Sensor Networks and Mobile Data Communications Coursework
CS912
2018-2019

January 27, 2019

Trickle

Trickle is an algorithm used for propagating and maintaining code updates in wireless sensor networks (WSNs). Code updates are important in WSNs for several reasons: (i) the requirements of the applications have changed and parameters need to be updated for optimal operation or (ii) a bug has been discovered which needs to be patched. However, energy is a primary driver in the design of WSNs protocols. Code propagation is an expensive process: (i) the size of the code may be large, requiring several messages, (ii) *all* nodes need to receive the new code and (iii) there may be many redundant messages on the network, and (iv) learning when to propagate the update is important.

Trickle uses techniques from the epidemic, scalable multicast and wireless broadcast research areas. To regulate itself, Trickle uses a technique, called "polite gossip" to exchange code metadata. Trickle has to balance between propagation speed and maintenance cost, and achieves that through several parameters. The parameters are: (i) a counter c, (ii) a redundancy constant k, and (iii) a gossiping interval τ. We call the tuple (c, k, τ) a *Trickle tuple* and a given value assignment to the Trickle tuple is called a *Trickle configuration. Trickle exists as a standard library in the Contiki Operating System.*

You will need to read the paper about Trickle, which is available for download from the module webpage.

Coursework

You are required to run a number of simulations to understand the relationship between these Trickle parameters. You will generate a network of 64 nodes and you will need to vary either the distance between the nodes or the transmit power to generate *two* different network topologies, namely a dense network and sparse network. You are allowed to use the existing implementation of Trickle in Contiki.

You will write a sensor network application in Cooja, a wireless sensor network simulator for Contiki. The application will drive the use of Trickle. For each experiment, you will choose a certain number of nodes, ranging from 1 to 5. We call these nodes *proposers*. Each proposer will propose a value in the range $1 \ldots 100$. Each proposer will then disseminate its proposed value using Trickle. At the end of the dissemination process, every node will evaluate the average of the values that have been proposed. So, for example, if there are 4 proposers $P_1 \ldots P_4$, and P_1 proposes 50, P_2 proposes 60, P_3 proposes 40 and P_4 proposes 70, then every node needs to return 55, which is the average of the 4 proposed values. You may need to have a few runs to determine the time taken for the execution to complete and use this as an approximation of the time it will take for the dissemination to complete.

You will produce the following set of plots for the above program:

1. Node id (X-axis) v/s computed average value (Y-axis), for a given Trickle configuration and a given number of proposers. You are to produce the plot for 4 different Trickle configurations for a single topology of your choice.

2. Network diameter v/s Dissemination latency, for a given Trickle configuration and a given number of proposers. You are to produce the plot for 4 different Trickle configurations and for both topologies.

3. Number of proposers v/s number of messages, for a given Trickle configuration. You are to produce the plot for 4 different Trickle configurations. You will also need to show for both topologies

4. Number of proposers v/s dissemination latency, for a given Trickle configuration. You are to produce the plot for 4 different Trickle configurations. You will also need to show for both topologies

The following are the definitions for the above metrics:

1. Network diameter: Maximum path length between any pair of nodes in the network.

2. Dissemination latency: Miminum time taken for *all* the nodes to compute the *correct* average.

3. Number of messages: The total number of message transmissions in the network, by all nodes.

4. Number of messages per node: Number of messages/ number of nodes.

Deliverables

You will submit the following via *Tabula*, as a zip file, by *Monday 4th March at 12pm*:

1. A pdf report, of *no more than 6* pages. You will explain the working of Trickle in your own words (around half a page). You will explain your experimental setup in detail, e.g., what Trickle configurations have been used, and discuss the results/plots that you obtain. You will have a short *Discussion* section in your report where you discuss whether the results you have obtained match those expected of Trickle.

2. A readme file, to explain how to run your experiments.

3. Any piece of code that you write, e.g., scripts, protocol changes etc.

Mark Breakdown

The coursework is worth 40% of the module mark. The mark allocation for each component of your submission will be as follows:

Table 1: Mark breakdown

Component	% Mark
Trickle explanation	10%
Experimental setup	20%
Each set of plots * 4	10% * 4 = 40%
Discussion	5%
Application code	20%
Code compilation & execution	5%
Total	100%

Important Information

The nature of this project is such that it will **not** be possible for a student to finish this in one week, as running wireless sensor networks simulations takes time. It is thus advised that students start the project early and plan ahead to ensure the project is successfully completed.