

## **EVALUASI KINERJA TCP NEW RENO DALAM WIRELESS MESH NETWORK**

### **PERFORMANCE EVALUATION OF TCP NEW RENO IN WIRELESS MESH NETWORK**

**Veronica Windha**

**Jurusan Teknik Elektro, Fakultas Teknik  
Universitas Katolik Indonesia Atma Jaya, Jakarta**

#### **Abstrak**

*Wireless Mesh Network (WMN) hadir sebagai bagian dari jaringan wireless dan diharapkan dapat mengatasi masalah-masalah yang ada pada jaringan wireless. Node di dalam WMN terdiri dari mesh router dan mesh client, dimana mesh router mempunyai mobilitas yang minimal dan membentuk backbone dari WMN. Mesh client dapat berupa desktop, laptop, telepon. Node yang berfungsi sebagai mesh clients tersebut akan terkoneksi secara langsung ke wireless mesh router. Dengan fungsi gateway dalam mesh router, memungkinkan node tersebut terkoneksi dengan node jaringan di luar jaringan mesh, seperti Internet. Untuk mendukung koneksi tersebut diperlukan protokol transport. Salah satu protokol transport yang dapat digunakan dalam WMN adalah protokol TCP New Reno. Oleh karena itu, dalam paper ini penulis melakukan uji kinerja dari protokol TCP New Reno dalam WMN dengan pendekatan simulasi menggunakan simulator jaringan NS-2. Indikator kinerja yang dievaluasi adalah throughput dan delay end-to-end. Dari serangkaian simulasi yang dilakukan, terlihat bahwa dengan bertambahnya jumlah koneksi 2, 3, dan 5 maka throughput semakin kecil, sedangkan delay end-to-end semakin besar. Kenaikan mobilitas dapat menaikkan ataupun menurunkan throughput dan delay end-to-end tergantung pergerakan dari node tujuan.*

**Kata kunci:** WMN, TCP New Reno, NS-2, throughput, end-to-end delay

#### **Abstract**

*Wireless Mesh Network (WMN) emerged as part of wireless network and is expected to resolve existing problems in wireless network. Nodes in WMN consists of mesh routers and mesh clients, where the mesh routers have minimal mobility and form the backbone of WMN. Mesh client can be a desktop, laptop, phone. The node which serves as mesh clients will be connected directly to the wireless mesh router. Gateway function in mesh router enables the nodes to connect to network nodes outside network mesh such as internet. Connection are required to support transport protocol. One of possible transport protocols in WMN is TCP New Reno protocol. Thus, in this paper, the authors conducted a performance test of TCP New Reno in WMN using network simulator NS-2 as simulation approach. Evaluated performance indicators are 'throughput' and 'delay end-to-end'. From a series of simulations conducted, it can be observed that with the increasing number of connection 2, 3, and 5, 'throughput' is decreasing, while the 'delay end-to-end' is increasing. The increasing mobility can raise or reduce 'throughput' and 'delay end-to-end', depending on the movement of the destination node.*

**Keywords:** WMN, TCP New Reno, NS-2, throughput, end-to-end delay

Tanggal Terima Naskah : 19 April 2016  
Tanggal Persetujuan Naskah : 29 Agustus 2016

## 1. PENDAHULUAN

*Wireless Mesh Network* (WMN) hadir sebagai bagian dari jaringan *wireless* yang menjanjikan untuk banyak aplikasi, seperti *broadband home networking*, *enterprise networking*, metropolitan area, dan lain-lain. *Node* di dalam WMN, terdiri dari *mesh router* dan *mesh client*. *Node* yang biasa digunakan sebagai *mesh client* adalah *laptop*, *desktop*, telepon terhubung secara langsung ke *wireless mesh router* [1]. Dengan fungsi *gateway* dalam *mesh router* memungkinkan *node* tersebut untuk terkoneksi dengan *node* jaringan di luar jaringan *mesh*, seperti Internet. Jenis informasi yang dapat dikirim melalui WMN juga bermacam-macam, seperti suara, data, dan video dimana untuk pengiriman informasi berupa data, diperlukan TCP yang dapat mendukung koneksi tersebut.

Terdapat banyak protokol TCP yang dapat digunakan dalam WMN, salah satunya adalah protokol TCP *New Reno*. Oleh karena itu, dalam *paper* ini, dilakukan uji kinerja protokol TCP *New Reno* pada WMN dengan konfigurasi *hybrid* WMN. Evaluasi kinerja dilakukan dengan pendekatan simulasi menggunakan NS-2. Indikator kinerja yang dievaluasi adalah *throughput* dan *end-to-end delay*.

## 2. KONSEP DASAR

### 2.1 Arsitektur WMN

Arsitektur WMN dapat dibagi dalam tiga kelompok sebagai berikut [2]:

#### a. Infrastruktur/Backbone WMN

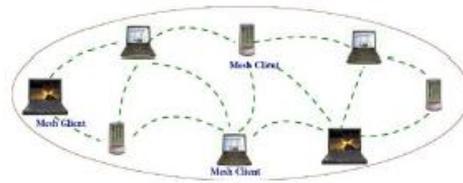
Tipe arsitektur ini meliputi *mesh router* yang saling dikoneksikan dengan menggunakan berbagai macam teknologi radio yang ada untuk membentuk *backbone*, biasanya menggunakan IEEE 802.11. Fungsi *gateway* juga diintegrasikan pada beberapa *router* untuk hubungannya dengan jaringan luar. Tipe arsitektur ini dapat dilihat pada Gambar 1.



Gambar 1. Infrastruktur/Backbone WMN [1]

#### b. Client WMN

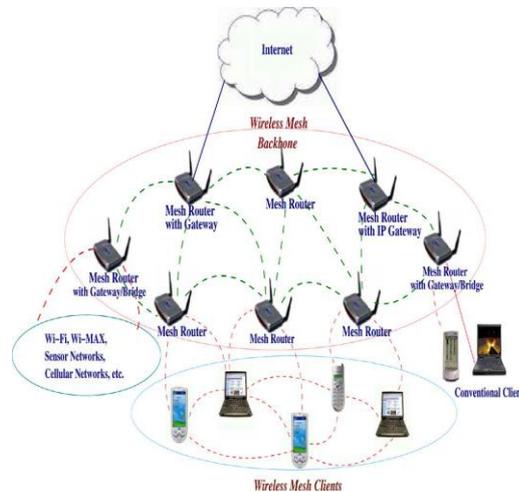
*Client WMN* membentuk *peer to peer* antara *device client*. Pada bentuk ini setiap *node* secara aktif berpartisipasi dalam menjalankan *routing* dan konfigurasi jaringan. Dengan demikian, setiap *node* berfungsi sebagai *host* pengirim/penerima maupun *router* seperti pada Gambar 2.



Gambar 2. Client WMN [1]

### c. Hybrid WMN

Arsitektur ini adalah gabungan dari *infrastructure* dan *client meshing* seperti yang terlihat pada Gambar 3. *Mesh clients* dapat mengakses jaringan melalui *mesh routers* dengan baik jika secara langsung *meshing* dengan *mesh client* yang lain. Meskipun *infrastructure* menyediakan konektivitas ke jaringan lain, seperti internet, Wi-Fi, WiMAX, seluler, dan *sensor networks*; kemampuan *routing* dari *client* memberikan perbaikan konektivitas dan mencakup bagian dalam dari WMN.



Gambar 3. Hybrid WMN [1]

## 2.2 Gateway

Infrastruktur/*backbone* WMN dan *hybrid* WMN menggunakan *gateway* sebagai sebuah pintu masuk ke *wired* Internet. Berikut ini cara-cara untuk mendapatkan informasi tentang *gateway* [3]:

### a. Proactive Gateway Discovery

*Gateway discovery* yang diinisialisasi oleh *gateway* tersebut dengan cara mem-*broadcast* sebuah pesan *gateway advertisement* (GW ADV) secara periodik ke semua *node*. *Node* yang menerima pesan GWADV ini melengkapi tabel *routing* dengan *default route* ke *gateway*.

### b. Reactive Gateway Discovery

*Node* yang memerlukan sebuah rute ke *gateway* akan mem-*broadcast* sebuah RREQ, kemudian *gateway* bereaksi terhadap pesan tersebut dengan mengirim sebuah RREP ke *node* tersebut. *Node* perantara hanya meneruskan RREQ.

### c. Hybrid Gateway Discovery

Cara ini mengkombinasikan kedua cara sebelumnya.

### 2.3 TCP New Reno

Di dalam TCP *New Reno*, mekanisme *fast recovery* dari TCP *Reno* dimodifikasi menurut RFC 2582, yaitu [4],[5]:

- a. Ketika *duplicate* ACK yang ketiga diterima maka nilai *ssthresh* di-*set* tidak lebih dari nilai yang diberikan oleh persamaan berikut:

$$ssthresh = \max \frac{flightsize}{2}, 2MSS \dots\dots\dots(1)$$

Selanjutnya *sequence number* tertinggi yang ditransmisikan dalam variabel “*recover*” dicatat.

- b. Mentransmisikan kembali segmen yang hilang dan men-*set* *cwnd* ke nilai:

$$cwnd = ssthresh + 3 * SMSS \dots\dots\dots(2)$$

- c. Untuk setiap tambahan *duplicate* ACK yang diterima, naikkan nilai *cwnd* dengan *SMSS*.

- d. Mentransmisikan sebuah segmen

- e. Jika ACK berikutnya tiba untuk memberikan *acknowledgement* data baru, maka ACK ini bisa jadi ACK yang diperoleh dari retransmisi yang dilakukan pada langkah b atau dari retransmisi berikutnya.

Jika ACK memberikan *acknowledgement* semua data yang terakhir dan mencakup “*recover*” maka ACK memberikan *acknowledgement* semua *intermediate segment* yang dikirim di antara transmisi segmen yang hilang dan penerimaan ketiga *duplicate* ACK. Men-*set* *cwnd* ke nilai:

1. *min ssthresh, Flightsize + SMSS* atau
2. *ssthresh*

dimana *ssthresh* adalah nilai yang ditentukan pada langkah a, dan ini adalah terminologi pengurangan ukuran *window*. Jika pilihan kedua yang dipilih maka jumlah data yang ada dalam jaringan harus lebih kecil dari nilai *cwnd* yang baru untuk menghindari terjadinya *burst data*. Jika ini semua sudah dilakukan maka keluar dari prosedur *fast recovery*.

## 3. PERANCANGAN SIMULASI

### 3.1 Parameter Simulasi

Parameter yang digunakan dalam simulasi adalah sebagai berikut:

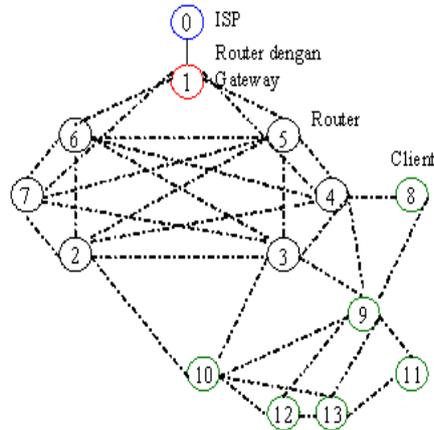
Tipe antrian <i>interface</i>	: <i>DropTail</i>
Model antena	: <i>OmniAntenna</i>
Dimensi <i>topography</i> X	: 950
Dimensi <i>topography</i> Y	: 650
Jumlah <i>wired node</i>	: 1
Jumlah <i>gateway</i>	: 1
Informasi <i>gateway</i>	: <i>Reactive gateway</i>
<i>Max packet</i> dalam antrian	: 50
Protokol <i>routing</i>	: AODV+
Waktu simulasi berhenti	: 300 detik

### 3.2 Skenario Simulasi

Skenario simulasi yang digunakan dalam simulasi ini adalah sebagai berikut:

## a. Simulasi variasi jumlah koneksi

Pada skenario ini, digunakan satu buah *gateway* yang terhubung ke *Internet service Provider* (ISP); tujuh buah *router* termasuk di dalamnya *gateway* yang membentuk *mesh* dan enam buah *client* yang membentuk *mesh*, seperti yang terlihat pada Gambar 4.



Gambar 4. Model skenario simulasi variasi jumlah koneksi

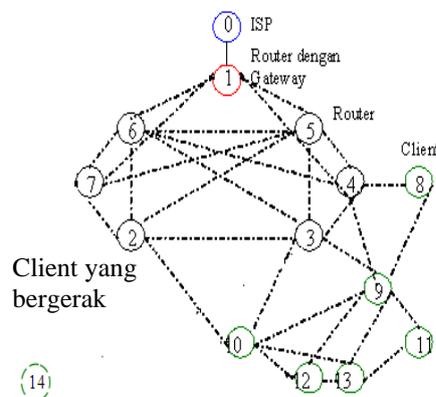
Dalam simulasi ini, *node* pengirim mengirimkan trafik FTP ke *node* tujuan dan jumlah koneksi TCP yang ada dalam jaringan akan divariasikan menjadi dua, tiga, dan lima koneksi dengan tujuan agar terjadi *loss*. Protokol TCP *New Reno* akan dievaluasi kinerjanya dalam mengatasi kondisi *loss* tersebut.

## b. Simulasi variasi mobilitas

Pada skenario ini, akan dilakukan simulasi variasi mobilitas pada *node* tujuan (*node* 14) dengan pergerakan yang *random*. Variasi mobilitas yang diamati adalah untuk *host* yang bergerak, yang dikelompokkan ke dalam tiga kategori, sebagai berikut:

1. *Host* yang bergerak lambat dengan kecepatan 1 dan 2 m/s
2. *Host* yang bergerak lambat dengan kecepatan 3 dan 4 m/s
3. *Host* yang bergerak lambat dengan kecepatan 5 dan 6 m/s

Model mobilitas yang digunakan dalam skenario ini adalah model *Random Waypoint Mobility*, karena jenis model tersebut paling mendekati keadaan jaringan yang sebenarnya. Skenario ini menggunakan tujuh buah *router* termasuk di dalamnya *gateway* yang membentuk *mesh*. *Gateway* terhubung dengan *wired* ke ISP; tujuh buah *client* (enam *client* membentuk *mesh*) dan satu *client* dibuat bergerak, seperti pada Gambar 5.



Gambar 5. Model *Random Waypoint Mobility*

Dalam simulasi ini, *node* pengirim (*node* 0) mengirim trafik FTP ke *node* tujuan (*node* 14).

#### 4. HASIL SIMULASI

##### 4.1 *Throughput*

##### 4.1.1 *Throughput* untuk Skenario Variasi Jumlah Koneksi

*Throughput* rata-rata untuk skenario variasi jumlah koneksi dapat dilihat pada Tabel 1.

Tabel 1. *Throughput* Rata-rata untuk skenario variasi jumlah koneksi

Jumlah Koneksi	<i>Throughput</i> Rata-rata [bps]
2	168687,605
3	125832,405
5	109637,117

Dari Tabel 1 terlihat bahwa dengan bertambahnya jumlah koneksi, *throughput* yang dihasilkan mengalami penurunan yang disebabkan karena *packet loss*. Hal ini dikarenakan, pada *TCP NewReno*, setiap *packet loss* yang terdeteksi selalu diasumsikan karena kongesti pada jaringan sehingga TCP langsung melakukan mekanisme *congestion control* yang berlebihan dalam mengurangi ukuran *window*. Hal ini menyebabkan pengiriman kembali paket yang hilang menjadi lambat, sehingga *throughput* akan kecil. Dengan bertambahnya jumlah koneksi, maka kongesti yang terjadi juga akan semakin besar sehingga *packet loss* akan semakin besar. Kondisi ini akan menyebabkan *throughput* semakin kecil dengan bertambahnya jumlah koneksi.

##### 4.1.2 *Throughput* untuk Skenario Variasi Mobilitas

*Throughput* rata-rata untuk skenario variasi mobilitas dapat dilihat pada Tabel 2.

Tabel 2. *Throughput* rata-rata untuk skenario variasi mobilitas

Kecepatan [m/s]	<i>Throughput</i> Rata-rata [bps]
1	250585,135
2	275312,846
3	262247,528
4	280771,701
5	332642,169
6	350051,080

Dari Tabel 2 terlihat bahwa dengan bertambahnya kecepatan mobilitas *node* 14 dapat meningkatkan dan menurunkan *throughput* rata-rata tergantung dari pergerakan *node* 14, bergerak mendekati *node* pengirim atau tidak.

#### 4.2. *Delay End-to-End*

##### 4.2.1 *Delay End-to-End* untuk Skenario Variasi Jumlah Koneksi

3. *Delay end-to-end* untuk skenario variasi jumlah koneksi dapat dilihat pada Tabel

Tabel 3. *Delay end-to-end* untuk skenario variasi jumlah koneksi

<b>Jumlah Koneksi</b>	<b><i>Delay End-to-End</i> Rata-rata [ms]</b>
2	250.575
3	342.696
5	373.181

Dari Tabel 3 terlihat bahwa dengan bertambahnya jumlah koneksi, *delay end-to-end* akan mengalami peningkatan. Hal ini disebabkan karena adanya *packet loss*.

#### 4.2.2 *Delay End-to-End* untuk Skenario Variasi Mobilitas

4. *Delay end-to-end* untuk skenario variasi jumlah koneksi dapat dilihat pada Tabel

Tabel 4 *Delay end-to-end* untuk skenario variasi mobilitas

<b>Kecepatan [m/s]</b>	<b><i>Delay End-to-End</i> [ms]</b>
1	469,019
2	736,547
3	706,873
4	1759,362
5	630,694

Dari Tabel 4 terlihat bahwa dengan bertambahnya jumlah koneksi, *delay end-to-end* akan mengalami peningkatan. Hal ini disebabkan karena adanya *packet loss*

## 5. KESIMPULAN

Dari hasil simulasi yang telah dilakukan, diperoleh kesimpulan:

- Dengan bertambahnya jumlah koneksi 2, 3, dan 5 koneksi, maka *throughput* semakin kecil, sedangkan *delay end-to-end* semakin besar
- Kenaikan mobilitas dapat menaikkan ataupun menurunkan *throughput* dan *delay end-to-end*, tergantung pergerakan dari *node* tujuan.
- Diperlukan protokol *transport* yang dapat membedakan *packet loss* yang terjadi, baik karena adanya kongesti pada jaringan, pengaruh *Bit Error Rate* (BER) yang tinggi dan adanya kegagalan perutean di tengah-tengah pengiriman paket-paket TCP akibat adanya mobilitas dari *mesh client*.

## REFERENSI

- [1]. Roy, A., Umaphathi. 2014. Study on Resource Allocation in Wireless Mesh Network. *International Journal of Emerging Technology and Advanced Engineering*, Vol 4, Issue 1.

- [2]. Akyildiz, I., Wang, X., Wang, W. 2005. Wireless Mesh Networks: a survey. *Computer Networks* 47.
- [3]. Staub, T. 2004. Implementing a Cooperation and Accounting Strategy For Multi-hop Cellular Networks. Diplomarbeit der Philosophisch-naturwissenschaftlichen Fakultät der Universität Bern.
- [4]. Floyd, S., Henderson, T. 1999. RFC2582-The New Reno Modification to TCP's Fast Recovery Algorithm. <http://www.faqs.org/rfcs/rfc2582.html>.
- [5]. Momaru, B., Copacio, F., Lazar, G., Dobrota, V. Practical Analysis of TCP Implementations: Tahoe, Reno, New Reno. Technical University of Cluj-Napoca.