updates and retransmissions should handle packet loss.

  * Imin SHOULD be 200 milliseconds but SHOULD NOT be lower. Note: Earliest transmissions may occur at Imin / 2.

  * Imax SHOULD be 7 doublings of Imin (i.e. 25.6 seconds) but

SHOULD NOT be lower.

- o ADNCP nodes MUST use the keep-alive extension on all multicast interface-based connections. The default keep-alive interval (DNCP_KEEPALIVE_INTERVAL) is 20 seconds, the multiplier (DNCP_KEEPALIVE_MULTIPLIER) MUST be 2.1, the grace-interval (DNCP_GRACE_INTERVAL) SHOULD be equal to DNCP_KEEPALIVE_MULTIPLIER times DNCP_KEEPALIVE_INTERVAL.

5. Point-To-Point Operations

For point-to-point operations such as discovery, negotiation, and synchronization, a single new class of DNCP messages is defined (TBD - more detail?). It is identified by the presence of an objective-specific TLV, and if specified by the objective, it SHOULD be responded to only via unicast at most. Therefore, if an ADNCP implementation does not recognize a message, it MUST be silently ignored. These messages SHOULD NOT in and of themselves establish a

DNCP-style bidirectional peering relationship between nodes, and therefore SHOULD NOT contain Node Connection TLV..

Such objective-specific messages should either define some transaction id scheme (TBD - should it be here), or include the request verbatim within the replies, if any.

6. Distributed Operations

6.1. Discovery

If point-to-point discovery (using either multicast-capable interface(s), or known unicast peers) is not chosen, discovery can be handled also either by participating in the ADNCP network, or by performing point-to-point operation with a node participating in the ADNCP.

Presence (or lack) of content with ADNCP can be used to discover nodes that support particular objectives in some specific way; for example, an objective might specify TLV which contains an address of some particular type of server (for example, DHCPv6 PD), and therefore by just using ADNCP information, "closest" node (in terms

of areas / in terms of routing of the address) could be determined.

6.2. Negotiation / Synchronization

ADNCP is not suitable for (especially net-wide) transmission of any data that changes rapidly. Therefore it should be used to sparingly publish data that changes at most gradually.

With that limitation in mind, ADNCP can be used to implement arbitrary multi-party algorithms, such as Prefix Assignment [I-D.ietf-homenet-prefix-assignment]. Given appropriate per-area hierarchical assignment (published net-wide), it could be also employed net-wide though, as the per-net prefix assignments would change only rarely.

For rapidly changing data, point-to-point exchanges (as needed) should be used instead and just e.g. relevant IP addresses published via ADNCP.

6.3. Intent Distribution

Arbitrary (operator-supplied) objective-specific intent can be supplied as TLVs within ADNCP, either per-area or per-network.

7. Area Support

Area support for DNCP is added so that non-area-capable implementations can benefit from it, but cannot support more than one interface (for same DNCP instance at any rate), as they cannot handle the logic for transferring data between areas.

Areas are uniquely identified by a 32-bit Area Identifier.

7.1. Area Boundaries

A single connection always belongs to exactly one area. Therefore the boundaries of the areas are within nodes that have multiple connections, and can transfer data between them.

For every remote area detected (=on other connections, not on that particular connection), a node should include a Remote Area TLV which contains an Area Identifier, a Node Identifier of the area owner, and pared down (recursive) list of Remote Area TLVs from that area, that MUST be loop free. An exception to the rule is the current area; if the current area is advertised elsewhere, it MUST be included if and only if the owner's Node Identifier differs from the local one. Longer paths to particular areas with matching owner Node Identifier MAY be also omitted.

TBD: Remote Area TLV - area id, area owner (+container for more Remote Area TLVs recursively)

7.2. Area Identifier

Area Identifier for every connection is chosen by the connection owner. The link is owned by the node with the highest Node Identifier on a connection which consists of a multicast-capable link, or the "server" node which other nodes are connecting to in case of an unicast link.

TBD: Area Identifier TLV - just area id - originated by the area owner, and then included in every unicast message on link.

7.3. Area Formation

Areas by definition are connected parts of the network. An operator may set explicit values for the Area Identifiers, thereby forming the areas, or alternatively an automatic formation process described here can be used by the connection owners. Non connection owners on a particular connection should simply follow the connection owner's lead.

If the connection owner does not have an area on a particular connection yet, it may use an existing area from some other connection if and only if following suitability criteria are met:

- The current set of links covered by that area (calculated by traversing through the neighbor graph) is not more than TBD.
- The number of nodes in that area is not more than TBD.

- The area owner does not publish an Area Full TLV.

If nothing suitable is present, areas connected directly to other nodes within the area can be also considered. For them, the suitability criteria are:

- A node within current area exists which publishes Remote Area TLV with the Area Identifier of the area.

- No published Area Full TLV for the area.

If choosing to use a particular area, the node MUST wait random [TBD1, TBD2] seconds before making the actual assignment, and ensure that the suitability criteria are still matched when it makes the assignment. If not, this process should be repeated again, starting from evaluating the candidates.

If no area is found at all, a new area should be created, with a random delay of [TBD1, TBD2] seconds before announcing. At the end of the interval, the presence of available areas to join should be checked before publishing the Area Identifier TLV.

Once the area owner notices that the directly connected suitability criteria enumerated above are no longer filled by the local area (=it is too large), the area owner MUST publish an Area Full TLV. It MAY be removed at later point, but if and only if the area is substantially below the maximum desired size in terms of number of links and number of nodes.

If the owner of an area detects the presence of a Remote Area TLV with an Area Identifier identical to that of the area it is advertising and with an owner having a higher Node Identifier than itself, then the area owner MUST choose a new (random) Area Identifier.

TBD: Area Full TLV - no content, but net-wide.

7.4. Import/Export

There is no explicit exporting of TLVs; any TLV type that has highest bit set (0x8000) will be considered area-originated, and spread net-wide, as opposed to the default area-wide node-originated. It is important to note that currently node identifier of the originating node is lost as it transitions to another area (TBD), but within the area the originator is still visible.

Given the node is on an area boundary, for all areas it is in, it must recursively traverse all Remote Area TLVs announced within the area, and keep track of the shortest recursion depth at which a particular area is first encountered. The Node Identifier of the Remote Area TLV originator is used for tie-breaking, with the higher one preferred. If encountering Remote Area TLV with the local area's Area Identifier, that TLV MUST NOT be recursed into to avoid loops.

For any areas for which the node is identified as the importer (by having shortest path of areas, or winning tie-break), the node MUST import Remote Area Content TLV from the first-hop remote area verbatim if there are other areas on the path. If the node is directly connected to the remote area, it MUST create and maintain Remote Area Content TLV which contains all TLVs marked for export.

When Remote Area Content TLV changes, or is no longer present in the "upstream" area, it must be also updated/removed by the importer.

TBD: Remote Area Content TLV - area id (+container for any exported TLVs from that area)

8. Security Considerations

TBD

9. IANA Considerations

TBD - TLVs values here + ADNCP-UDP-PORT, ADNCP-TCP-PORT

All-ADNCP-Nodes-4, All-ADNCP-Nodes-6

10. References

10.1. Normative references

[I-D.ietf-homenet-dncp]
Stenberg, M. and S. Barth, "Distributed Node Consensus Protocol", draft-ietf-homenet-dncp-00 (work in progress), January 2015.

[RFC6234] Eastlake, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, May 2011.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

10.2. Informative references

[I-D.ietf-homenet-prefix-assignment]
Pfister, P., Paterson, B., and J. Arkko, "Distributed Prefix Assignment Algorithm", draft-ietf-homenet-prefix-assignment-03 (work in progress), February 2015.

[I-D.irtf-nmrg-autonomic-network-definitions]
Behringer, M., Pritikin, M., Bjarnason, S., Clemm, A., Carpenter, B., Jiang, S., and L. Ciavaglia, "Autonomic Networking - Definitions and Design Goals", draft-irtf-nmrg-autonomic-network-definitions-05 (work in progress), December 2014.

Appendix A. Open Issues

Should hierarchical PA be defined here or not? [I-D.ietf-homenet-prefix-assignment], with cross-area hierarchical extension, would facilitate even very large scale PA (with potentially multiple upstreams). Perhaps the current mention is enough.

Should areas importers / area ID choice TLVs include precedence value?

Should we include node-data signatures or not? They improve security, but are not visible across areas in any case - it would need per-TLV signature(!) in that case with a hefty footprint due to needing to include way to identify the public key too. So I think not.

Should some way to publish certificate id / raw public key be defined? So it can be verified that e.g. node identifier is really generated based on one. Perhaps..

Should some sort of more granular delta transfer scheme be defined? For a large network, the current scheme's TLV set published by a

single node can grow to substantial size. This may occur either here or in DNCP.

Appendix B. Changelog

draft-stenberg-anima-adncp-00: Initial version.

Appendix C. Draft Source

As usual, this draft is available at https://github.com/fingon/ietf-drafts/ in source format (with nice Makefile too). Feel free to send comments and/or pull requests if and when you have changes to it!

Appendix D. Acknowledgements

Thanks to Pierre Pfister, Mark Baugher and Steven Barth for their contributions to the draft.

Author's Address

Markus Stenberg
Helsinki 00930
Finland

Email: markus.stenberg@iki.fi