

IMPLEMENTASI LOAD BALANCING MENGGUNAKAN NS-3 PADA NAMED DATA NETWORKING

Implementation Load Balancing Using NS-3 in Named Data Networking

Nesya Alvioni Safitri¹, Ratna Mayasari², Tody Ariefianto Wibowo³

^{1,2,3} Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

nesyaas@students.telkomuniversity.ac.id¹, ratnamayasari@telkomuniversity.ac.id²,
ariefianto@telkomuniversity.ac.id³

Abstrak - Bertambahnya kebutuhan pengguna (*consumer*) membutuhkan arsitektur jaringan yang semakin handal. *Named Data Networking* (NDN) hadir sebagai salah satu arsitektur jaringan masa depan yang berfokus pada *data-centric*. Arsitektur NDN memiliki potensi terkait pengelolaan data yang lebih dikonsentrasikan pada pertukaran data daripada letak asal data tersebut. Jaringan NDN yang masih baru memerlukan mekanisme *load balancing* untuk mengatasi ketidakseimbangan pembagian tugas saat penyedia layanan (*producer*) melayani *request* yang datang dari *consumer*. *Random Load Balancing* (RLB) merupakan salah satu algoritma pada mekanisme *load balancing* NDN. Penerapan algoritma RLB mampu memaksimalkan peran *producer* agar dapat mendistribusikan *request consumer* secara tepat sehingga beban komputasi lebih rendah. Pada Tugas Akhir ini dilakukan simulasi penelitian algoritma RLB pada mekanisme *load balancing* NDN. Perubahan parameter pengujian meliputi frekuensi *interest packet*, ukuran *bandwidth*, ukuran *payload*, jumlah *node consumer*, jumlah *node producer*, jumlah *node router* dan perbandingan tanpa menggunakan algoritma RLB untuk mengetahui performa jaringan. Dari simulasi yang dilakukan didapatkan data 48,4% lebih baik dengan menggunakan algoritma RLB daripada tanpa menggunakan algoritma tersebut. Algoritma ini mendukung mekanisme *load balancing* pada jaringan NDN untuk memaksimalkan potensi *producer* yang tersedia. Hasil analisis berdasarkan parameter *Quality of Service* (QoS) menunjukkan nilai yang konsisten terhadap efek perubahan selama pengujian.

Kata Kunci: NDN, *load balancing*, algoritma RLB, NS-3, ndnSIM

Abstract – The increasing needs of users (*consumers*) so it is necessary to have a more reliable network architecture. *Named Data Networking* (NDN) as one of the future network architectures that focus on *data-centric*. The NDN architecture has more potential for concentrating data management on data exchange than the data location. NDN requires additional load balancing mechanism to overcome the imbalance of the task distribution when the service provider (*producer*) serves requests from consumers. *Random Load Balancing* (RLB) is one of the algorithms of load balancing NDN. The application of the RLB can maximize the producers role to distribute consumer demand appropriately so the computational load is getting lower. This final project examine the RLB algorithm on the load balancing NDN. The testing parameters involves interest packet frequency, bandwidth size, payload size, number of consumer nodes, number of producer nodes, number of router nodes, and comparisons without using the RLB. From the simulation, the obtaining data using the RLB is 48.4% better than without using the algorithm. This algorithm supports load balancing NDN to maximize the potential of available producers. The analysis results are based on the *Quality of Service* (QoS) parameter in accord with the change effect during the test.

Keywords: NDN, *load balancing*, RLB algorithm, NS-3, ndnSIM.

1. Pendahuluan

1.1 Latar Belakang

Teknologi telekomunikasi masa depan berkembang seiring dengan permintaan pengguna (*consumer*) yang semakin meningkat. Adanya penelitian baru mengenai arsitektur jaringan yang semakin tangguh dan kompatibel diharapkan dapat memfasilitasi perancangan dan implementasi teknologi bidang jaringan ke depannya[1]. Berdasarkan ITU-T salah satu tujuan pengembangan jaringan masa depan berkonsentrasi pada optimalisasi pengelolaan data. Hal ini akan memberikan kenyamanan *consumer* untuk mengakses data yang diinginkan secara mudah, cepat, tepat dan aman tanpa dipengaruhi faktor lokasi ketersediaan data. *Named Data Networking* (NDN) merupakan arsitektur jaringan masa depan yang berfokus mendukung proses komunikasi berbasis konten. NDN memiliki konsep transisi dari pengalamatan *host* secara tradisional (*Internet Protocol Networking*) menjadi pengalamatan dengan konten yang dikehendaki. NDN memungkinkannya *host* dapat memberi nama berdasarkan data atau konten tersebut[2]. Untuk mendukung pengembangan jaringan NDN maka juga dibutuhkan adanya mekanisme *load balancing* internet masa depan[3]. *Load balancing* merupakan mekanisme pendistribusian permintaan *consumer* di antara beberapa penyedia layanan (*producer*) terutama pada layanan yang bersifat dinamis[4]. *Producer* dengan kondisi beban tinggi berpotensi tidak dapat memenuhi permintaan *consumer* sehingga data yang diinginkan tidak terdistribusi dengan baik dan disinyalir terdapat *producer* yang

kurang dimanfaatkan. Pendistribusian beban komputasi secara adaptif dan merata dapat memaksimalkan pemanfaatan penggunaan sumber daya yang tersedia. Terdapat juga dampak positif lainnya dari penerapan mekanisme *load balancing*, yaitu berpotensi menurunkan waktu respon tugas rata-rata *consumer* sehingga dapat meningkatkan kepuasan *consumer*. Penerapan mekanisme *load balancing* pada jaringan NDN dapat menggunakan salah satu algoritma, yaitu *Random Load Balancing* (RLB)[5]. Algoritma RLB berkonsentrasi untuk mengatasi permasalahan penjadwalan *server*/penyedia layanan (*server scheduling*). Algoritma RLB diaplikasikan saat jaringan berada di kondisi beban tinggi. Hal ini berguna untuk mengurangi beban pelayanan pada sisi *producer*. Permintaan dari *consumer* diterima terlebih dahulu oleh *Load Balancer* lalu akan diteruskan ke *producer* yang paling berpotensi dapat melayaninya. Algoritma RLB dapat memetakan *producer* mana yang memiliki beban komputasi lebih rendah sehingga dapat merespon dengan lebih cepat dan menghemat waktu pelayanan yang dibutuhkan. Pada Tugas Akhir ini akan dilakukan implementasi uji coba algoritma RLB pada mekanisme *load balancing* di jaringan NDN dengan menggunakan *Network Simulator 3* (NS-3). Pada penelitian sebelumnya parameter pengujian yang diterapkan masih terbatas pada frekuensi *interest packet*, ukuran *bandwidth* dan ukuran *payload* dengan parameter *Quality of Service* (QoS) *packet drop* saja[5]. Pada pelaksanaan Tugas Akhir ini menggunakan parameter pengujian yang lebih lengkap daripada sebelumnya untuk meningkatkan evaluasi sistem algoritma RLB. Implementasi algoritma ini dipengaruhi dengan adanya variasi perubahan parameter pengujian meliputi frekuensi *interest packet*, ukuran *bandwidth*, ukuran *payload*, jumlah *node consumer*, jumlah *node producer*, jumlah *node router* dan tanpa menggunakan algoritma RLB. Selanjutnya hasil pengujian akan dianalisis berdasarkan parameter QoS, yaitu aspek *delay*, *throughput* dan *packet drop*.

1.2 Tujuan dan Manfaat

Tujuan dan manfaat dari disusunnya Tugas Akhir ini adalah :

1. Melakukan simulasi dengan menggunakan *Network Simulator 3* (NS-3) dalam rangka pengujian algoritma RLB pada jaringan NDN sehingga dapat melihat aspek performa jaringan meliputi *delay*, *throughput* dan *packet drop*.
2. Mengetahui dan menganalisis aspek performa jaringan dikarenakan adanya perubahan beban jaringan dalam skenario pengujian.
3. Mengevaluasi dampak dari implementasi algoritma RLB yang mampu mendukung mekanisme *load balancing* pada jaringan NDN.
4. Memberikan penjelasan dari data yang didapatkan pada penelitian ini bahwa algoritma RLB dapat meningkatkan efisiensi mekanisme *load balancing* di jaringan NDN.

1.3 Rumusan Masalah

Rumusan permasalahan yang akan dibahas pada Tugas Akhir ini adalah :

1. Bagaimana pengaruh implementasi algoritma RLB dengan menggunakan NS-3 pada jaringan NDN?
2. Bagaimana pengaruh perubahan parameter pengujian terhadap implementasi algoritma RLB dengan menggunakan NS-3 pada jaringan NDN?

1.4 Batasan Masalah

Batasan masalah yang ditetapkan pada Tugas Akhir ini adalah :

1. Implementasi dimodelkan dalam skala virtualisasi dengan menggunakan NS-3 pada *NDN Simulator* (ndnSIM).
2. Mekanisme sistem yang dibuat tidak termasuk mekanisme pertukaran data ke *consumer*, melainkan berfokus seberapa efisien RLB dapat mendukung mekanisme *load balancing* pada jaringan NDN.
3. Tidak membahas sistem penamaan (*prefix*) jaringan NDN dalam pengujian.
4. Pengujian dilaksanakan dalam kondisi jaringan ideal (tidak terdapat serangan keamanan jaringan).
5. Tidak dibahas mekanisme *routing* pada jaringan NDN.
6. Analisis performa jaringan ditinjau berdasarkan parameter *delay*, *throughput* dan *packet drop*.

2. Dasar Teori

2.1 Named Data Networking

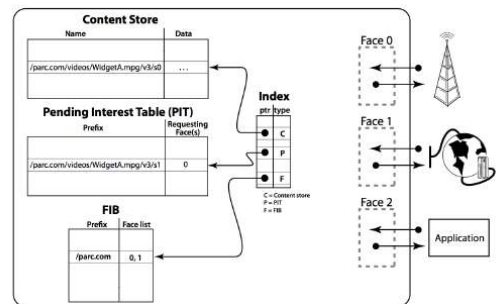
Named Data Networking (NDN) tergolong teknologi baru pada perkembangan jaringan internet[8]. Paradigma ini muncul dikarenakan semakin meningkatnya permintaan pelanggan terhadap penggunaan internet. NDN merupakan model komunikasi yang proses permintaan konten berbasis pada nama konten. NDN bergerak dari pihak yang meminta layanan menuju pihak yang menyediakan layanan. NDN menyediakan panggilan dengan nama dan penyimpanan dalam *cache* sebagai mekanisme utama untuk pengiriman konten dengan *throughput* tinggi.

2.1.1 Arsitektur Named Data Networking

Penerapan komunikasi NDN dimulai dari permintaan *consumer* yang mengirimkan *Interest Packet* berisi nama data yang diinginkan seperti contoh berikut `/parc/videos/WidgetA.mpg` menuju *router* NDN[8]. Pada setiap *node router* NDN memiliki beberapa fungsi, diantaranya :

1. *Content Store* (CS) adalah bagian penyimpanan memori utama pada *router* untuk menyimpan konten dan memetakan nama konten dengan *Interest Packet* yang sesuai.
2. *Forwarding Information Base* (FIB) adalah struktur data yang sama dengan tabel IP *routing*.
3. *Pending Interest Table* (PIT) adalah suatu penyimpanan kumpulan *Interest Packet* beserta *face* yang meminta data.

Untuk memahami lebih jelas mengenai konfigurasi NDN *router* dapat dilihat pada Gambar 2.1.



Gambar 2.1 : Konfigurasi Sebuah NDN Router[8]

2.2 Load Balancing

Load balancing mutlak diperlukan untuk mendistribusikan beban di antara replika layanan di NDN[4]. Hal ini bertujuan menghindari situasi di mana beberapa replika kelebihan beban sementara yang lain tidak aktif. Mekanisme *load balancing* dapat ditinjau lebih lanjut berdasarkan beberapa aspek yang dibahas pada sub-bab berikutnya.

Terdapat tiga dimensi untuk desain solusi *load balancing* melalui NDN:

1. *Metrics* adalah dimensi yang memungkinkan *load balancing* untuk membuat keputusan.
2. *Purity* adalah dimensi dengan penerapan sistem pada *load balancing*.
3. *Location* adalah dimensi yang berkaitan dengan komponen jaringan yang bertanggung jawab pada penerapan sistem *load balancing* di NDN.

2.2.1 Congestion Control

Secara umum, *congestion control* adalah mekanisme yang diterapkan di lapisan *transport* untuk mendeteksi *overloaded links* dan menyesuaikan aliran jaringan untuk menjaga stabilitas jaringan, *throughput*, dan alokasi sumber daya yang tepat untuk klien[4]. Berikut adalah karakteristik dasar *congestion control* pada NDN :

1. NDN adalah sistem *pull-based*.
2. Router NDN tidak menyebarkan *interest* jika sudah ada di PIT mereka.
3. Router NDN memiliki kemampuan *caching*.

2.2.2 Forwarding Strategy

Pada NDN, *forwarding strategy* merupakan proses penerusan *interest* menuju *interface* terbaik berdasarkan *routing protocol* yang dipengaruhi oleh beberapa parameter diantaranya *link load* dan *availability*[4].

2.2.3 Data Centre Management

Pada NDN terdapat perkembangan *monitoring resource* dan *load balancing* pada *data centre* di arsitektur NDN[4]. Pada penerapannya terdapat dua hal yang dapat diperhatikan yaitu menggabungkan jaringan berbasis IP dan NDN dalam model jaringan *hybrid* atau hanya menggunakan NDN untuk *data center management*.

2.3 Random Load Balancing (RLB)

Algoritma *Random Load Balancing* (RLB) merupakan sebuah mekanisme *load balancing* NDN yang berfokus pada *server scheduling* menggunakan *packet forwarding strategy*[5]. RLB akan memilih *producer* mana yang dapat melayani *interest* lalu akan mengirimkan *data packet* menuju *customer*. Algoritma ini diterapkan dengan adanya *multiple producers* yang menyediakan beragam nama paket untuk *multiple consumers*. Untuk memahami lebih lanjut mengenai algoritma RLB maka dapat dijabarkan berdasarkan kaidah tiga dimensi desain solusi *load balancing* sebagai berikut[4] :

1.Metric

RLB berfokus pada *server metric* dimana dapat mengidentifikasi kondisi producer dalam keadaan *overloaded* atau *underloaded*[5].

2.Location

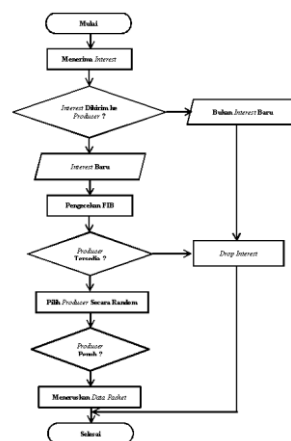
RLB *forwarding strategy* terletak di *router*. Router akan menerima semua incoming interest lalu meneruskannya menuju producer berdasarkan algoritma load balancing yang digunakan dan melewati data packet kembali ke consumer. Router memiliki metode metric collection tertentu dengan penerapan forwarding strategy khusus[5].

3.NDN-Purity

Infrastruktur NDN yang digunakan yaitu *Pure NDN*. di mana masih menerapkan konsep NDN awal. Penerapan *management* dan *control communication* dengan *controller* mengikuti paradigma *interest* atau data yang normal di NDN[5].

2.3.1 Konfigurasi Algoritma RLB

Konfigurasi algoritma RLB disusun menggunakan *framework Network Simulator 3 (NS-3)* dan didukung oleh *platform* simulasi *NDN Simulator (ndnSIM)*. Algoritma ini dibuat pada NS-3 berupa *file* dengan format *.cpp* dan ditulis dengan bahasa pemrograman C++[13]. Untuk memudahkan memahami mekanisme algoritma RLB maka dapat dipahami lebih lanjut menggunakan Gambar 2.2.



Gambar 2.4 : Flowchart Algoritma RLB

2.4 Simulator Jaringan

2.4.1 Network Simulator 3 Framework

Network Simulator 3 (NS-3) adalah *platform* simulasi *open-source network* berdasarkan *discrete event scheduling*[11]. NS-3 juga dapat mengurangi kebutuhan memori dalam simulasi skala besar dikarenakan paket NS-3 dapat berisi *virtual payload*. Penerapan simulasi jaringan pada NS-3 ditentukan pada *key abstractions* sebagai berikut :

1. *Node* merupakan *computing entity* dasar yang dapat diprogram oleh pengguna.
2. *Application* merupakan sebuah program yang ditentukan oleh pengguna.
3. *Channel* merupakan *entity* saluran komunikasi yang terhubung pada node tertentu.
4. *Net Device* merupakan representasi dari *driver hardware* dan *software*.
5. *Topology Helpers* merupakan komponen *software* yang digunakan untuk memfasilitasi pembuatan, koordinasi dan konfigurasi parameter pada sebuah program.

2.5 NDN Simulator

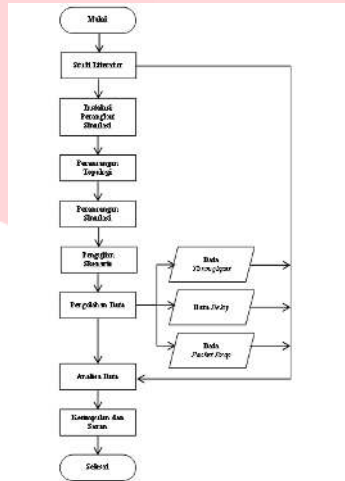
Struktur yang menyusun ndnSIM meliputi NS-3 *framework*, *NDN Forwarding Daemon (NFD)* dan *ndncxx* [11]. Simulasi ndnSIM juga dapat menggunakan komponen yang sudah ada meliputi *module*, *models*, implementasi *NetDevice* dan komponen NS-3 yang terintegrasi. Pada ndnSIM juga dapat dilakukan pengukuran hasil simulasi dengan beberapa variabel sebagai berikut[12] :

1. **L3 Rate Tracer** bertujuan untuk mengukur laju *interest packet* dan *data packet* dengan ukuran *byte* yang dapat diteruskan melalui *node* NDN.
2. **L2 Rate Tracer** bertujuan untuk mengukur *drop packet* pada *layer 2* (dua).
3. **Cs Tracer** bertujuan untuk mengukur statistik *cache hit* (data CS yang melayani permintaan) dan *cache miss* (data CS yang tidak melayani permintaan) pada *node* simulasi.
4. **App Delay Tracer** bertujuan untuk mengukur *delay* saat proses *interest packet* telah muncul dan *data packet* yang sesuai telah diterima.

3. Perancangan Sistem

3.1 Tahapan Penelitian

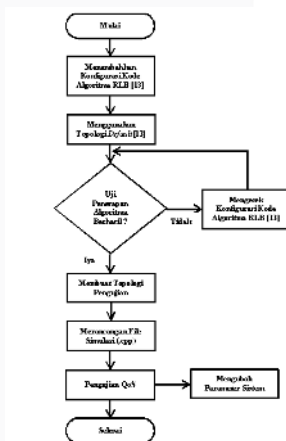
Pada pelaksanaan Tugas Akhir ini menggunakan arsitektur jaringan NDN dan menerapkan mekanisme *load balancing*[13]. Untuk memudahkan memahami proses tahapan pelaksanaan penelitian maka dapat dipahami lebih lanjut pada Gambar 3.1.



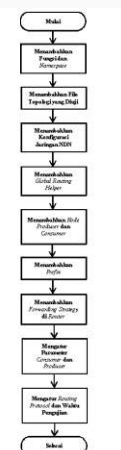
Gambar 3.1 : Flowchart Proses Tahapan Penelitian

3.2 Desain Sistem

Pada perancangan sistem dimulai dari tahapan penambahan konfigurasi algoritma RLB hingga pengujian performansi. Untuk memahami lebih lanjut proses perancangan sistem dijelaskan lebih lanjut pada Gambar 3.2. Untuk memahami lebih lanjut proses perancangan file simulasi dijelaskan lebih lanjut pada Gambar 3.3.



Gambar 3.2 : Flowchart Proses Perancangan Sistem



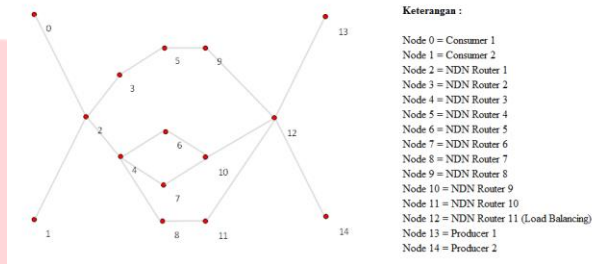
Gambar 3.3 : Flowchart Proses File Simulasi

3.3 Perangkat Simulasi

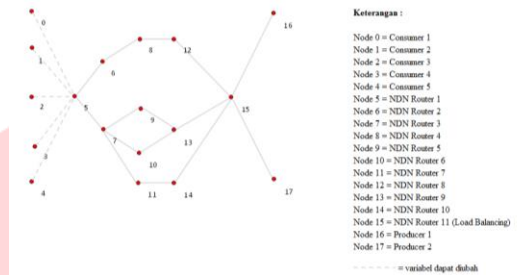
Pelaksanaan pengujian sistem dilakukan pada framework *Network Simulator 3 (NS-3)* dan didukung kinerja platform simulasi *NDN Simulator (ndnSIM)*. Semua kebutuhan simulasi ini perlu diinstal pada *Operating System (OS) Ubuntu 18.04.5 LTS* dengan *processor Intel(R) Core (TM) i3-4005U CPU @1.70 GHz* dan *RAM 8 GB*.

3.4 Topologi Pengujian

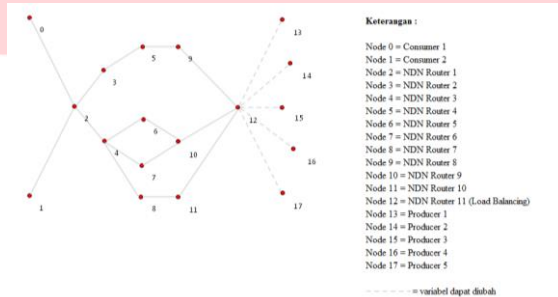
Pada saat simulasi dilaksanakan terdapat sejumlah *node* yang bertugas sebagai *consumer* dan berperan sebagai pengguna yang mengirimkan permintaan data atau *interest*. Selain itu, terdapat juga *node* yang bertugas sebagai *producer* dan berperan sebagai penyedia konten utama. Simulasi dilakukan dengan menerapkan algoritma RLB pada mekanisme *load balancing* NDN yang diterapkan pada *load balancer* di dalam topologi. Beberapa desain topologi dapat dicermati lebih lanjut seperti pada Gambar 3.4, Gambar 3.5, Gambar 3.6, Gambar 3.7 dan Gambar 3.8.



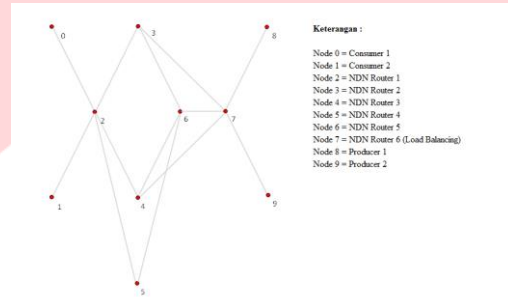
Gambar 3.4 : Topologi I dengan 11 Router



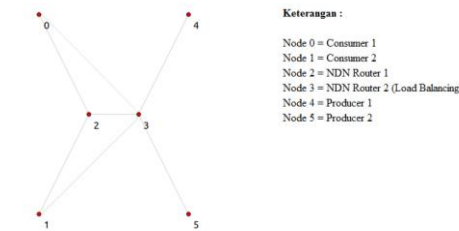
Gambar 3.5 : Topologi II dengan 11 Router dan Variabel Consumer Dapat Diubah



Gambar 3.6 : Topologi III dengan 11 Router dan Variabel Consumer Dapat Diubah



Gambar 3.7 : Topologi IV dengan 6 Router



Gambar 3.8 : Topologi V dengan 2 Router

3.5 Parameter Evaluasi Jaringan

3.5.1 Delay

Delay merupakan waktu tunggu *node* tertentu setelah mengirim *interest packet* hingga mendapat *data packet* yang sesuai[14].

$$\text{Delay} = \text{waktu}_{\text{penerimaan data}} - \text{waktu}_{\text{permintaan data}} \quad (3.1)$$

3.5.2 Throughput

Throughput merupakan total ukuran paket yang diterima *node* tujuan pada setiap satuan waktu.

$$\text{Throughput} = \frac{\sum \text{Ukuran Paket Diterima}}{\text{Waktu Pengamatan}} \quad (3.2)$$

3.5.3 Packet Drop

Packet Drop merupakan total paket yang gagal diterima pada *node* tertentu.

$$\text{Packet Drop} = \sum \text{Ukuran Paket Gagal Diterima} \quad (3.3)$$

4. Pengujian dan Analisis

4.1 Skenario I : Perubahan Frekuensi Interest

Perubahan frekuensi interest dari nilai 100, 200, 300 dan 400 *interest/detik*. Pengujian ini juga menggunakan beberapa nilai parameter yang diatur tetap meliputi menggunakan *forwarding strategy multicast*, *multipath routing protocol*, *payload (producer)* sebesar 8192 Bytes, *bandwidth (link)* 1 Mbps, 2 *node consumer*, 11 *node router* dan 2 *node producer*. Topologi yang digunakan seperti pada Gambar 3.4. Peningkatan frekuensi interest mempengaruhi permintaan data dari sisi *consumer* yang dikirimkan semakin meningkat setiap detiknya. Hasil simulasi dapat dilihat pada Tabel 4.1.

Tabel 4.1. Hasil Simulasi Skenario I

No	Parameter	Hasil Skenario I			
		100 Interest/s	200 Interest/s	300 Interest/s	400 Interest/s
1	<i>Delay (s)</i>	1.28083	7.82361	16.78968	27.05142
2	<i>Throughput (Kbps)</i>	95.69128	171.33769	256.03681	337.71634
3	<i>Packet Loss (packets)</i>	28.09943	68.22259	108.03954	148.50388

4.2 Skenario II : Perubahan Ukuran Bandwidth

Perubahan ukuran *bandwidth* dari nilai 1, 5, 10 dan 15 Mbps. Pengujian ini juga menggunakan beberapa nilai parameter yang diatur tetap meliputi menggunakan *forwarding strategy multicast, multipath routing protocol, payload (producer)* sebesar 8192 Bytes, frekuensi *interest* bernilai 200 *interest/detik*, 2 *node consumer*, 11 *node router* dan 2 *node producer*. Topologi yang digunakan seperti pada Gambar 3.4. Peningkatan ukuran *bandwidth* mempengaruhi kapasitas maksimum *communication link* pada sebuah jaringan untuk mengirimkan data dalam waktu tertentu. Hasil simulasi dapat dilihat pada Tabel 4.2.

Tabel 4.2. Hasil Simulasi Skenario II

No	Parameter	Hasil Skenario II			
		1 Mbps	5 Mbps	10 Mbps	15 Mbps
1	<i>Delay (s)</i>	7.82361	5.87239	3.19247	1.20983
2	<i>Throughput (Kbps)</i>	171.33769	350.66591	620.34912	890.45183
3	<i>Packet Loss (packets)</i>	68.22259	38.55672	18.50917	5.98564

4.3 Skenario III : Perubahan Ukuran Payload

Perubahan ukuran *payload* dari nilai 2048, 5120 dan 8192 Bytes. Pengujian ini juga menggunakan beberapa nilai parameter yang diatur tetap meliputi menggunakan *forwarding strategy multicast, multipath routing protocol, bandwidth (link)* 1 Mbps, frekuensi *interest* bernilai 200 *interest/detik*, 2 *node consumer*, 11 *node router* dan 2 *node producer*. Topologi yang digunakan seperti pada Gambar 3.4. *Interest* yang dikirimkan dari *consumer* akan dibalas *virtual payload* sebesar 2048, 5120 atau 8192 Bytes oleh *producer* pada tiap pengujian. Hasil simulasi dapat dilihat pada Tabel 4.3.

Tabel 4.3. Hasil Simulasi Skenario III

No	Parameter	Hasil Skenario III		
		2 KB	5 KB	8 KB
1	<i>Delay (s)</i>	4.36573	5.59927	7.82361
2	<i>Throughput (Kbps)</i>	109.26541	144.75703	171.33769
3	<i>Packet Loss (packets)</i>	13.44406	36.78887	68.22259

4.4 Skenario IV : Perubahan Jumlah Node Consumer

Perubahan jumlah *node consumer* dari nilai 2, 3 dan 5 *node*. Pengujian ini juga menggunakan beberapa nilai parameter yang diatur tetap meliputi menggunakan *forwarding strategy multicast, multipath routing protocol, bandwidth (link)* sebesar 1 Mbps, *payload (producer)* sebesar 8192 Bytes, frekuensi *interest* bernilai 200 *interest/detik*, 11 *node router* dan 2 *node producer*. *Interest* yang dikirimkan setiap detiknya dari sisi *consumer* akan meningkat seiring dengan adanya penambahan jumlah *node consumer*. Topologi yang digunakan seperti pada Gambar 3.5. Hasil simulasi dapat dilihat pada Tabel 4.4.

Tabel 4.4. Hasil Simulasi Skenario IV

No	Parameter	Hasil Skenario IV		
		2 Consumer	3 Consumer	5 Consumer
1	Delay (s)	7.82361	8.22416	9.87323
2	Throughput (Kbps)	171.33769	185.76469	197.29372
3	Packet Loss (packets)	68.22259	73.95868	78.88744

4.5 Skenario V : Perubahan Jumlah Node Producer

Perubahan jumlah *node producer* dari nilai 2, 3 dan 5 *node*. Pengujian ini juga menggunakan beberapa nilai parameter yang diatur tetap meliputi menggunakan *forwarding strategy multicast*, *multipath routing protocol*, *bandwidth (link)* sebesar 1 Mbps, *payload (producer)* sebesar 8192 Bytes, frekuensi *interest* bernilai 200 *interest/detik*, 11 *node router* dan 2 *node consumer*. Topologi yang digunakan seperti pada Gambar 3.6. Peningkatan jumlah *node producer* mempengaruhi adanya peningkatan kemampuan *producer* untuk melayani *interest packet* yang datang. Hasil simulasi dapat dilihat pada Tabel 4.5.

Tabel 4.5. Hasil Simulasi Skenario V

No	Parameter	Hasil Skenario V		
		2 Producer	3 Producer	5 Producer
1	Delay (s)	7.82361	6.01745	3.57842
2	Throughput (Kbps)	171.33769	226.43423	272.23634
3	Packet Loss (packets)	68.22259	57.92796	46.88761

4.6 Skenario VI : Perubahan Jumlah Node Router

Perubahan jumlah *node router* dari nilai 2, 6 dan 11 *node*. Pengujian ini juga menggunakan beberapa nilai parameter yang diatur tetap meliputi menggunakan *forwarding strategy multicast*, *multipath routing protocol*, *bandwidth (link)* sebesar 1 Mbps, *payload (producer)* sebesar 8192 Bytes, frekuensi *interest* bernilai 200 *interest/detik*, 2 *node consumer* dan 2 *node consumer*. Topologi yang digunakan seperti pada Gambar 3.4, Gambar 3.7 dan Gambar 3.8. Penambahan *router* menyebabkan pilihan rute *next hop* yang semakin banyak. Hasil simulasi dapat dilihat pada Tabel 4.6.

Tabel 4.6. Hasil Simulasi Skenario VI

No	Parameter	Hasil Skenario VI		
		1 Router	5 Router	10 Router
1	Delay (s)	1.59796	3.44864	7.82361
2	Throughput (Kbps)	427.51741	256.35442	171.33769
3	Packet Loss (packets)	18.44627	40.21542	68.22259

4.7 Skenario VII : Perubahan Mekanisme Dengan RLB vs Tanpa RLB

Pengujian berdasarkan perubahan penerapan mekanisme dengan algoritma RLB dan tanpa RLB. Pengujian ini juga menggunakan beberapa nilai parameter yang diatur tetap meliputi menggunakan *forwarding strategy multicast*, *multipath routing protocol*, *bandwidth (link)* sebesar 1 Mbps, *payload (producer)* sebesar 8192 Bytes, frekuensi *interest* bernilai 200 *interest/detik*, 11 *node router* 2 *node consumer* dan 5 *node producer*. Topologi yang digunakan seperti pada Gambar 3.6. Dengan menggunakan algoritma RLB menunjukkan hasil evaluasi lebih baik daripada tidak menggunakan RLB. Hasil simulasi dapat dilihat pada Tabel 4.7.

Tabel 4.7. Hasil Simulasi Skenario VII

No	Parameter	Hasil Skenario VII	
		Dengan RLB	Tanpa RLB
1	<i>Delay</i> (s)	3.57842	5.31195
2	<i>Throughput</i> (Kbps)	272.23634	140.38072
3	<i>Packet Loss</i> (packets)	46.88761	69.55065

4.8 Evaluasi Kerja Sistem

Berdasarkan hasil simulasi yang telah dilakukan dapat diketahui bahwa penerapan mekanisme *load balancing* pada jaringan NDN memberikan dampak yang signifikan. Jaringan NDN yang tergolong terobosan baru pada pengembangan jaringan masa depan tentunya masih perlu dilengkapi mekanisme *load balancing* yang mendukung kinerjanya semakin handal. Penerapan RLB sebagai salah satu algoritma pada mekanisme *load balancing* di jaringan NDN dapat diterapkan dengan baik selama pengujian berlangsung. Algoritma RLB memaksimalkan potensi *producer* yang tersedia pada sebuah topologi dengan berfokus pada fungsi *server scheduling*[5]. Penerapannya memberikan dampak baik pada sistem simulasi yang diujikan sehingga performa sistem responsif terhadap pemetaan beban komputasi pada sisi *producer* dengan nilai 48,4% lebih baik daripada tanpa menggunakan algoritma RLB. Pada pengujian *delay* dengan perubahan parameter ukuran *bandwidth* dan perubahan penerapan algoritma RLB menunjukkan nilai yang semakin membaik. Pada pengujian *throughput* dengan perubahan parameter frekuensi *interest*, ukuran *bandwidth*, ukuran *payload*, jumlah *node consumer*, jumlah *node producer* dan perubahan penerapan algoritma RLB menunjukkan nilai yang semakin membaik. Pada pengujian *packet drop* dengan perubahan parameter ukuran *bandwidth*, jumlah *node producer* dan perubahan penerapan algoritma RLB menunjukkan nilai yang semakin membaik. Hasil data yang didapat telah sesuai berdasarkan evaluasi parameter *Quality of Service* (QoS) karena menunjukkan nilai yang konsisten terhadap efek perubahan yang diberikan selama pengujian.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Pada penelitian Tugas Akhir ini telah disimulasikan penerapan algoritma *Random Load Balancing* (RLB) pada jaringan *Named Data Networking* (NDN). Adapun kesimpulan yang dapat diambil berdasarkan hasil Tugas Akhir ini sebagai berikut :

1. Perubahan frekuensi *interest* yang semakin meningkat menyebabkan semakin bertambahnya jumlah permintaan yang dikirimkan dari sisi *consumer* menuju *producer*. Oleh karena itu, nilai *delay*, *throughput* dan *packet drop* semakin meningkat.
2. Perubahan ukuran *bandwidth* yang semakin meningkat menyebabkan peningkatan kapasitas maksimum *link* saat proses transmisi data. Oleh karena itu, nilai *delay* dan *packet drop* semakin menurun, sedangkan nilai *throughput* jaringannya semakin meningkat.
3. Perubahan ukuran *payload* yang semakin meningkat menyebabkan ukuran *data packet* yang dikirimkan dari *producer* juga semakin meningkat. Oleh karena itu, nilai *delay*, *throughput* dan *packet drop* meningkat.
4. Perubahan jumlah *node consumer* yang semakin meningkat menyebabkan semakin bertambahnya jumlah permintaan (*interest packet*) yang dikirimkan dari sisi *consumer* menuju *producer*. Oleh karena itu, nilai *delay*, *throughput* dan *packet drop* semakin meningkat.
5. Perubahan jumlah *node producer* yang semakin meningkat menyebabkan kemampuan pelayanan *interest packet* yang datang ke sisi *producer* juga semakin meningkat. Oleh karena itu, nilai *delay* dan *packet drop* semakin menurun, sedangkan nilai *throughput* jaringannya semakin meningkat.
6. Perubahan jumlah *node router* yang semakin meningkat menyebabkan pilihan rute *next hops* yang semakin meningkat pula. Oleh karena itu, nilai *delay* dan *packet drop* semakin meningkat, sedangkan nilai *throughput* jaringannya semakin menurun.
7. Dari analisa yang dilakukan dapat diketahui bahwa algoritma RLB dapat mendukung mekanisme *load balancing* pada jaringan NDN dengan nilai 48,4% lebih baik daripada tanpa menggunakan algoritma tersebut. Algoritma RLB dapat berjalan dengan baik dan menunjukkan hasil performa jaringan yang sesuai dengan perubahan nilai parameter pengujian yang diterapkan.

5.2 Saran

Saran yang dapat diberikan untuk penelitian selanjutnya terkait dengan penelitian Tugas Akhir ini sebagai berikut :

1. Dapat membahas proses sistem penamaan (*prefix*) pada mekanisme *load balancing* di jaringan NDN.

2. Dapat membahas proses *routing* pada mekanisme *load balancing* di jaringan NDN agar meningkatkan potensi kinerja sistem.
3. Dapat membahas proses keamanan jaringan dengan adanya serangan pada sebuah topologi agar mengetahui lebih lanjut performa jaringannya menggunakan algoritma RLB.

Referensi

- [1] ITU-T, "Future networks: Objectives and design goals," *Recomm. ITU-T Y.3001*, 2012, [Online]. Available: <https://www.itu.int/rec/T-REC-Y.3001-201105-I>.
- [2] D. Saxena, V. Raychoudhury, N. Suri, C. Becker, and J. Cao, "Named Data Networking: A survey," *Comput. Sci. Rev.*, vol. 19, pp. 15–55, 2016, doi: 10.1016/j.cosrev.2016.01.001.
- [3] S. Song, T. Lv, and X. Chen, "Load balancing for future internet: An approach based on game theory," *J. Appl. Math.*, vol. 2014, 2014, doi: 10.1155/2014/959782.
- [4] D. Mansour, H. Osman, and C. Tschudin, "Load Balancing in the Presence of Services in Named-Data Networking," *J. Netw. Syst. Manag.*, vol. 28, no. 2, pp. 298–339, 2020, doi: 10.1007/s10922-019-09507-x.
- [5] M. A. Al Fuad, M. S. S. Sabuj, M. Z. Hasan, and M. Naznin, "RLB: Randomized load balanced packet forwarding strategy in name based data networking," *Proc. 2017 Int. Conf. Networking, Syst. Secur. NSysS 2017*, vol. 2018-Janua, pp. 1–4, 2017, doi: 10.1109/NSYSS2.2017.8267798.
- [6] A. Tariq, R. A. Rehman, B. S. Kim, "Forwarding Strategies in NDN based Wireless Networks : A Survey," *IEEE Communications Surveys & Tutorials*, 2019, doi: 10.1109/COMST.2019.2935795
- [7] Named Data Networking: Executive Summary. (n.d.). — <https://named-data.net/project/execsummary/>
- [8] L. Zhang *et al.*, "NDN Project 2010," *Relatório Técnico NDN-0001, Xerox Palo Alto Res. Center-PARC*, vol. 157, no. October, p. 158, 2010, [Online]. Available: <http://www.named-data.net/techreport/TR001ndn-proj.pdf>.
- [9] NDN Packet Format Specification 0.2.1 documentation. (n.d.). *Interest Packet*. Interest Packet — NDN Packet Format Specification 0.2.1 documentation.html.
- [10] NDN Packet Format Specification version 0.3. (n.d.). *Interest Packet*. Interest Packet — NDN Packet Format Specification version 0.3 — <https://named-data.net/doc/NDN-packet-spec/current/interest.html>
- [11] S. Mastorakis, A. Afanasyev, and L. Zhang, "Public Review for On the Evolution of ndnSIM Artifacts Review for On the Evolution of ndnSIM," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 47, no. 3, pp. 19–33, 2017, [Online]. Available: <https://pdfs.semanticscholar.org/a5f4/a1ac695eadea269efc6f7f7a68490add8ef0.pdf>.
- [12] Obtaining metrics. *Packet-level trace helpers*. Packet-level trace helpers — Obtaining metrics — <https://ndnsim.net/2.2/metric.html>.
- [13] "ChristianKreuzberger/amus-ndnSIM: amus-ndnSIM - Adaptive Multimedia Streaming Framework for ndnSIM." <https://github.com/ChristianKreuzberger/amus-ndnSIM>
- [14] G. P. Satriawan, "Analisis Performa Strategi Forwarding Pada Protokol Routing Loop-Free Inport-Dependent (LFID) pada Jaringan Named Data Network (NDN)," Fakultas Teknik Elektro Universitas Telkom, Bandung, 2020.
- [15] NS-3 based Named Data Networking (NDN) simulator. (n.d). ndnSIM: simulation of NDN, ICN, Information-Centric Networking. ndnSIM documentation. *Getting Started* — <https://ndnsim.net/2.8/getting-started.html>