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

July 1, 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

TODO list, for each known implementations (I think we have only one at this point?)

- conducted by who?

- open/close source? if open source, the link?

- 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 uloop 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 comsumed.

- Delay: The spped-of-light propagation delay over the medium. Imagine there is a symmetrical hub that is of equal cable lenghth 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

- Convergence time: The time that dncp takes for the network to converge.

- 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.

3.3.2. String topology

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

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

The average value is calculated over 10 experiments

10 nodes	20 nodes	30 nodes	40 nodes	50 nodes	60 nodes	70 nodes	80 nodes
1.843 s	3.092 s	*4.432 s	5.143 s	6.535 s	*8.61 s	11.57 s	14.05 s

*: 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

4.2. Scenario 2: String 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
1.84 s	3.653 s	5.244 s	7.099 s	8.786 s	11.111 s	12.869 s	15.032 s

Table 2: the average convergence time of string topology

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	3.205 s	4.833 s	*6.19 s	10.638 s	13.018 s	15.334 s	17.931 s

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

Table 3: the average convergence time of mesh topology

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
1.165	1.574 s	1.861 s	1.997 s	2.33 s	2.42 s	2.561 s	2.597 s

Table 4: the average convergence time of tree topology

4.5. Scenario 5: 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
1.044 s	1.436 s	1.5 s	1.704 s	1.96 s	1.98 s	2.06 s	2.086 s

Table 5: the average convergence time of double 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: