

Home Networking  
Internet-Draft  
Intended status: Informational  
Expires: January 2, 2016

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

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 January 2, 2016.

## Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as

described in the Simplified BSD License.

## Table of Contents

1.	Introduction . . . . .	3
2.	Implementations . . . . .	3
3.	Simulation Setup . . . . .	4
3.1.	Simulation Environment . . . . .	4
3.2.	Performance metric . . . . .	5
3.3.	Chosen topologies . . . . .	5
3.3.1.	Link topology . . . . .	5
3.3.2.	String topology . . . . .	5
3.3.3.	Mesh topology . . . . .	5
3.3.4.	Tree topology . . . . .	5
3.3.5.	Double Tree topology . . . . .	5
4.	Performance Evaluation . . . . .	5
4.1.	Scenario 1: xxx . . . . .	5
4.2.	Scenario 2: xxx . . . . .	5
5.	Conclusion . . . . .	6
	Authors' Addresses . . . . .	6

## 1. Introduction

- o 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 validate 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.
- o 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.
- o 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?)

- o conducted by who?
- o open/close source? if open source, the link?
- o if available, number of lines/foot print
- o if available, operational experience.

### 3. Simulation Setup

#### 3.1. Simulation Environment

The current dn timer implementation relies largely on linux library (for opening sockets, sending and receiving packets..etc) and uses uloop for scheduling events. To integrate dn timer 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 dn timer, 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 first 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 dn timer takes for the network to converge.
- o Traffic consumption: The amount of traffic that dn timer 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 topologies

#### 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: xxx

### 4.2. Scenario 2: xxx

## 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:

