Computer Science and Engineering,
University of Nevada, Reno

CPE 400 Technical Report

Topic #2:
Dynamic routing mechanism design with focus on energy conservation.

Team:
Benjamin Estela, Sidney Inouye, Jonathon Hewitt

Instructor:
Shamik Sengupta, PhD

Fall 2019 Semester
December 4, 2019

Abstract

In computer communication networks, there may be a sensor node network that is energy constrained and may not be as powerful as a standard desktop computer. Over time, nodes in a sensor node network will lose energy. They will shut off if they are out of energy such that packets cannot flow through them anymore. In order to maintain efficiency as packets need to flow through the network, a dynamic routing mechanism design must be created to maximize energy conservation. Using the Routing Information Protocol (RIP), a dynamic routing mechanism can be created.

Introduction

Wireless sensor networks may contain a sensor node network with a limited amount of energy. In this paper, the problem of sensor node network having a limited amount of energy will be discussed where a dynamic routing scheme will be needed to figure out how to both send packets with respect to time efficiency as well as energy conservation. When a sensor is out of energy, no packets will be able to be forwarded through that node. An appropriate solution to tackle this problem can be found through RIP, a dynamic routing protocol that was included in the BSD-UNIX distribution in 1982 as a distance vector algorithm.

This protocol, in the context of this paper, uses both standard RIP and a modified version of RIP using Breadth First Search (BFS) augmented with the standard Bellman-Ford algorithm to allow the program to have dynamic routing. With this, the novel contribution presented shows how the modified RIP algorithm can maximize energy efficiency while minimizing possible errors in runtime. The results afterwards show a sample graph on a test run as well as the average packets sent using the presented algorithms. The results also show that the modified RIP algorithm is more reliable, getting results consistently.

The Problem: Sources with Limited Energy

The network that is created is a series of twenty sensors. Each sensor has a limited amount of energy. Each sensor is connected through a link that has a weighted cost associated with it. Every time a packet is sent, the node that is sending the packet decreases its energy by the cost of sending the packet. The goal of the project is to keep the network online for as long as possible. This needs to be done with a dynamic routing protocol, which is able to keep online, even when some sensors are offline when energy is low. The problem is shown in Figure 1.

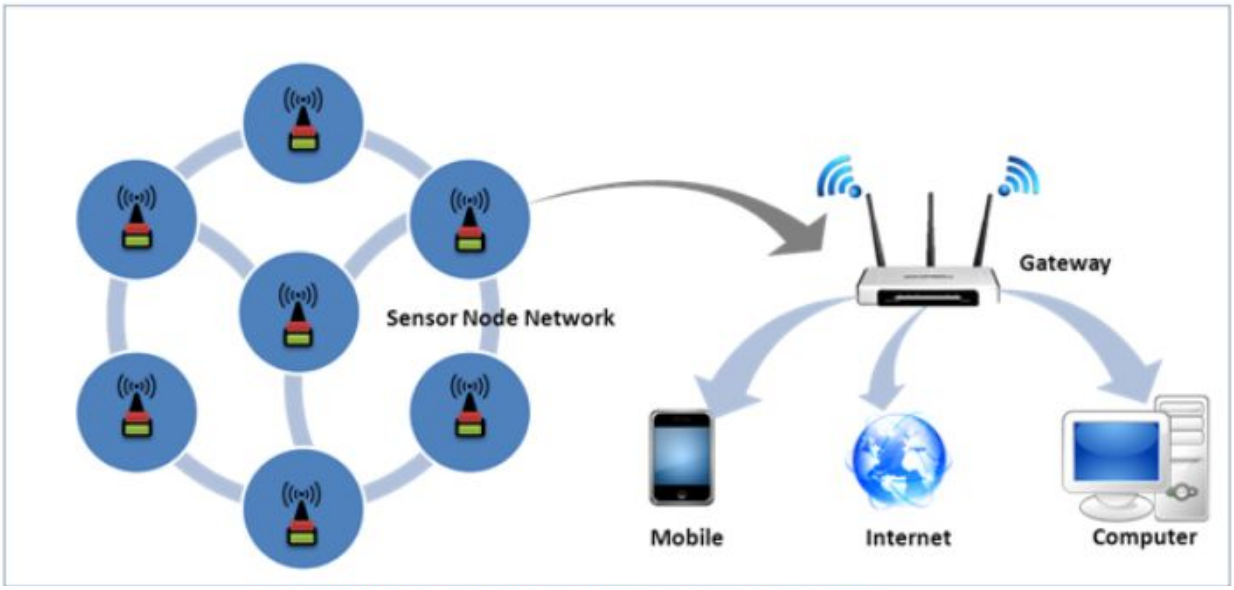

Fig. 1: The network is a series of sensors that are linked together with random costs.

The team has designed the network of sensors to be as random as possible. Every time the program runs, the twenty nodes will be connected to each other completely randomly, creating a different graph of sensors every time. Every link has a different cost that is randomly generated between one and five. The limited energy each sensor has is randomly generated between 100 and 199 units. The source and destination sensors are also randomly generated.

The Solution

In order to keep the network online for as long as possible, a shortest weighted path must be calculated. This will make sure that every node that forwards a packet is conserving as much energy as possible. RIP is implemented as a way of dynamically finding the shortest path. Additionally, the team changed the protocol to be augmented with the BFS search algorithm, which allows the network routing protocol to be more reliable, though it does exchange reliability for efficiency.

Routing Information Protocol History

RIP, distributed in the Berkeley Software Distribution (BSD) UNIX in 1982, has its roots back in 1969 basing itself off of the Bellman-Ford and Ford-Fulkerson algorithms. Distance-vector routing developed over the years prior with Xerox's Gateway Information Protocol which paved the way for early routing protocols until 1982. According to Network Encyclopedia, "there are two versions of RIP." The first is RIP version 1 with the following cleverly titled RIP version 2. This second version expands

on the original by allowing routing tables to be multicasted as well as password protection for some basic security.

Routing Information Protocol Implementation

RIP is a distance vector algorithm. This means that the costs to get to the neighbors are known, but the cost to get to every single node is not known. The protocol normally works by having a max number of hops. The max number of hops is fifteen. Information about nodes, such as the closest routes, are sent to neighboring nodes as advertisements every 30 seconds. When an advertisement isn't sent within 180 seconds, the link is assumed to be dead and closest paths are recalculated and once again advertised to the neighboring nodes. This protocol was chosen due to its ability to account for dead nodes and adjust accordingly.

RIP has a few disadvantages. RIP calculates the closest path, but not the shortest path. It also is limited by a max number of hops, meaning that a solution may not always be in reach. The team solved this by combining the Bellman-Ford shortest path algorithm to find the shortest possible path and the BFS algorithm to guarantee that a solution will be reached without being limited by the max number of hops. BFS also guarantees that no loops will occur, as the way the algorithm works is by only visiting nodes once.

RIP was chosen as it only needs to know information about neighboring nodes. RIP updates information every time a node goes offline, which works well with a program that has nodes with limited energy. Every time a sensor goes offline, the algorithm can just recalculate the shortest paths. RIP guarantees that no infinite loops will occur in the program. Although normally the loops are stopped by the max number of hops in the original protocol, the team recreated this functionality using BFS, which only visits and expands nodes once.

Realistic Aspects

Several error handling exceptions may be thrown during the lifetime of the simulation. There are three exceptions that are not likely to occur. These exceptions are in the case that information is being accessed that are outside of the bounds of the adjacency matrix or the energy array. These are unlikely to occur due to the way all the randomly generated values in the problem are created as well as the fact that the user never calls the functions directly when using the program. These functions are usually called within for loops, where the information being called can never be outside of the bounds of either the adjacency matrix or the energy array.

The two most important exceptions that occur, and have occurred consistently every simulation are the exceptions that occur during the protocol call and the travel algorithm. The first exception states that the cost of travelling to the next node is greater than the energy that the node has. This means that the packet cannot be sent to the next node in the path, therefore resulting in a network failure. This error is caused due to the nodes each containing a limited amount of energy as stated in the problem. If this energy is decreased enough, the node will not be able to forward any packets. Once this error occurs, the simulation stops running.

The second exception states that there isn't a path that exists between the source (A) and the destination (B). This exception can occur twice. When the adjacency matrix is initialized, it sets all the weights between nodes to -1, which means that there doesn't exist a path between them. The first time the error can be called is if the travel algorithm notices that the path between A and B is equal to -1, meaning there is no path that exists between A and B. This may occur if a dead node was not updated by the time the RIP function was called, so the function returned a shortest path using an already dead node.

The second time the error can be called is during the RIP function call. This error is called when the destination never gets a shortest path. This is checked by seeing if the distance array that finds the shortest distance during the Bellman-Ford algorithm is never updated for the destination node. If this happens, that means that every node that connects the source to the destination originally is dead. This is especially prevalent if two islands are created when a node dies. When this exception is called, in either situation, the program immediately stops simulation and prints out the error that no path was found.

Novel Contribution

The RIP algorithm finds the shortest route within fifteen hops. The route RIP finds is not guaranteed to be the shortest possible path as the algorithm is designed to find the closest path; however, in most cases the path it finds is short enough. In the case of trying to minimize energy consumption on sensors to increase the lifetime of a network, it is important to find the guaranteed shortest path to minimize energy usage. It is also important to guarantee a path from the source to its destination. This is done by augmenting RIP with BFS. With this change, RIP will always find the shortest path which drastically increases the lifetime of the network. The modified algorithm also guarantees a solution will be found.

The BFS algorithm is how the path searches through all possible paths to find the shortest one. The algorithm begins by storing the source node in a queue. While this queue isn't empty, the program traverses the graph by searching through all the paths that are neighbors with the front of the queue. This allows the graph to expand the closest paths first. The algorithm also uses a boolean array that holds information about which nodes have been visited already and which nodes have yet to be visited. This allows the program to prevent infinite loops from occurring.

The BFS algorithm becomes the base for the Bellman-Ford shortest path algorithm. The Bellman-Ford algorithm creates an array of distances, which are all set to infinity. As the program loops through the BFS algorithm, the distances array stores the smallest distance from a source node to the node being expanded. The RIP algorithm the team creates stores the shortest paths to get from the source node to all the nodes, including the destination node.

Although modifying RIP this way may cause errors within cycles due to the maximum amount of hops not being capped, the BFS algorithm accounts for this by including a boolean to check whether nodes have been visited or not. The issue with this modified protocol is that RIP is a slow protocol to begin with, and augmenting the slower protocol with BFS only makes sending packets slower. Although this difference cannot be seen well with the size of the experiment that was conducted, if more sensors were to be added, the protocol may suffer from slow transmissions of packets. In such a case, the modified protocol will not be the most efficient solution, but rather the unmodified RIP should be implemented.

Results

Graphs were randomly generated along with initial energy arrays. After the graphs and energy arrays were generated, random source and destination nodes are selected. The shortest path is found and travelled. This is repeated until there is a failure in delivering a packet, and the number of packets delivered is returned. An example output from one test using a graph with five nodes and initial energies ranging from 20 to 29 is shown in Figure 2. Table 1 shows each step in the simulation, which successfully ends after sending 22 packets. The simulation to get these results uses the modified RIP algorithm, using BFS. The actual tests runs on graphs with 20 nodes and initial energies ranging from 100 to 199. The final program runs the same way, with 20 nodes and initial energies ranging from 100 to 199.

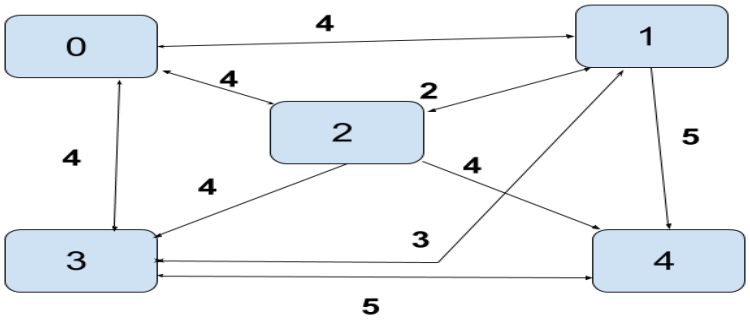

Fig. 2: The graph of the example test shows the sensors and the weighted connections that link each node together.

Table 1: The table shows 22 packets being sent during a single simulation with the graph shown in Figure 2.

Packet	Source	Destination	Route taken	Energy array
1	2	0	2->0	[20, 27, 25, 25, 29]
2	0	3	0->3	[16, 27, 25, 25, 29]
3	2	3	2->3	[16, 27, 21, 25, 29]
4	1	3	1->3	[16, 24, 21, 25, 29]
5	0	2	0->2	[12, 24, 21, 25, 29]
6	4	0	4->1->0	[12, 20, 21, 25, 24]
7	1	4	1->4	[12, 15, 21, 25, 24]
8	0	4	0->1->4	[8, 10, 21, 25, 24]
9	4	1	4->1	[8, 10, 21, 25, 19]
10	3	2	3->2	[8, 10, 21, 21, 19]
11	2	4	2->4	[8, 10, 17, 21, 19]
12	0	1	0->1	[4, 10, 17, 21, 19]
13	3	2	3->2	[4, 10, 17, 17, 19]
14	0	2	0->2	[0, 10, 17, 17, 19]
15	2	0	2->0	[0, 10, 13, 17, 19]
16	1	0	1->0	[0, 6, 13, 17, 19]
17	4	3	4->3	[0, 6, 13, 17, 14]
18	2	4	2->4	[0, 6, 9, 17, 14]
19	4	0	4->1->0	[0, 2, 9, 17, 9]
20	3	1	3->1	[0, 2, 9, 14, 9]
21	4	2	4->2	[0, 2, 9, 14, 5]
22	2	0	2->0	[0, 2, 5, 14, 5]

After running the algorithm on a million randomly generated graphs, the average number of packets sent using the BFS augmented RIP was 65.893. The average number of packets sent using the original RIP algorithm was 1.87. Most trials with RIP resulted in zero packets being sent. Although the BFS algorithm runs slower, it is able to get the shortest path between nodes, where RIP cannot due to its limitation of a maximum of fifteen hops. This indicates a massive improvement in the reliance of the network using the modified RIP with the BFS algorithm.

Conclusion

In conclusion, the team approached the problem of maximizing the packets sent and energy efficiency in a sensor node network with limited energy with two key solutions. The first is a normal implementation of the RIP using a maximum of fifteen hops. The second is a modified implementation of RIP that uses BFS as the base of the protocol. The modified implementation continuously gives reliable results due to removing RIP's reliance on a maximum number of hops. RIP is shown to fail to find a path during tests with both algorithms. The modified RIP may decrease the routing protocols efficiency due to BFS slowing down the protocol as it searches every single node, however it is a more reliable approach to finding the shortest path, necessary for the energy efficiency needed to solve the problem.

References

https://networkencyclopedia.com/routing-information-protocol-rip/
The website, Network Encyclopedia, stores the histories and definitions of various protocols, terms, interfaces, and more. The posts on the website are detailed on its implementation as well as the history of computer networking as a whole, dating back to the 1960s. Network related products such as hardware are also discussed in reviews.