

Analisis Performa *Routing* SPIN (*Sensor Protocol for Information Via Negotiation*) pada *Wireless Sensor Network*

Salsabila¹, Rakhmadhany Primananda², Edita Rosana Widasari³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹salsa2496@gmail.com, ²rakhmadhany@ub.ac.id, ³editarosanaw@ub.ac.id

Abstrak

Wireless Sensor Network adalah suatu jaringan yang terdiri dari beberapa *node* yang memiliki sifat dinamis. Teknologi ini dapat digunakan untuk menyelesaikan masalah yang ada pada *routing*. Contohnya adalah *routing* SPIN. Pada penelitian ini penulis mencoba menganalisis *routing* SPIN-BC dan SPIN-RL, yang mana *routing* SPIN-BC dan SPIN-RL dapat menanggulangi antrian lalu lintas pada pengiriman data dan dapat berkomunikasi lebih dari satu *node* (*broadcast*). Penelitian ini bertujuan untuk mengetahui performa *routing* SPIN-BC dan SPIN-RL. Untuk mengetahui performa dari kedua *routing* dilakukan pengujian dengan beberapa parameter. Parameter tersebut adalah rata-rata *Latency*, *Number data pakets forwarded*, *Tx Power*, *Number Tx Frames* dan *Memori Node*. Pada pengujian yang telah dilakukan *routing* SPIN-BC dan SPIN-RL dengan *node* sebanyak 4,8,12,16, dan 20 memiliki nilai rata-rata *latency* 0.01856 ms, sedangkan pada *routing* SPIN-RL nilai rata-rata nya 0.83633 ms. Pada pengujian *NB Packet Data Forward* pada *routing* SPIN-BC rata-rata yang didapat 1.605 ms dan SPIN-RL rata-ratanya 7.88917 ms. Untuk nilai rata-rata yang diperoleh *TX Power* pada pengujian *routing* SPIN-BC dan SPIN-RL adalah 0,63292 mw. Selanjutnya hasil rata-rata yang diperoleh dari parameter *Nb Tx Frame* pada *routing* SPIN-BC adalah 1,7 ms dan pada SPIN-RL rata-ratanya 7,9 ms. Kemudian pengujian dari memori *node* pada SPIN-BC 7,56 MB sedangkan SPIN-RL 34,454 MB. Berdasarkan dari hasil rata-rata diatas dapat disimpulkan bahwa SPIN-RL memiliki performa yang lebih baik di bandingkan dengan SPIN-BC, karena SPIN-RL dapat menanggulangi kesalahan transmisi perintah yang disebabkan oleh *data loss*.

Kata kunci: *Routing*, *wireless sensor network*, *SPIN-BC*, *SPIN-RL*.

Abstract

Wireless Sensor Network is a network consisting of several nodes that have dynamic properties. This technology can be used to solve existing problems on routing. An example is SPIN routing. In this research the writer tries to analyze SPIN-BC and SPIN-RL routing, which SPIN-BC and SPIN-RL routing can overcome traffic queue on data delivery and can communicate more than one node (*broadcast*). The research aimed at determining the performance of SPIN-BC and SPIN-RL. routing anymore and the performance of both routing that were done by testing with some parameters. These parameters were Average Latency, Number Data Packets Forwarded, Tx Power, Number Tx.Frames and Memory Node. In the test that had been done by SPIN-BC and SPIN-RL routing with nodes of 4,8,12,16, and 20 had latency mean value of 0.01856 ms, while the SPIN-RL routing had an average value of 0.83633 ms. The average test of NB Packet Data Forward on SPIN-BC routing was obtained averaged of 1.605 ms and SPIN-RL was obtained average of 7.88917 ms. The average value that was obtained by TX Power on SPIN-BC and SPIN-RL routing testing was 0.63292 mw. Furthermore, the average result that was obtained from Nb Tx Frame parameter on SPIN-BC routing was 1.7 ms, while SPIN-RL was obtained average of 7.9 ms. Then the test of the average result from the memory node was on SPIN-BC 7,56 MB, and SPIN-RL 34,454 MB. Based on the average results above, it can be concluded that SPIN-RL had better performance compared with SPIN-BC, because SPIN-RL can cope with command transmission error that was caused by *data loss*.

Keywords: *Routing*, *wireless sensor network*, *SPIN-BC*, *SPIN-RL*

1. PENDAHULUAN

Teknologi di dunia sudah semakin berkembang dengan pesat. Salah satu teknologi yang akhir-akhir ini banyak digunakan adalah *Wireless Sensor Network*. *Wireless Sensor Network* merupakan suatu jaringan yang terdiri dari beberapa *node* yang bersifat dinamis. Teknologi tersebut dapat diaplikasikan dimanapun tanpa menggunakan jaringan infrastruktur yang telah ada. Kemudian dapat langsung melakukan komunikasi antara *node* satu dengan *node* yang lain karena *node* juga berfungsi sebagai *router*.

Wireless Sensor Network dapat digunakan untuk menyelesaikan masalah yang ada pada berbagai *routing*. Contohnya adalah *routing* SPIN yang memiliki masalah *Blindly Forward* dan *Data Unccessible*. *Blindly forward* merupakan suatu masalah yang terjadi ketika jaringan memiliki data baru yang hendak dikirim, sehingga harus mengulangi proses ADV, REQ, serta pengiriman data yang diulang sampai paket mencapai tujuan. Kemudian untuk *Data Unccessible* terjadi jika *node* sensor mengumpulkan data baru yang perlu diteruskan, maka akan langsung *broadcast* pesan ADV ke *node* tetangganya. Untuk mengatasi hal tersebut maka akan dilakukan modifikasi pada *routing* SPIN dengan menambahkan VC table. Hasil simulasi menunjukkan bahwa modifikasi *routing* SPIN memiliki perbaikan dari segi *packet loss* sebesar 10,39 % dari sebelumnya dikarenakan adanya perbaikan di sisi penentuan *next hop*-nya (Ratna Mayasari et al. 2013).

Pada penelitian sebelumnya yang berjudul *Performance Analysis of SPIN and LEACH Routing Protocol in WSN* yang mana pada penelitian ini melakukan analisis performa dengan menggunakan *routing* SPIN dan LEACH pada *Wireless Sensor Network*. Dalam pengimplementasiannya menggunakan dua simulator jaringan nirkabel versi 2.34 untuk pengujiannya (Gettu Sonia Juneja, 2012). Hasil yang didapat adalah pada algoritma LEACH membutuhkan energi yang lebih banyak dibandingkan dengan algoritma SPIN.

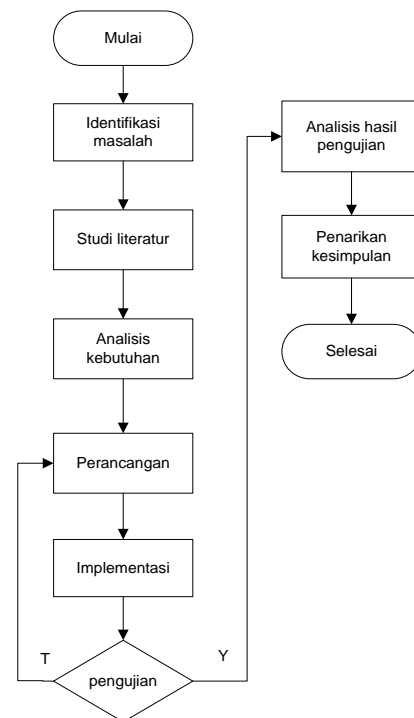
Dengan demikian penulis mencoba menganalisis lebih luas lagi dari beberapa macam *routing protocol* yang berupa *SPIN-BC* dan *SPIN-RL*. Untuk lebih mengetahui performa kinerja *routing* berdasarkan tingkat *Latency*, *Number data pakets forwarded*, *TxPower*, *NumberTx* .*Frames* dan *Memori*

Node. Selanjutnya untuk mengetahui *routing* SPIN mana yang baik dalam mengirim data pada jaringan *Wireless Sensor Network* (WSN) yang diimplementasikan pada OMNET ++.

Beberapa kelebihan pada *routing* SPIN-BC dan SPIN-RL yaitu *routing* SPIN-BC dan SPIN-RL menerima keuntungan dari kemampuan saluran penyiaran dan mengharuskan *node* yang telah menerima perintah ADV untuk tidak segera merespon dengan perintah REQ. Sebaliknya, *node* akan menunggu dalam waktu tertentu, selama kemampuan saluran penyiaran memonitor saluran komunikasi. Yang bertujuan untuk menanggulangi penumpukan data. Selanjutnya pada SPIN-BC dan SPIN-RL dapat mengirim data ke *node* yang lebih dari satu secara bersamaan (Sohraby et al. 2007).

2. METODE PENELITIAN

Metode penelitian ini menjelaskan tentang langkah-langkah yang dilakukan dalam perancangan, dan pengujian dari simulasi yang akan dijalankan dan dianalisa..

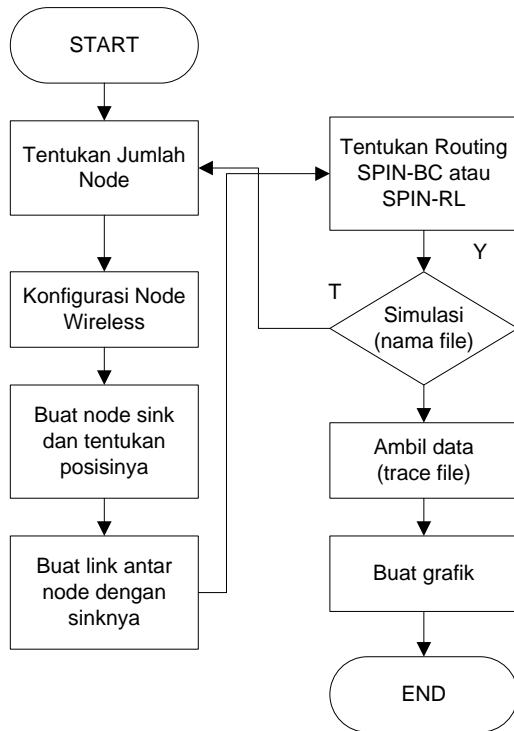


Gambar 1. Diagram Alir Metodologi

3. PERANCANGAN

Perancangan sistem didasarkan pada perencanaan area simulasi, jumlah *node* sensor, dan sistem komunikasi antar *node* dengan *sink*. Area simulasi ditetapkan sebagai luasan yang didefinisikan sebagai sumbu x dan sumbu y. Jumlah *node* yang telah ditentukan kemudian

diletakkan atau di posisikan dengan aturan yang telah ditentukan, dimana dalam perencanaan ini sensor diposisikan dengan aturan statik. Keseluruhan perencanaan parameter-parameter simulasi ini diinisialisasikan pada program yang dibuat. Adapun keseluruhan cara kerja dari simulasi ini ditampilkan dengan *flowchart* pada Gambar 2:



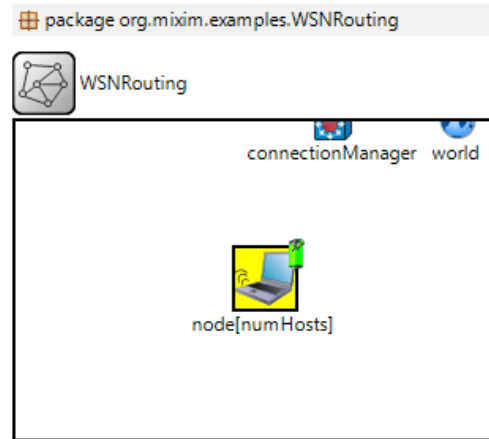
Gambar 2. Flowchart perancangan sistem SPIN-BC dan SPIN-RL

3.1 Implementasi Routing

3.1.1 Routing SPIN-BC

Pada topologi SPIN-BC dibawah ini peneliti menggunakan simulator OMNet++, dan perlu dibuat dahulu *template network* yang akan digunakan. *Template network* dibuat pada *file* dengan ekstensi (.ned), pada simulasi ini dibuat pada *WSNRouting.ned*. berikut adalah *template* yang digunakan.

Gambar 3 merupakan gambar dari *file WSNRouting.ned* yang telah dibuat. Gambar yang bertuliskan *node[numHosts]* apd agambar 3 merupakan gambar dari perumpamaan *node*. Sedangkan gambar yang bertuliskan *connection Manager* dan *world* pada gambar 3 adalah bagian dari simulasi jaringan.



Gambar 3. Template Network Omnet++

Setelah melakukan pembuatan *file .ned* tersebut kemudian dilakukan konfigurasi dengan *code* sebagai berikut:

```

package org.mixim.examples.WSNRouting;

import org.mixim.base.modules.BaseNetwork;
import org.mixim.modules.node.Host802154_2400MHz;

//
// This example shows how to simulate wireless sensor networks.
// Three different routing protocols are considered, for different traffic
// types:
// convergecast, with Wiseroute;
// network-level broadcast, with flooding;
// and probabilistic broadcast.
//
network WSNRouting extends BaseNetwork
{
    parameters:
        int numHosts; // total number of hosts in the network

    submodules:
        node[numHosts]: Host802154_2400MHz {
            parameters:
                numHosts = numHosts;
                @display("p=158,99;b=42,42,rect,yellow;i=device/wifilaptop");
        }
}
    
```

Gambar 4. Konfigurasi Template

Pada Gambar 4 merupakan *Template network WSN Routing* menggunakan konfigurasi yang digunakan juga pada *BaseNetwork*. Lalu ada parameter *numHosts* yang digunakan untuk mengatur jumlah *host* yang ada di dalam *network*.

Berikut adalah *code* yang digunakan untuk membuat topologi beserta simulasi SPIN pada *omnetpp.ini*.

```
[General]
network = WSNRouting
sim-time-limit = ${simDuration=1} min
tkenv-image-path = ../../images;
repeat = 5
```

Gambar 5. Konfigurasi General

Pada Gambar 5 ditunjukkan bahwa template *network* yang digunakan adalah *WSN Routing* yang sudah dibuat pada file *WSNRouting.ned*. Selain itu durasi simulasi maksimal diatur selama 1 menit.

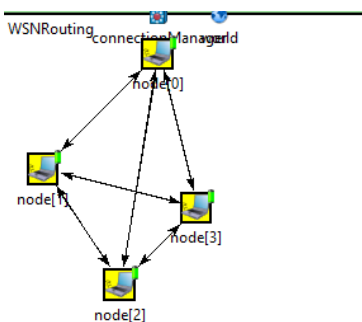
```
**coreDebug = false
**debug = false
**playgroundSizeX = 1000 m
**playgroundSizeY = 1000 m
**playgroundSizeZ = 0 m
**numHosts = ${numHosts=4,20}
```

Gambar 6. Konfigurasi Ukuran Area

Pada Gambar 6 ditunjukkan Ukuran area dari WSN diatur dengan ukuran panjang 1000 meter dan lebar 1000 meter. Lalu jumlah *node* ditentukan 4,8,12,16,dan 20.

3.1.1.1 Skenario Topologi Sebanyak 4 Node

Gambar 7 merupakan implementasi topologi setelah dijalankan. Pada Gambar 7 mencoba *node* dari yang paling sedikit, yaitu *node* yang berjumlah 4.



Gambar 7. Topologi Sebanyak 4 Node

```
**mobilityType = "StationaryMobility"
**node[*].mobility.initFromDisplayString = false
**node[*].mobility.initialZ = 0m
**node[0].mobility.initialX = 200m
**node[0].mobility.initialY = 50m
**node[1].mobility.initialX = 50m #+ 1*${dist=10,20,30,40,50}m
**node[1].mobility.initialY = 200m
**node[2].mobility.initialX = 150m #+ 2*${dist}m
**node[2].mobility.initialY = 350m
**node[3].mobility.initialX = 250m #+ 3*${dist}m
**node[3].mobility.initialY = 250m
**node[*].nic.mac.txPower = ${txPower=5.5,1} mW
```

Gambar 8. Konfigurasi Node sebanyak 4

Pada Gambar 8 merupakan konfigurasi Node sebanyak 4. Jenis topologi bersifat statis dan tidak berpindah lokasi. Tiap *node* ditempatkan dengan koordinat x dan y yang ditunjukkan pada *code* diatas, koordinat z yang tidak digunakan diset bernilai 0 pada semua *node*. Power yang digunakan *node* untuk mengirim data dibatasi pada range 5.5 sampai 1 mW.

Untuk penjelasan konfigurasi tiap node sebagai berikut:

1. ****mobilityType = "StationaryMobility"** → tipe yang digunakan "stationary"
2. ****node[*].mobility.initialZ=0m**
→ koordinat Z tidak digunakan diset bernilai 0
3. ****node[0].mobility.initialX=200m**
→ node 0 pada koordinat X diset bernilai 200m
4. ****node[0].mobility.initialY=50m**
→ node 0 pada koordinat Y diset bernilai 50m
5. ****node[1].mobility.initialX=50m**
→ node 1 pada koordinat X diset bernilai 50m
6. ****node[1].mobility.initialY=200m**
→ node 1 pada koordinat Y diset bernilai 200m
7. ****node[2].mobility.initialX=150m**
→ node 2 pada koordinat X diset bernilai 150m
8. ****node[2].mobility.initialY=350m**
→ node 2 pada koordinat Y diset bernilai 350m
9. ****node[3].mobility.initialX=250m**
→ node 3 pada koordinat X diset bernilai 250m
10. ****node[3].mobility.initialY=250m**
→ node 3 pada koordinat Y diset bernilai 250m
11. ****node[*].nic.mac.txPower=\${txPower=5.5,1}mW**, node pengiriman data dibatasi range 5.5 sampai 1mW.

3.1.2 Konfigurasi SPIN-BC

Berikut ini merupakan konfigurasi dari routing SPIN-BC..

```
[Config Spin-BC]
description="Wireless sensor network with the sink Spin-BC the network periodically."
**.node[*].networkType = "Flood"
**.node[*].netwl.stats = true
**.node[*].netwl.headerLength = 24 bit
**.node[0].appl.nbPackets = 10
**.node[*].appl.nbPackets = 0
**.node[*].appl.destAddr = -1
**.node[*].appl.broadcastPackets = true
**.node[*].appl.trafficType = "periodic"
**.node[*].appl.trafficParam = 30 s # each node sends 1 packet every 30 seconds
**.node[*].appl.initializationTime = 10 s
**.node[*].appl.headerLength = 50 byte

[Config probabilisticBcast]
description="Wireless sensor network using probabilistic broadcast."
extends=Spin-BC
**.node[*].networkType = "AdaptiveProbabilisticBroadcast"
```

Gambar 9. Konfigurasi SPIN-BC

Pada Gambar 9 konfigurasi simulasi menggunakan SPIN-BC, maka config yang dijalankan adalah config SPIN-BC. Dalam config ini *networkType* diatur menjadi Flood. Berikut adalah penjelasan beberapa parameter penting yang digunakan.

1. *****.node[*].networkType* = "Flood"**
 ➔ Tipe Jaringan protokol SPIN-BC yang digunakan yaitu "Flood"
2. *****.node[*].netwl.headerLength* = 24 bit**
 ➔ panjang header *network* adalah 24 bit
3. *****.node[0].appl.nbPackets* = 10**
 ➔ *Node* yang menjadi sender data adalah *node* [10]
4. *****.node[*].appl.nbPackets* = 0**
 ➔ *Node* paket number [0]
5. *****.node[*].appl.destAddr* = -1**
 ➔ *Node* yang dituju adalah *node* [-1]
6. *****.node[*].appl.broadcastpackets* = true**
7. *****.node[*].appl.trafficType* = "periodic"**
 ➔ Data yang dikirimkan bertipe "periodik"
8. *****.node[*].appl.trafficParam* = 30 s**
 ➔ Waktu yang dibutuhkan dalam pengiriman 1 paket membutuhkan 30 s
9. *****.node[*].appl.initializationTime* = 10 s**
 ➔ Data akan dikirimkan secara periodik setiap detik oleh sender.
10. *****.node[*].appl.headerLength* = 50 byte**
 ➔ panjang header aplikasi adalah 50 byte.

3.1.3 Routing SPIN-RL

Routing SPIN-RL adalah mekanisme broadcast data dari satu set sensor menuju ke semua *node*. Routing SPIN-RL memperluas kemampuan dari SPIN-BC untuk menambahkan kehandalan dan menanggulangi kesalahan transmisi perintah yang disebabkan oleh data lossy.

```
[Config Spin-RL]
description="Wireless sensor network with the sink Spin-RL the network periodically."
**.node[*].networkType = "AdaptiveProbabilisticBroadcast"
**.node[*].netwl.stats = true
**.node[*].netwl.headerLength = 24 bit
**.node[0].appl.nbPackets = 10
**.node[*].appl.nbPackets = 0
**.node[*].appl.destAddr = 0
**.node[*].appl.broadcastPackets = true
**.node[*].appl.trafficType = "periodic"
**.node[*].appl.trafficParam = 30 s # each node sends 1 packet every 30 seconds
**.node[*].appl.initializationTime = 30 s
**.node[*].appl.headerLength = 50 byte
```

Gambar 10. Konfigurasi SPIN-RL

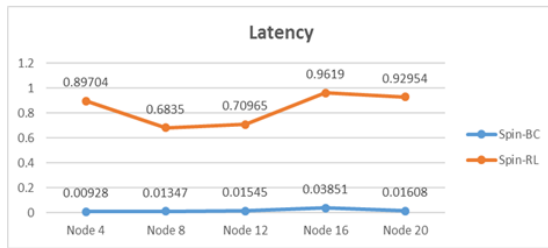
Pada Gambar 10 konfigurasi simulasi menggunakan SPIN-RL, maka config yang dijalankan adalah config SPIN-RL. Dalam config ini *networkType* diatur menjadi AdaptiveprobabilisticBroadcast. Berikut adalah penjelasan beberapa parameter penting yang digunakan.

1. *****.node[*].networkType* = "AdaptiveprobabilisticBroadcast"**
 ➔ Tipe Jaringan protokol SPIN-RL yang digunakan yaitu "AdaptiveprobabilisticBroadcast"
2. *****.node[*].netwl.headerLength* = 24 bit**
 ➔ panjang header *network* adalah 24 bit
3. *****.node[0].appl.nbPackets* = 10**
 ➔ *Node* yang menjadi sender data adalah *node* [10]
4. *****.node[*].appl.nbPackets* = 10**
 ➔ *Node* paket number [10]
5. *****.node[*].appl.destAddr* = 0,**
6. *****.node[*].broadcastpackets* = true**
7. *****.node[*].appl.trafficType* = "periodic"**
 ➔ Data yang dikirimkan bertipe "periodik"
8. *****.node[*].appl.trafficParam* = 30s**
 ➔ Waktu yang dibutuhkan dalam pengiriman 1 paket membutuhkan 30 s
9. *****.node[*].appl.initializationTime* = 30 s**
 ➔ Data akan dikirimkan secara periodik setiap detik oleh sender.
10. *****.node[*].appl.headerLength* = 50 byte**
 ➔ panjang header aplikasi adalah 50 byte.

4. PENGUJIAN DAN ANALISIS

4.1 Pengujian Terhadap Parameter Latency

Pada pengujian parameter *Latency* dengan banyaknya *node* sebanyak 4,8,12,16 dan 20 maka didapatkan hasil sebagai berikut:



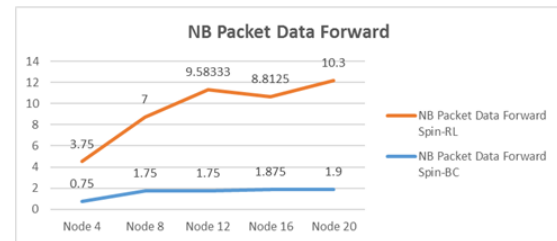
Gambar 11. Grafik Hasil Latency

Berdasarkan hasil pengujian, Pada grafik tersebut dapat diketahui bahwa rata-rata Latency pengujian protokol routing SPIN-BC dengan waktu 30 s adalah sebagai berikut node sebanyak 4 bernilai 0,00928, node sebanyak 8 bernilai 0,01347, node sebanyak 12 bernilai 0,01545, node sebanyak 16 bernilai 0,03851, dan node sebanyak 20 bernilai 0,01608. Sedangkan protokol routing SPIN-RL dengan waktu yang sama memiliki hasil sebagai berikut node sebanyak 4 bernilai 0,89704, node sebanyak 8 bernilai 0,6835, node sebanyak 12 bernilai 0,70965, node sebanyak 16 bernilai 0,9619, dan node sebanyak 20 bernilai 0,92954. Dari hasil grafik 5.1 dapat dianalisis bahwa protokol routing SPIN-BC dan routing SPIN-RL memiliki perbedaan dalam parameter Latency di pengiriman paket data. Protokol routing SPIN-RL memiliki hasil node dari 4,8,12,16, dan 20 yang cukup terlihat dengan rata-rata 0.018558. Sedangkan Protocol routing SPIN – BC juga memiliki perbedaan namun dengan loncatan angka yang tidak begitu jauh dengan rata-rata 0,836326. Pada hal ini paket yang dikirimkan akan diterima oleh semua node lain dalam jarak tertentu dari node pengirim. Protokol SPIN menerima keuntungan dari kemampuan saluran penyiaran dan mengharuskan node yang telah menerima perintah ADV untuk tidak segera merespon dengan perintah REQ. Sebaliknya, node akan menunggu dalam waktu tertentu, selama kemampuan saluran penyiaran memonitor saluran komunikasi. Jika perintah nodehears REQ dikeluarkan oleh node lain yang tertarik untuk menerima data, permintaan tersebut akan ditolak secara otomatis, sehingga menghilangkan permintaan yang sama. Ketika hal ini terjadi maka waktu pengiriman data setiap node akan berbeda.

4.2 Pengujian Terhadap Parameter NB Packet Data Forward

Pada pengujian parameter NB Packet Data Forward dengan banyaknya node sebanyak 4,8,12,16 dan 20 maka didapatkan hasil sebagai

berikut:

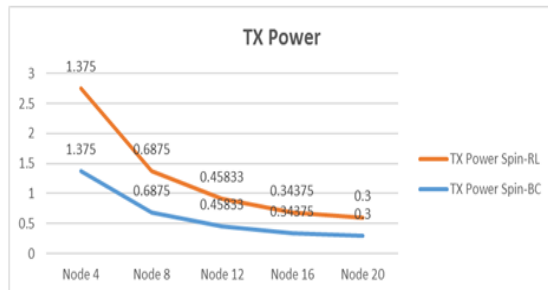


Gambar 12. Grafik Hasil NB Packet Data Forward

Berdasarkan pengujian Rata-rata NB Packet Data Forward yang digunakan untuk meneruskan jumlah paket data yang di teruskan oleh pesan REQ pada routing SPIN-BC dan SPIN-RL. Pada grafik tersebut dapat diketahui bahwa rata-rata NB Packet Data Forward pada pengujian protokol routing SPIN-BC dengan waktu 30 s dengan node sebanyak 4 bernilai 0,75, node sebanyak 8 bernilai 1,75, node sebanyak 12 bernilai 1,75, node sebanyak 16 bernilai 1,875, dan node sebanyak 20 bernilai 1,9. Sedangkan protokol routing SPIN-RL dengan waktu 30 s dengan node sebanyak 4 bernilai 3,75, node sebanyak 8 bernilai 7, node sebanyak 12 bernilai 9,58333, node sebanyak 16 bernilai 8,8125, dan node sebanyak 20 bernilai 10,3. Dari hasil grafik 5.2 dapat dianalisis bahwa protokol routing SPIN-BC dan routing SPIN-RL dapat diketahui hasil rata-rata NB Packet Data Forward semakin naik jika jumlah node semakin banyak dengan rata-rata routing SPIN-BC 1.605 dan rata-rata routing SPIN-RL 7,88917. Protokol SPIN menggunakan 3 tipe jenis perintah. Perintah jenis pertama adalah ADV, yang biasanya digunakan untuk mengiklankan data baru antara node-node. Sebuah node jaringan yang memiliki data yang bisa dibagikan dengan node yang ada di sebuah jaringan dapat mengiklankan datanya sendiri dengan terlebih dahulu mengirimkan perintah ADV yang berisi metada yang menguraikan data-data tersebut. Jenis perintah kedua adalah REQ, yang biasanya digunakan untuk meminta data menarik yang telah diiklankan. Setelah menerima perintah ADV yang mengandung metadada, sebuah node jaringan biasanya tertarik untuk menerima data spesifik yang dikirimkan perintah REQ node iklan metadada, yang kemudian akan mengirimkan data yang diminta. Pada routing SPIN tidak semua node mau menerima perintah data REQ dikarenakan tidak semua node mau menerima perintah ADV. Ketika hal tersebut terjadi maka setiap node akan menghasilkan data yang berbeda.

4.3 Pengujian Terhadap Parameter TX Power

Pada pengujian parameter TX Power dengan banyaknya node sebanyak 4,8,12,16 dan 20 maka didapatkan hasil sebagai berikut:

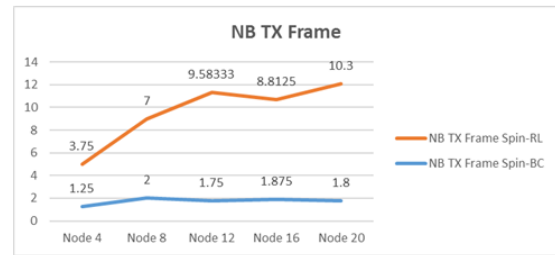


Gambar 13. Grafik Hasil TX Power

Pada grafik tersebut dapat diketahui bahwa rata-rata TX Power pada pengujian protokol routing SPIN-BC dengan waktu 30 s dengan node sebanyak 4 bernilai 1.375, node sebanyak 8 bernilai 0.6875, node sebanyak 12 bernilai 0.45833, node sebanyak 16 bernilai 0.34375, dan node sebanyak 20 bernilai 0.3. Sedangkan protokol routing SPIN-RL dengan waktu 30 s dengan node sebanyak 4 bernilai 1.375, node sebanyak 8 bernilai 0.6875, node sebanyak 12 bernilai 0.45833, node sebanyak 16 bernilai 0.34375, dan node sebanyak 20 bernilai 0.3. Dari hasil grafik 5.3 dapat dianalisis bahwa protokol routing SPIN-BC dan routing SPIN-RL dapat diketahui rata-rata TX Power semakin menurun jika jumlah node semakin banyak dengan rata-rata 0.63292. Dan yang digunakan dalam negosiasi SPIN disebut meta-data. Dari hasil tersebut, ia membuktikan bahwa negosiasi meta-data membuat node SPIN mengirimkan bahkan satu paket data berlebihan dalam sebuah jaringan. Hasilnya energi yang digunakan oleh node secara drastis menurun. Ini karena node terhubung dengan node jalur terpendek dan setelah itu link gradien menggunakan banyak energi atau mentransmisikan dan menerima paket. Dengan demikian, mereka menghasilkan overhead dan mengurangi waktu hidup node dalam jaringan. Ketika ini terjadi maka hasil setiap node akan berubah.

4.4 Pengujian Terhadap Parameter Nb TxFrame

Pada pengujian parameter Nb Tx Frame dengan banyaknya node sebanyak 4,8,12,16 dan 20 maka didapatkan hasil sebagai berikut:

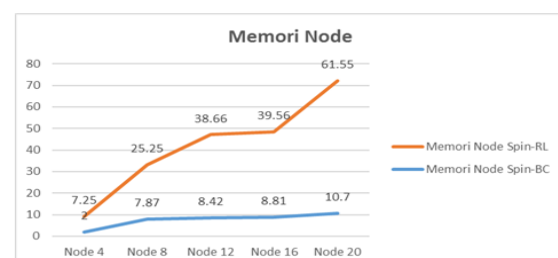


Gambar 14. Grafik Hasil Nb Tx Frame

Pada grafik tersebut dapat diketahui bahwa rata-rata protokol routing SPIN-BC dengan waktu 30 s dengan node sebanyak 4 bernilai 0,75, node sebanyak 8 bernilai 1,75, node sebanyak 12 bernilai 1,75, node sebanyak 16 bernilai 1,875, dan node sebanyak 20 bernilai 1,9. Sedangkan protokol routing SPIN-RL dengan waktu 30 s dengan node sebanyak 4 bernilai 3,75, node sebanyak 8 bernilai 7, node sebanyak 12 bernilai 9,58333, node sebanyak 16 bernilai 8,8125, dan node sebanyak 20 bernilai 10,3. Dari hasil grafik 5.2 dapat dianalisis bahwa protokol routing SPIN-BC dan routing SPIN-RL dapat diketahui bahwa rata-rata Nb Tx Frame, jika jumlah node semakin bertambah maka hasil jumlah node semakin meningkat dengan rata-rata routing SPIN-BC 1.605 dan rata-rata routing SPIN-RL 7,88917. Node SPIN mengirimkan bahkan satu paket data berlebihan dalam sebuah jaringan. Ini karena node yang terhubung dengan node jalur terpendek dan setelah itu link gradien menggunakan banyak energi atau mentransmisikan dan menerima paket Dengan demikian, mereka menghasilkan overhead dan mengurangi waktu hidup node dalam jaringan. Ketika ini terjadi link untuk setiap node akan berubah.

4.5 Pengujian Terhadap Parameter Memori Node

Pada pengujian parameter Memori Node dengan banyaknya node sebanyak 4,8,12,16 dan 20 maka didapatkan hasil sebagai berikut:



Gambar 15. Grafik Hasil Memori Node

Pada grafik tersebut dapat diketahui bahwa

rata-rata protokol *routing* SPIN-BC dengan waktu 30 s pada *node* sebanyak 4 bernilai 2, *node* sebanyak 8 bernilai 7.87, *node* sebanyak 12 bernilai 8.42, *node* sebanyak 16 bernilai 8.81, dan *node* sebanyak 20 bernilai 10.7. Sedangkan pada protokol *routing* SPIN-RL dengan waktu 30 s diketahui rata-rata pada *node* sebanyak 4 bernilai 7.25, *node* sebanyak 8 bernilai 25.25, *node* sebanyak 12 bernilai 38.66, *node* sebanyak 16 bernilai 39.56, dan *node* sebanyak 20 bernilai 61.55. Dari hasil grafik 5.5 dapat dianalisis bahwa protokol *routing* SPIN-BC dan *routing* SPIN-RL dapat diketahui bahwa rata-rata Memori Node akan semakin membutuhkan memory yang lebih banyak dengan jumlah *node* semakin banyak dengan rata-rata *routing* SPIN-BC 7,56 dan rata-rata *routing* SPIN-RL 34,454.

5. KESIMPULAN

1. Algoritma *routing* SPIN-BC dan SPIN-RL dapat diimplementasikan pada *Wireless SensorNetwork* menggunakan *network simulator* (OMNET++). Dengan menentukan area simulasi yang akan digunakan pada simulator. Setelah area simulasi telah ditentukan, *node-node* yang akan digunakan disiapkan dan ditentukan jumlahnya sesuai dengan kebutuhan. Jika sudah dilakukan, maka tahap selanjutnya adalah konfigurasi pada *node* yang digunakan dalam simulasi. Ketika konfigurasi yang dilakukan berhasil, maka *node sink* dibuat dan memposisikan *node sink* beserta *node* yang lain sesuai dengan aturan. Kemudian tambahkan *link* pada *node* yang ada dengan *sinknya*. Jika semua sudah selesai, tentukan *routing* yang digunakan pada simulasinya. Pada percobaan ini, dipilih *routing* SPIN-BC dan SPIN-RL. Selanjutnya simulasi dapat dilakukan. Jika simulasi tidak berhasil, maka simulasi harus dimulai dari penentuan jumlah *node* kembali. Jika simulasi berhasil, dapat dilakukan pengambilan data sesuai dengan parameter yang dibutuhkan. Dari pengambilan data tersebut dapat ditarik kesimpulan menggunakan grafik.
2. Berdasarkan hasil analisis yang telah dilakukan *routing*
 - 2.a Pengujian hasil rata-rata latency 0.01856 ms, SPIN-BC menggunakan *node* sebanyak 4,8,12,16, dan 20 sedangkan pada *routing* SPIN-RL menggunakan *node* sebanyak 4,8,12,16

dan 20 memiliki nilai rata-rata 0.83633 ms.

- 2.b Pengujian hasil rata-rata *NB Packet Data Forward* yang digunakan untuk meneruskan jumlah paket data yang diteruskan oleh pesan REQ pada *routing* SPIN-BC rata-rata yang didapat 1.605 ms dan SPIN-RL rata-rata yang didapat 7.88917 ms.
- 2.c Pengujian hasil rata-rata yang diperoleh TX Power pada pengujian protocol *routing* SPIN-BC dan SPIN-RL adalah 0,63292 mw.
- 2.d Pengujian hasil rata-rata dari parameter Nb Tx *Frame* pengiriman paket data *routing* SPIN-BC adalah 1,7 ms sedangkan SPIN-RL rata-rata yang diperoleh 7,9 ms dari *sourcenode* menuju paket yang diterima oleh *sinknode*.
- 2.e Pengujian hasil rata-rata dari memori *node* yang digunakan saat *node* mengirim paket data pada SPIN-BC 7,56 MB sedangkan SPIN-RL 34,454 MB.

Berdasarkan dari hasil rata-rata diatas dapat disimpulkan bahwa SPIN-RL memiliki performa yang lebih baik di bandingkan dengan SPIN-BC, karena SPIN-RL dapat menanggulangi kesalahan transmisi perintah yang disebabkan oleh data loss.

DAFTAR PUSTAKA

- A. Varga & R. Hornig. 2008. *An overview of omnet++ simulator*, March 03-07, France.
- Ahmad Ramadhan. 2014. Pengertian *network latency*. [Online]
Available at : <http://dadankun.blogspot.co.id/2014/11/pengetian-network-latency.html>
[Diakses 20 juni 2017]
- Alexander., & Sekercioglu, Y. Ahmed. *Performance of Telecommunication Networks Experiment V: Introduction to Modeling Wireless Networks and Their Network Layer Protocols*. Department of Electrical & Computer Systems Engineering, Monash University.
- Alexandru, & Valentin. 2015. *Performance Evaluation of routing algorithms used in largescale WSN: A Step Towards A Smart*

City, Stefan cel Mare, University of Suceava.

- G. Simon, et al. 2004. *Sensor network-based countersniper system*. In Proc. 2nd ACM International Conference on Embedded Networked Sensor Systems (SenSys04), 2
- J. Kulik, et al. 2002. *Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks*, Vol. 8, 2002, pp. 169–185
- M. Thangaraj et al. 2014. *A study of energy model in wsn routing using omnet++*, International Journal of Application or Innovation in Engineering & Management (IJAIEM), vol. 3, no. 6.