Home Networking
Internet-Draft
Intended status: Informational
Expires: January 3, 2016

K. Jin
Ecole Polytechnique / Cisco
P. Pfister
Cisco
J. Yi
LIX, Ecole Polytechnique

July 2, 2015

Experience with the Distributed Node Consensus Protocol (DNCP)
draft-jin-homenet-dncp-experience-00

Abstract

This document reports experience with Distributed Node Consensus Protocol (DNCP). It includes an introduction of existed known implementations and simulation results of DNCP.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

- DNCP is a protocol used for state synchronization. It is described in [draft-ietf-homenet-dncp-03], and right now its use case is mostly in homenet. DNCP provides a way for each node in the network to publish a set of TLV tuples, which is the data that it wants to share with other nodes, and valide the data received from other nodes by making sure the source node of the data is reachable. It takes advantage of trickle algorithm to control the sending of the status updates, thus decrease the amount of traffic especially when there are no updates in the network.

- Since DNCP is a protocol not yet standardized, it has not been widely deployed, but in order for the homenet to run on it, it is important to evaluate its performances under various scenarios which is not so easy to do in real life but relatively easy using network simulators such as ns3. With the help of ns3, we can create various topologies and get logs for analyzing. This draft documents our experience of implementing dncp and integrating it in ns3, as well as the results of performance evaluation. We believe that the results obtained from the simulation are helpful for the implementation of dncp and can be a useful reference for the potential users of dncp.

- The document is organized as follows: First we introduce the current implementation of dncp in Section 2. Then the draft describes simulation setup, including simulation environment, the metrics being evaluated and the topologies used for simulation. The third part documents the results of performance evaluations under different scenarios. And finally from all the above, we draw our conclusions.

2. Implementations

- The implementation that we use in the simulation is condected by

- This implementation is completely open sourse, ans is available at https://github.com/sbyx/hnetd/tree/libdncp

- if available, number of lines/foot print

- if available, operational experience.

3. Simulation Setup

3.1. Simulation Environment

The current dncp implementation relies largely on linux library (for opening sockets, sending and receiving packets..etc) and uses libubox for scheduling events. To integrate dncp into ns3, we have to redefine all the functions in the code that are related to these two parts so that packets can be sent and received in ns3 and events can be scheduled using ns3 scheduler.

We used CSMA model in ns3 to simulate layer one and layer two. CSMA model is designed in the spirit of Ethernet but different from the real-life Ethernet in the sense that the CSMA channel can provide instantaneous carrier sense and priority-based collision avoidance. The channel has three states: TRANSMITTING, PROPAGATING and IDLE, the states can be seen immediately by the devices attached to the channel so collision never happens. CSMA model consists of two parts: CSMA channel and CSMA device. CSMA channel is the model of the transmission medium, and CSMA device is like the an Ethernet device, the CSMA devices are connected to the channel.

Listed below are several attributes of the CSMA device that we can configure:

o MTU:The mac level maximum transmission unit, set to 1500

o Encapsulation Mode: Type of link layer encapsulation to use. In our simulation we use the default mode "Dix" which is commonly used in Ethernet.

o TxQueue: Type of the transmit queue used by the device. In ns3, we have the possibility to choose from Codel queue, drop tail queue and RED (random early detection) queue. Here we use the drop tail queue and set the buffer of the queue to 100 packets. (bytes can also be used as the maximum queue size metrics)

o Interframe gap: The pause between two frames

And the attributes of the CSMA channel that we can configure:

o Data rate: The transmission data rate to be provided to the devices connected to the channel. That is the rate of the device pushing data into the channel. This attribute applies to all the devices on the same channel. In the simulation we set it to 1000Mbps thus providing an infinite throughput to eliminate the impact of throughput on the performance of dncp, in order to calculate the actual throughput consumed.

- o Delay: The speed-of-light propagation delay over the medium. Imagine there is a symmetrical hub that is of equal cable length to all the devices of the channel. When one device sends a packet to another device, the packet fist reaches the hub and is forwarded to the destination device, so the propagation delay is always the same for a given channel. In our simulation, this delay is set to 1 micro second.

3.2. Performance metric

- o Convergence time: The time that dncp takes for the network to converge. We use a concept of converging percentage to represent the converging state, basically the converging percentage is the proportion of the biggest cluster of nodes that share the same network hash. Apparently when this percentage is 100%, the network hqs coverged.

- o Traffic consumption: The amount of traffic that dncp uses to converge. To evaluate the traffic consumption we count the overall amount of bytes sent during the converging process as well as the throughput per second.

3.3. Chosen toplogies

3.3.1. Link topology

All the nodes are on the same link, share the medium:

```
n1      n2      n3      n4
|       |       |        |
---------------------------
```

A simple example of link topology

Figure 1

3.3.2. String topology

Nodes connected by point-to-point link in a line.

```
         ---------       ---------
         |       |       |       |
   n1    n2      n3      n4      n5      n6
   |     |       |       |       |       |
   ---------       ---------       ---------
```

A simple example of String topology

Figure 2

3.3.3. Mesh topology

A fully connected mesh.

3.3.4. Tree topology

A binary tree

3.3.5. Double Tree topology

Add some redundancy on the top of binary tree topology

4. Performance Evaluation

4.1. Scenario 1: Link topology of different size

- Convergence time

The average value is calculated over 10 experiments

10 nodes	20 nodes	30 nodes	40 nodes	50 nodes	60 nodes	70 nodes	80 nodes
1.84s	3.09s	*4.43s	5.14s	6.53s	*8.61s	11.57s	14.05s

*: the average value is calculated over the results of 9 experiments because the other one diverges too much

Table 1: the average convergence time of link topology

Note that we observed two accidents during the simulation. One happens in one experiment among the 10 that we ran for 30-node network, the network first converges at 4.016s, which is very close to the averge convergence time, but at 25.949s this converging state is broken and the network finally reconverges at 26.12s. The other one happens in the case of 60-node network where it fist converges at 7.081s then gets disturbed at 25.822s and comes back to converging at 26.303. As for the the reason of this happening, we will dig deeper into the logs and hope to find an explaination soon.

o Verbosity

The first row shows the overall bytes sent in the converging process, the second shows the bytes sent per node, calculated over 10 experiments

10 nodes	20 nodes	30 nodes	40 nodes	50 nodes	60 nodes	70 nodes	80 nodes
85.3KB	604.7KB	2.3MB	5.4MB	11.9MB	23.7MB	51.7MB	88.1MB
---	----	----	----	----	----	----	----
8.5KB	30.2KB	79.6KB	140.7KB	245.KB	404.8KB	757.2KB	1.1MB

Table 2: the traffic of dncp in link topology

4.2. Scenario 2: String topology of different size

o Convergence time

The average value is calculated over 10 experiments

10 nodes	20 nodes	30 nodes	40 nodes	50 nodes	60 nodes	70 nodes	80 nodes
1.84s	3.65s	5.24s	7.09s	8.79s	11.11s	12.87s	15.03s

Table 3: the average convergence time of string topology

If we plot the average convering time against the number of nodes, it is discernible that the graph is leaner. This result is exactly the same as we expected.

o Verbosity

The first row shows the overall bytes sent in the converging process, the second shows the bytes sent per node, calculated over 10 experiments

10 nodes	20 nodes	30 nodes	40 nodes	50 nodes	60 nodes	70 nodes	80 nodes
51.5KB	243.44 KB	605KB	1.2MB	2MB	3MB	4.1MB	5.6MB
---	----	----	----	----	----	----	----
5.1KB	12.2KB	20.1KB	30.9KB	40.5KB	50.4KB	59.2KB	70.1KB

Table 4: the traffic of dncp in string topology

o Convergence time

4.3. Scenario 3: Mesh topology of different size

The average value is calculated over 10 experiments

10 nodes	20 nodes	30 nodes	40 nodes	50 nodes	60 nodes	70 nodes	80 nodes
1.71 s	3.2s	4.83s	*6.19s	10.64s	13.02s	15.33s	17.93s

*: the average value is calculated over the results of 9 experiments because the other one diverges too much

Table 5: the average convergence time of mesh topology

o Verbosity

The first row shows the overall bytes sent in the converging process, the second shows the bytes sent per node, calculated over 10 experiments

10 nodes	20 nodes	30 nodes	40 nodes	50 nodes	60 nodes	70 nodes	80 nodes
202.7KB	1.6MB	6.6MB	18.1MB	49.1MB	95.8MB	167.4MB	271.9MB
---	----	----	----	----	----	----	----
20.3KB	83.5KB	222.1KB	453.8KB	983.1KB	1.6MB	2.4MB	3.4MB

Table 6: the traffic of dncp in mesh topology

o Convergence time

4.4. Scenario 4: Tree topology of different size

The average value is calculated over 10 experiments

10 nodes	20 nodes	30 nodes	40 nodes	50 nodes	60 nodes	70 nodes	80 nodes
1.16s	1.57s	1.86s	2s	2.33s	2.42s	2.56s	2.6s

Table 7: the average convergence time of tree topology

o Verbosity

The first row shows the overall bytes sent in the converging process, the second shows the bytes sent per node, calculated over 10 experiments

10 nodes	20 nodes	30 nodes	40 nodes	50 nodes	60 nodes	70 nodes	80 nodes
40.7KB	166.7KB	374KB	644.5KB	1MB	1.3MB	1.9MB	2.4MB
---	----	----	----	----	----	----	----
4.1KB	8.3KB	12.4KB	16.1KB	20.2KB	22.8KB	26.7KB	29.9KB

Table 8: the traffic of dncp in tree topology

4.5. Scenario 5: Tree topology of different size

- Convergence time

The average value is calculated over 10 experiments

10 nodes	20 nodes	30 nodes	40 nodes	50 nodes	60 nodes	70 nodes	80 nodes
1.04s	1.44s	1.5s	1.7s	1.96s	1.98s	2.06s	2.09s

Table 9: the average convergence time of double tree topology

- Verbosity

The first row shows the overall bytes sent in the converging process, the second shows the bytes sent per node, calculated over 10 experiments

10 nodes	20 nodes	30 nodes	40 nodes	50 nodes	60 nodes	70 nodes	80 nodes
66.9KB	265KB	605.1KB	1MB	1.5MB	2MB	2.8MB	3.5MB
---	----	----	----	----	----	----	----
6.7KB	13.2KB	20.2KB	25.3KB	30.8KB	33.2KB	39.7KB	44.7KB

Table 10: the traffic of dncp in tree topology

5. Conclusion

conclusions

Authors' Addresses

Kaiwen Jin
Ecole Polytechnique / Cisco
France

Phone:
Email:
URI:

Pierre Pfister
Cisco
France

Phone:
Email:
URI:

Jiazi Yi
LIX, Ecole Polytechnique
France

Phone:
Email:
URI: