The Net2 Protocol Benchmark

TEP:	140
Group:	Network Working Group
Type:	Informational
Status:	Draft
TinyOS-Version:	> 2.1
Author:	Thanh Dang, Kaisen Lin, Chieh-Jan Mike Liang, and Omprakash Gnawali
Draft-Created:	12-Jul-2010
Draft-Version:	1.1
Draft-Modified:	2010-07-12
Draft-Discuss:	TinyOS Developer List <tinyos-devel at mail.millennium.berkeley.edu>

Note

This memo documents a part of TinyOS for the TinyOS Community, and requests discussion and suggestions for improvements. Distribution of this memo is unlimited. This memo is in full compliance with TEP 1.

Abstract

The memo describes the metrics, test scenarios, test applications, and analysis tools for testing collection and dissemination protocols in TinyOS 2.x. Our goal is to design benchmark suites for network protocols.

1. Introduction

TinyOS 2.x has a number of collection and dissemination protocols. Collection is a network-layer best-effort protocol that forms routes from all the nodes in a network to one or more collection roots or sinks[1]. CTP[2] and MultihopLQI are examples of collection protocols available in TinyOS 2.x. Dissemination is a network-layer reliable protocol that provides the eventual delivery of key-value pairs (items) to all the nodes in a network[3]. Drip, DIP, and DHV are examples of dissemination protocols available in the TinyOS 2.x distribution. Although a large number of mature protocols are available, the community lacks standard performance tests to evaluate these protocols.

In this document, we describe benchmark suites (i.e., test cases, scenarios, metrics, and tools) designed with the following two goals. First, benchmark suites should allow us to systematically compare the performance of protocols of the same category. Second, benchmark suites should enable us to quantify the performance difference from the different versions of a protocol. We hope the community will adopt these benchmark suites so that the evaluation results from different research projects can be more meaningfully compared.

2. The Net2 Protocol Benchmark Overview

There are two components in the benchmark test cases. First, test cases need to specify the topologies, scenarios, and protocol parameters to vary across the experiments. Second, we need to define a set of metrics to interpret the results and logs collected from running test cases.

3. Definitions

Target state : target state of the network is the state where all nodes in the network achieve the collection or dissemination goals. For example, all data is reported to a sink in collection or data is disseminated to all nodes in dissemination.

Stable network: Stable network refers to network that is in target state for a reasonablly long time.

Updating time : For dissemination protocols, this is the duration from the time the first of a set of items being disseminated until the time all nodes receive allitems.

Collection time : For collection protocols, this is the duration from the time the first data is generated to the time all the data is reported to the sinks.

Stabilizing time : Stablizing time refers to the period from the time the network first reach the target state to the time the amount of control traffic generated by the network reaches a constant rate and is still in the target state. For dissemination protocols, stabilizing time is the duration from the last node receives the disseminated item until the network reaches a stable state. For collection protocols, this is the duration from any change in the network topology until the network reaches a target state. [COMMENT: change in terms of data generated by sensors or link status?]

Testbed topology : This refers to the connectivity graph of a testbed.

Network density : This defines the average number of neighborhood size in the network.

4. Test Case Topologies

To promote experiment repeatability and fair comparisons, the network topologies MUST remain the same when the same experiment is repeated for several runs. However, due to the dynamic nature of testbeds, it is difficult to repeat experiments under the same conditions on testbeds. We therefore focus on only specifying topologies for simulations in this document.

Topologies used in simulations can be either created by duplicating topologies of real-world testbeds or from synthesizing topologies as outlined later in this section. For testbed topologies, there are available topologies from Motelab, Mirage,...?, which can be accessible at urls..?. For synthesized topologies, experiments SHOULD use the following topologies.

a) Single-hop star topology

b) Single-hop clique topology

c) Multi-hop chain topology

d) Multi-hop grid topology

e) Multi-hop random topology

f) Multi-hop binary tree topology

```
                _____                         __ __
  \ | /          /\   /\                      |   |   |
__\|/__         /__\/__\        _ _ _ _       |__|__|                 \|/__\
  /|\           \   /\   /                    |   |   |          |\   /
 / | \           \/__\/                       |__|__|                  |_\/
   a)               b)             c)             d)                e)

    /\
   /\/\
  /\../\
 /\/\/\/\
    f)
```

The network size (total number of nodes) and the network density (average number of neighbors) can be varied but should remain the same for the same set of experiments.

The network dynamics, which are defined by the rate at which nodes (e.g., unreliable or mobile nodes) join and leave a network MUST also remain the same for the same set of experiments.

5. Test Case Scenarios

The following scenarios SHOULD be considered in evaluating dissemination protocols:

Scenario 1: A node disseminates one or more new items to a stable network. This scenario occurs in practice when a node or a base station reprograms a network or disseminates events to all nodes in the network.

Scenario 2: A node with old items joins an updated network. This scenario occurs in practice when one or more items have been disseminated to the network. However, there are new nodes (eg. mobile nodes moving into the network region, or intermittent nodes that were off during the dissemination) join the network.

Scenario 3: Multiple nodes join and leave a network at a specified rate. This scenario occurs in practice where mobile nodes moving in and out a region under dissemination.

Scenario 4: Multiple items with different versions are disseminated to the network from different nodes. This scenario occurs in practice where multiple users accessing a shared sensing network and commanding the network to perform different tasks at the same time.

The following scenarios SHOULD be considered in evaluating collection protocols:

Scenario 1: All nodes periodically generate one data packet in a network with only one sink. Nodes can start sampling at different time to mimic deployments where nodes boot at different time instances.

Scenario 2: All nodes periodically generate one data packet in a network with multiple sinks. This exercises the case where the network leverages multiple sinks to increase the overall network capacity.

Scenario 3: Multiple nodes join and leave a network at a specified rate.

Scenario 4: Links go down and come up at a specified order and rate.

6. Test Case Parameters

There are a number of parameters affecting the performance of collection and dissemination protocols. Protocol test cases MUST be described together with the parameters' values. The parameters are organized in the order of stack layers.

Layer	Parameter	Description	Value(s)
Application	Item size	Size of data item	2 or 10 bytes
	Number of items		2 to 128
	Number of new items	Number of items to be disseminated	2 to 128
Network layer	Suppression constant	Number of messages a node heard before suppressing its own transmission (trickle-based protocol only)	1, 2, 3
Link layer	Link quality	Can be gain, PRR ETX,...	
	Duty cycle or LPL	Can simply be duty cycle of the nodes or LPL parameters	

In addition, the following parameters that characterize a network MUST also be considered.

Parameters	Description	Value(s)
Density	Can be the average number of neighbors or the average gain	3,4,5,6 neighbors

Network size	Total number of nodes	
Network Dynamics	Rates at which nodes are joining and leaving a network (eg. due to mobility)	1%, 5%, 10%, 15%

7. Running the Test Cases

A complete benchmark requires considering all possible parameter settings for all network topologies. The total number of test cases is 11,340,000. To remove random factors, each test case SHOULD be repeated 10 times. Hence, the total number of experiments are 113,400,000.

Depending on which metrics are being compared, experimental design techniques SHOULD be used to reduce the number of experiments.

Each experiment should be ran sufficiently long for the interested comparison metrics to reach stability. For collection protocols, the stability could refer to observing stability in the data latency of the whole network.

While there are various methods of logging the node behavior during experiments, nodes SHOULD not broadcast any log packets on the radio until the experiment finishes. Instead, nodes can use the external flash or dump log data to the serial port, if available.

8. Comparison Metrics

The following metrics SHOULD be considered for dissemination protocol comparisons.

Total number of transmitted messages:

- For dissemination protocols, this defines the number of messages transmitted from the time new items are disseminated to the time the network is completely updated (or stable in the case of dynamic network topologies).
- For collection protocols, this defines the total number of messages and acknowledgements exchanged for the entire test case duration.

Data latency for a node:

- For dissemination protocols, this defines the latency calculated from the time new items are disseminated or the time the node is active (whichever later) to the time the node is updated.
- For collection protocols, this defines the latency of a payload calculated from the time it is generated on the node until it is successfully delivered to any sink.

Data latency for the whole network:

- For dissemination protocols, this defines the latency calculated from the time new items are disseminated to the time the network is completely updated (or stable in dynamic network case).

- For collection protocols, this is calculated by averaging the data latency for each node in the network.

Fraction of the network that is updated:

- This applies to dissemination protocols only. In the dynamic network case, it is impossible to achieve 100% of the network updated because there are always new nodes joining the network and nodes leaving the network. It is suitable to consider the fraction of the network (number of updated nodes/total number of nodes) that is updated. This fraction is measured when it is stable.

ROM and RAM usage:

- The memory usage of a simple program providing only the dissemination or collection service.

9. Result Analysis Tools

Analysis scripts (eg. python, shell, matlab) are to be developed and released (in tinyos-2.x/benchmark/tools?)

10. Authors

Thanh Dang
135 FAB, 1900 SW 4th Ave
Portland State University
Portland, OR 97201

email - dangtx@pdx.edu

Kaisen Lin
UCSD
email - kaisenl@cs.ucsd.edu

Chieh-Jan Mike Liang
213 NEB, 3400 N Charles St
Johns Hopkins University
Baltimore, MD 21211

email - cliang4@cs.jhu.edu

Omprakash Gnawali
S255 Clark Center, 318 Campus Drive
Stanford University
Stanford, CA 94305

phone - +1 650 725 6086

email - gnawali@cs.stanford.edu

11. Citations

[1] TEP 119: Collection

[2] TEP 123: The Collection Tree Protocol (CTP)

[3] TEP 118: Dissemination of Small Values