

## Analisis Performansi Metode Redundant Multipath TCP pada Bufferbloat Channel di Jaringan LTE/Wi-Fi

Adrian Gusti Nurcahyo<sup>1</sup>, Parman Sukarno, S.T, M.Sc., Ph.D.<sup>2</sup>, Aji Gautama Putrada, S.T., M.T.<sup>3</sup>

<sup>1,2,3</sup>Fakultas Informatika, Universitas Telkom, Bandung

<sup>1</sup>adrianurcahyo@student.telkomuniversity.ac.id, <sup>2</sup>psukarno@telkomuniversity.ac.id,

<sup>3</sup>ajigp@telkomuniversity.ac.id

---

### Abstrak

Singlepath TCP adalah salah satu protokol yang umumnya digunakan pada saat pengiriman data, akan tetapi dalam kondisi *bufferbloat*, *throughput* yang dihasilkan singlepath TCP tidak optimal. *Multipath TCP* adalah salah satu teknologi yang menggabungkan beberapa *singlepath TCP* untuk mendapatkan *bandwidth* yang besar demi mencapai *throughput* yang tinggi, untuk mengatasi kerugian *bufferbloat* salah satu metode yang digunakan pada MPTCP adalah metode redundan yaitu menggunakan seluruh antarmuka yang disediakan untuk pengiriman data sekaligus secara bersamaan, serta memanfaatkan ketahanan koneksi walaupun terdapat kerugian pada jalur yang digunakan. Sehingga *throughput* yang di dapatkan tinggi atau paling tidak sama dengan *singlepath TCP*. Melalui simulasi dapat dibuktikan bahwa *Multipath TCP* dapat mengatasi *bufferbloat* pada *singlepath TCP*. Hasil *throughput* yang dihasilkan pada LTE sebesar 1.63 Mbit/s, untuk WiFi sebesar 1.08 Mbit/s sedangkan *Multipath TCP* sebesar 2.59 Mbit/s. Hasil *delay* yang dihasilkan pada LTE sebesar 4.6 ms, pada WiFi sebesar 4.09 ms, dan MPTCP sebesar 3.00 ms.

**Kata kunci :** *Multipath TCP, Bufferbloat, Throughput, Delay.*

---

### Abstract

Singlepath TCP is one of the protocols commonly used when sending data, but under *bufferbloat* conditions, the throughput generated by TCP singlepath is not optimal. *Multipath TCP* is one technology that combines several TCP singlepaths to get large bandwidth in order to achieve high throughput, to overcome *bufferbloat* losses one of the methods used in MPTCP is a redundant method that uses all interfaces provided for data transmission simultaneously, as well as utilize connection resilience even though there are losses on the path used. So that the throughput obtained is high or at least equal to TCP singlepath. Through simulation it can be proved that *Multipath TCP* can overcome *bufferbloat* on *singlepath TCP*. The result of throughput generated in LTE is 1.63 Mbit / s, for WiFi is 1.08 Mbit / s while *Multipath TCP* is 2.59 Mbit / s. The resulting delay in LTE is 4.6 ms, in WiFi is 4.09 ms, and MPTCP is 3.00 ms.

**Keywords:** *Multipath TCP, Bufferbloat, Throughput, Delay.*

---

## 1. Pendahuluan

Saat ini jaringan seluler tersebar luas di masyarakat modern. Kemacetan lalu lintas menjadi masalah yang terus berlanjut setiap tahunnya karena bertambahnya lalu lintas yang terjadi pada jaringan seluler [10]. Antarmuka yang paling umum digunakan pada jaringan seluler adalah jaringan *Wireless Fidelity* (WiFi), 3G dan *Long Term Evolution* (LTE). Pada jaringan seluler, transmisi data yang dilakukan oleh perangkat seluler melalui WiFi maupun LTE menggunakan *Singlepath Transmission Control Protocol* (TCP). Dalam hal ini, protokol *Singlepath TCP* masih memiliki kekurangan dalam hal *throughput* dan *bandwidth* yang masih rendah yang menyebabkan kemacetan lalu lintas pada jaringan seluler bertambah besar [13].

Protokol *transport* seperti TCP hanya melintasi satu rute dengan hanya menggunakan satu antarmuka akses. TCP juga masih memiliki kekurangan dalam kondisi *bufferbloat*. Saat kondisi *bufferbloat*, TCP berpotensi memiliki dampak negatif karena menyamaratakan semua jenis kerugian dalam *round trip time* (RTT) dan *throughput* pada perangkat menyebabkan hasil yang tidak optimal serta penundaan yang berlebihan [8].

*Bufferbloat* adalah suatu keadaan dimana pengisian yang dilakukan terus-menerus yang nantinya buffer akan penuh. Kerugian yang dihasilkan dari *bufferbloat* dapat menghasilkan paket drop yang akan menyebabkan *throughput* yang dihasilkan berkurang atau tidak maksimal [19]. Pada jaringan seluler, TCP memiliki masalah ketika dalam keadaan *bufferbloat*, termasuk *delay* yang lama dan hasil yang tidak optimal.

Multipath TCP (MPTCP) adalah solusi untuk memecahkan masalah diatas. MPTCP diusulkan oleh *Internet Engineering Task Force (IETF)*. *Multipath TCP* memungkinkan koneksi jaringan TCP dengan menggunakan beberapa antarmuka secara bersamaan untuk meningkatkan pemanfaatan sumber daya dan ketangguhan koneksi [3]. *Multipath TCP* meningkatkan dan mengoptimalkan *bandwidth* dan menciptakan banyak *subflows* secara dinamis dan dihapus saat berjalan dengan opsi *singlepath TCP* [5]. Karena MPTCP memaksimalkan dua atau beberapa antarmuka, ukuran *buffer* yang diperoleh akan besar karena akumulasi dari antarmuka yang ada.

Untuk mengatasi masalah diatas, sudah ada beberapa penelitian sebelumnya yang membahas tentang *buffer* pada MPTCP [7] [11] [12]. Dari penelitian [12], merancang metode *Receive Buffer based Path Management (RBPM)* dimana metode ini beroperasi berdasarkan ukuran *buffer* yang tersedia dengan perbedaan karakteristik pada jalur MPTCP. RBPM juga memprediksi *buffer* terlebih dahulu dan berhenti mentransfer pada jalur yang buruk. Namun, simulasi yang dilakukan masih menggunakan jaringan biasa.

Pada penelitian ini, diajukan sebuah metode *redundant* untuk mengetahui performansi dari MPTCP dalam jaringan *bufferbloat*. Metode ini mentransmisikan data pada semua antarmuka jaringan yang tersedia sehingga memaksimalkan semua jaringan pada MPTCP.

Beberapa batas yang terdapat pada tugas akhir ini adalah :

- Penelitian bersifat simulasi.
- Pengukuran hanya pada *throughput* dan *delay*.
- Antarmuka yang digunakan hanya pada jaringan WiFi dan LTE.
- Jarak pada simulasi diabaikan.
- Hanya Menggunakan satu *device* sebagai *client*.
- Pengukuran *bufferbloat* hanya dilihat dari *delay* yang dihasilkan.

Penelitian ini bertujuan untuk menganalisis performansi MPTCP menggunakan metode *redundant* pada jaringan *bufferbloat* di LTE dan WiFi. Performansi yang dimaksud adalah hasil dari MPTCP terhadap LTE dan WiFi dalam jaringan *bufferbloat* dengan tolak ukur pada *throughput* dan *delay*.

Urutan penulisan laporan ini adalah sebagai berikut : Bagian 2 menunjukkan penelitian-penelitian terkait dengan tugas akhir ini. Sistem yang diajukan untuk akan dijelaskan di bagian 3. Pada bagian 4 akan didiskusikan mengenai hasil pengujian dan evaluasi sistem. Akhirnya, kesimpulan akan dipaparkan pada bagian 5.

## 2. Studi Terkait

### 2.1 Studi Literatur

#### 2.1.1 Delay

Delay adalah suatu kondisi dimana suatu paket dikirim melebihi dari jangka waktu normal. Kondisi ini akan menghasilkan angka yang biasa disebut waktu delay. *Delay* juga dapat diartikan sebagai waktu tunda yang disebabkan oleh kepadatan lalu lintas data pada suatu jaringan. Pada kondisi lain, *delay* juga dapat diakibatkan oleh antrian paket yang ada di *router* [18].

#### 2.1.2 Bufferbloat

*Bufferbloat* adalah suatu keadaan dimana pengisian yang dilakukan terus-menerus yang nantinya buffer akan penuh. Kerugian yang dihasilkan dari *bufferbloat* dapat menghasilkan paket drop yang akan menyebabkan *throughput* yang dihasilkan berkurang atau tidak maksimal [19]. Pada jaringan seluler, TCP memiliki masalah ketika dalam keadaan *bufferbloat*, termasuk *delay* yang lama dan hasil yang tidak optimal. Sebagian besar implementasi TCP menggunakan *loss-based congestion control* dimana pengirim tidak akan memperlambat mengirim hingga melihat *packet loss* [9].

### 2.2 Studi Terkait

#### 2.2.1 Management of Buffer Space for the Concurrent Multipath Transfer over Dissimilar Paths

Pada penelitian [7], membahas tentang perpanjangan aliran kontrol protokol transmisi pada *multipath*. *Multipath* menggunakan *concurrent multipath transfer (CMT-SCTP)* sebagai protokol perpanjangan aliran. Pada prosesnya dua jalur tidak sama karena perbedaan karakteristik seperti *delay* dan *bandwidth*. Selain itu, CMT-SCTP

memiliki *buffer* yang sama pada pengirim dan penerima. Tidak peduli berbeda dalam hal kualitas, jalur cepat tidak dapat meningkatkan agregat throughput karena data out-of-sequence pada penerima. Dalam hal ini, *data out-of-sequence* dapat menyebabkan *buffer blocking*. Teknik *state-of-the-art* menunjukkan bahwa *buffer* harus dibagi sesuai dengan jumlah jalur yang ada. Teknik ini memberika ruang *buffer* yang sama pada jalur cepat maupun lambat. Ini juga membatasi kinerja jalur cepat karena *buffer* pada jalur lambat. Maka dari ini penelitian ini mengusulkan teknik CMT-RTTA. Teknik manajemen *buffer* baik pada penerima dan pengirim untuk CMT-SCTP berdasarkan RTT (CMT-RTTA) untuk jaringan dimana jaringan tidak sama. Teknik ini secara implisit mendefinisikan kontrol aliran paket data. Tepatnya, ruang lingkup pekerjaan ini terbatas pada ketidakseimbangan diantara jalur yang disebabkan oleh *bandwidth*. Untuk melengkapi CMT-RTTA, penulis menetapkan prioritas ke jalur yang menggunakan RTT tempat pengirim memberikan tempat yang tersedia untuk transmisi dalam RTT yang sesuai.

### 2.2.2 The Effect of Routing Path Buffer Size On Throughput of Multipath TCP

Pada penelitian [11], meneliti kinerja efek *throughput* yang disebabkan oleh ukuran *buffer* pada *router* dalam jalur transmisi pada *multipath tcp* menggunakan metode *default*. Penulis menyelidiki hubungan antara RTT dan keluar yang bersifat simulasi dimana perangkat memiliki dua jalur transmisi kabel yang terdiri dari *server*, *client*, dan dua *router*. Pada simulasi, memiliki dua skenario untuk mengetahui *throughput* yang dihasilkan. Skenario pertama berdasarkan *delay* yang sudah ditentukan pada *multipath tcp*. Skenario kedua berdasarkan ukuran *buffer* yang sudah ditentukan pada *multipath tcp*. Hasil menunjukkan bahwa *throughput* akan besar ketika MPTCP memiliki *delay* yang rendah di kedua jalur dibandingkan dengan *singlepath TCP*. Sedangkan pada ukuran *buffer* yang sudah ditentukan, *singlepath TCP* lebih baik ketika *buffer* pada MPTCP rendah dan RTT rendah.

### 2.2.3 Receive Buffer based Path Management for MPTCP in Heterogeneous Networks

Pada penelitian [12], merancang metode sederhana dan efektif untuk mengelola MPTCP yang disebut *Receive Buffer based Path Management* (RBPM). Metode ini beroperasi berdasarkan ukuran *buffer* yang tersedia perbedaan karakteristik pada jalur MPTCP. RBPM memperkirakan *out-of-ordered* paket, memprediksi masalah *buffer blocking* terlebih dahulu dan berhenti mentransfer pada jalur yang buruk. RBPM juga memperkirakan jumlah paket yang tidak dipesan dan *buffer* yang diperlukan untuk transmisi dan memprediksi masalah *buffer blocking* terlebih dahulu. Menurut penelitian ini, RBPM mengelola beberapa jalur MPTCP untuk meningkatkan *throughput* dan pemanfaatan jaringan dengan membatasi penerimaan *buffer* dalam jaringan heterogen. Penelitian ini bersifat simulasi. Hasil menunjukkan bahwa metode yang diusulkan secara signifikan meningkatkan *throughput* dan pemanfaatan jaringan dengan *buffer* penerima yang dibatasi.

## 3. Sistem yang Dibangun

*Multipath TCP* pada dasarnya adalah protokol *singlepath TCP* dengan menggunakan dua atau beberapa antarmuka jaringan secara bersamaan dengan memanfaatkan sumber daya dan ketangguhan koneksi. Dengan menggabungkan kedua antarmuka ini, MPTCP dapat memaksimalkan *throughput* yang ada pada setiap antarmuka jaringan sehingga dapat menggunakan kecepatan *transfer* yang besar. MPTCP berasal dari modifikasi *singlepath TCP* yang bekerja pada *transport layer*. MPTCP juga memiliki tiga metode dengan cara kerja yang berbeda-beda [17], yaitu:

1. Metode *default*: dua antarmuka aktif tetapi satu antarmuka sebagai cadangan ketika antarmuka *default* terjadi *loss* atau *congestion*.
2. Metode *round robin*: menggunakan dua antarmuka aktif dengan pengiriman secara bergantian pada setiap paket. Metode ini memiliki kinerja yang buruk dan pengiriman yang lambat karena perpindahan jalur yang selalu bergantian.
3. Metode *redundant*: menggunakan dua antarmuka aktif dengan pengiriman sekaligus secara bersamaan.

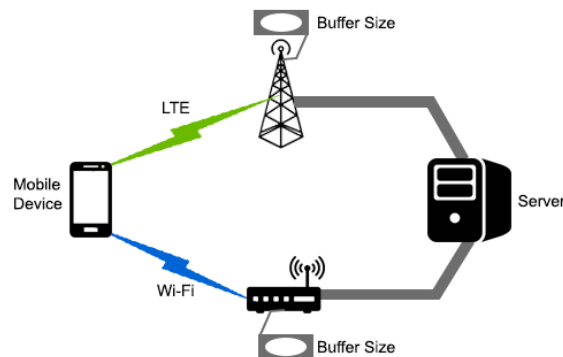
Dari ketiga metode diatas, dalam penelitian ini menggunakan metode *redundant* dengan memaksimalkan antarmuka yang ada untuk digunakan dalam mengatasi terjadinya paket *drop/bufferbloat* [4]. Kedua metode lainnya (*default* dan *round robin*) biasa digunakan untuk penelitian lain, salah satu contohnya adalah konsumsi energi dari MPTCP menggunakan metode *default* [14] [1].

Sistem yang digunakan dalam penelitian ini menggunakan metode *redundant* pada *Multipath TCP*. Metode *redundant* memungkinkan penggunaan semua antarmuka yang ada sekaligus secara bersamaan sehingga memaksimalkan antarmuka yang ada. Gambaran umum sistem terdapat pada Gambar 1.



Gambar 1. Gambaran Umum Sistem

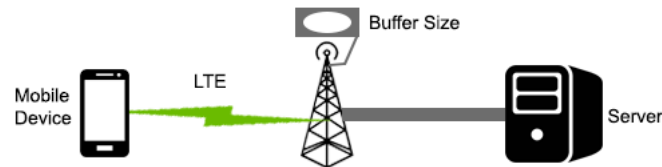
Pada penelitian ini bersifat simulasi menggunakan *Network Simulator 3 DCE (NS-3 DCE)* [15]. Dimulai dengan menentukan parameter-parameter seperti *bandwidth*, ukuran *buffer*, dan *delay*. Kemudian simulasi dilakukan menggunakan metode *redundant* yaitu metode yang menggunakan semua jalur yang ada sekaligus pada NS-3 DCE. Hasil dari simulasi di analisis dengan membandingkan antara MPTCP *redundant* dengan *singlepath* TCP apakah MPTCP dapat dengan baik mengatasi paket *drop/bufferbloat*. Setelah di analisis, evaluasi dilakukan dari hasil perbandingan pada tahap sebelumnya.



Gambar 2. Topologi MPTCP

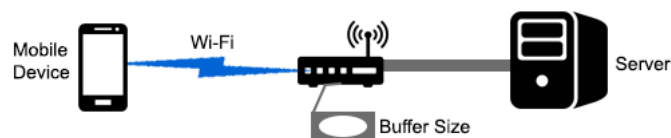
Pada topologi diatas (Gambar 2), MPTCP menggunakan dua antarmuka jaringan dengan menggunakan antarmuka LTE dan WiFi karena ini adalah skenario umum pada MPTCP [3] dan sudah banyak dilakukan pada studi

sebelumnya [1] [14] [9] [2]. Cara kerja dari topologi diatas adalah paket dikirim dari *client* melewati jaringan LTE dan WiFi kemudian diteruskan ke *server* menggunakan *point-to-point*. Dalam penelitian ini, kondisi ukuran *buffer* sudah ditentukan pada *router* agar dapat mengetahui dengan menggunakan metode *redundant* MPTCP dapat mengurangi paket *drop/bufferbloat* pada *singlepath* TCP.



**Gambar 3.** Topologi LTE

Gambar 3 merupakan topologi dari *singlepath* TCP pada antarmuka LTE. Cara kerja topologi diatas adalah paket dikirim dari *client* melewati eNB (*base station*) kemudian dikirim ke *server* menggunakan *point-to-point/LAN*. Kondisi ukuran *buffer* sudah ditentukan pada eNB/*base station* untuk mengetahui hasil *throughput* dan *delay* pada LTE.



**Gambar 4.** Topologi WiFi

Gambar 4 merupakan topologi dari *singlepath* TCP pada antarmuka WiFi. Cara kerja topologi diatas adalah paket yang dikirim dari *client* akan melewati *router* yang kemudian dikirim ke *server* melewati *point-to-point/LAN*. Kondisi ukuran *buffer* yang sudah ditentukan pada *router* untuk mengetahui hasil *throughput* dan *delay* pada WiFi. Dari ketiga antarmuka diatas, terdapat tiga skenario dalam penelitian ini, yaitu:

- Skenario Pertama: Mengetahui hasil *throughput* dan *delay* dengan berbagai ukuran *buffer* pada antarmuka LTE. Diawali dengan menentukan parameter pengujian yang akan dilakukan. Kemudian mengatur ukuran *buffer* sesuai dengan parameter yang sudah ditentukan pada jaringan LTE yaitu 25000, 50000, 75000, 116736 dalam bentuk *byte*. Waktu simulasi selama 45 detik. Kemudian hasil yang didapat berupa pcap dilihat *throughput* dan *delay* nya.
- Skenario kedua: Mengetahui hasil *throughput* dan *delay* dengan berbagai ukuran *buffer* pada antarmuka WiFi. Pada antarmuka ini, pengujian dilakukan dengan menggunakan berbagai ukuran *buffer*. Ukuran

*buffer* tersebut adalah 25000, 50000, 75000, 116736 dalam bentuk *byte*. Waktu dalam simulasi selama 45 detik. Hasil dari simulasi berupa pcap untuk mengetahui hasil *throughput* dan *delay* dari antarmuka WiFi.

- Skenario ketiga: Mengetahui hasil *throughput* dan *delay* dengan berbagai ukuran *buffer* pada antarmuka MPTCP. Pada antarmuka MPTCP, simulasi dilakukan dengan berbagai ukuran *buffer*. Ukuran *buffer* nya adalah 25000, 50000, 75000, 116736. Waktu simulasi selama 45 detik. Hasil dari simulasi berupa pcap untuk mengukur *throughput* dan *delay* yang dihasilkan dari berbagai ukuran *buffer* dan antarmuka MPTCP.

Penggunaan tiga skenario ini (ditunjukkan pada Tabel 1) bertujuan untuk membandingkan performansi dari MPTCP dengan antarmuka LTE dan WiFi sehingga dapat mengetahui seberapa besar pengaruh MPTCP terhadap LTE dan WiFi dalam keadaan *bufferbloat*. Pengukuran *throughput* menggunakan ukuran *buffer* 25000, 50000, 75000, 116736. Pengukuran dengan ukuran *buffer* tersebut sesuai dengan penjelasan pada [16] bahwa ukuran *default buffer* pada *router* dengan linux kernel adalah 65536. Maka dari itu, penelitian ini menggunakan ukuran *buffer* dibawah 65536. Pengukuran parameter pada pengujian ini menggunakan tools iperf yang hasilnya akan berupa file pcap. Untuk jenis paket yang dikirimkan adalah paket TCP.

## 4. Evaluasi

Bagian ini berisi tiga sub-bagian, yaitu Parameter Pengujian, Hasil Pengujian, dan Analisis Hasil Pengujian. Pengujian dan analisis yang dilakukan selaras dengan tujuan Tugas Akhir sebagaimana dinyatakan dalam Pendahuluan.

### 4.1 Parameter Pengujian

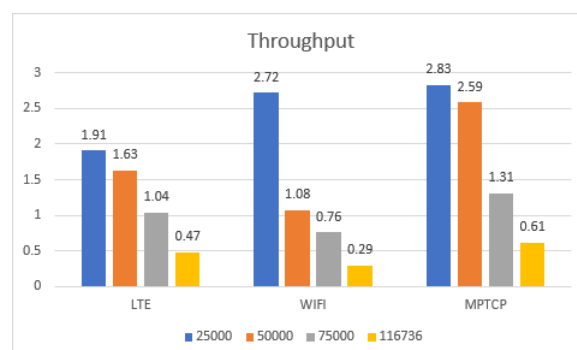
Tabel 1. Parameter Simulasi

	LTE	WiFi	MPTCP
Ukuran <i>Buffer</i>	25000 dan 116736	25000 dan 116736	25000 dan 116736
<i>Bandwidth</i>	100 Mbps	100 Mbps	100 Mbps
Waktu Simulasi	45 detik	45 detik	45 detik

Parameter yang sudah ditentukan akan dianalisis dari ketiga antarmuka (LTE, WiFi, MPTCP) dengan berbagai ukuran *buffer* (25000, 50000, 75000, 116736) untuk mendapatkan berbagai acuan dalam analisa performansi dari ketiga antarmuka yang sudah ditentukan pada Tabel 1.

### 4.2 Hasil Pengujian

#### 4.2.1 *Throughput*

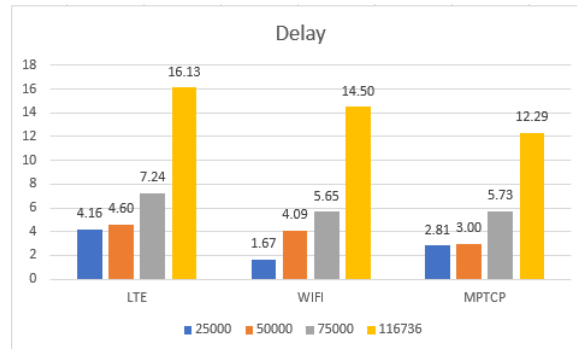


Gambar 5. Hasil *Throughput* Berdasarkan Ukuran *Buffer*

Hasil pengujian *throughput* di antarmuka yang berbeda pada ukuran *buffer* 50000 yaitu pada antarmuka LTE sebesar 1.63 Mbit/s sedangkan pada WiFi sebesar 1.08 Mbit/s dan pada MPTCP sebesar 2.59 Mbit/s. Berdasarkan grafik pada Gambar 5, ukuran *buffer* mempengaruhi *throughput* yang dihasilkan. Perolehan *throughput* pada MPTCP lebih besar dari kedua antarmuka *singlepath* TCP. Tetapi untuk ukuran *buffer* yang besar, *throughput* yang

dihasilkan menurun dibandingkan dengan ukuran buffer kecil. Jika dibandingkan dengan *singlepath* TCP, hasil throughput MPTCP masih diatas dari yang lain. Ini sesuai dengan teori bahwa MPTCP memaksimalkan *throughput* dengan menggunakan dua antarmuka jaringan dan menggunakan metode *redundant* dengan memaksimalkan kedua antarmuka untuk mengirim sekaligus dan dalam keadaan *bufferbloat* throughput yang dihasilkan tinggi.

#### 4.2.2 Delay



**Gambar 6.** Hasil Delay Berdasarkan Ukuran *Buffer*

Hasil pengujian *delay* pada Gambar 6 di antarmuka yang berbeda pada ukuran *buffer* 50000 yaitu pada antarmuka LTE memiliki *delay* sebesar 4.6 ms sedangkan pada antarmuka WiFi sebesar 4.09 ms dan pada antarmuka MPTCP sebesar 3.00 ms. Berdasarkan grafik pada Gambar 6, menunjukkan bahwa ukuran *buffer* sangat mempengaruhi *delay* yang dihasilkan. Perolehan *delay* pada MPTCP lebih kecil dari kedua antarmuka lainnya pada *singlepath* TCP. Tetapi untuk ukuran *buffer* yang besar, *delay* yang dihasilkan besar juga. Jika dibandingkan dengan *singlepath* TCP, *delay* yang dihasilkan lebih kecil dari yang lain. Ini sesuai dengan teori bahwa dengan menggunakan dua antarmuka jaringan sekaligus akan memperkecil *delay* sehingga MPTCP mencerminkan peningkatan daripada *singlepath* TCP.

Pengukuran *bufferbloat* semestinya diukur pada *buffer* nya dilihat dari antrian yang diakibatkan ukuran *buffer* yang besar. Pada penelitian ini, pengukuran *bufferbloat* yang dihasilkan hanya berdasarkan gejalanya saja yaitu diukur dari hasil *delay* pada setiap antarmuka. Seperti menurut [6] bahwa *bufferbloat* terjadi karena keadaan *buffer* yang besar sehingga *delay* yang dihasilkan akan semakin besar juga. Maka dari itu analisis yang dilakukan diukur dari *delay* yang dihasilkan.

#### 4.3 Analisis Hasil Pengujian

Dari hasil pengujian diatas, ukuran *buffer* sangat berpengaruh pada *throughput* dan *delay*. Pada teori *bufferbloat*, ukuran *buffer* yang semakin besar akan menghasilkan *delay* yang semakin besar juga karena antrian yang ditimbulkan pada *router*. Dalam hal ini membuat kinerja dari jaringan akan menurun. Dari masalah ini, MPTCP *redundant* sudah terbukti memperbaiki kinerja pada jaringan *lossy*. Setelah diterapkan pada jaringan *bufferbloat*, terbukti MPTCP dapat menurunkan *delay* yang ditimbulkan pada *router* akibat ukuran *buffer* tinggi. Namun kelemahan kinerja tidak sebaik pada *buffer* kecil.

Untuk perbandingan LTE, WiFi, dan MPTCP. *Throughput* yang dihasilkan pada jaringan MPTCP lebih besar dibandingkan dengan yang lain karena menggunakan dua antarmuka jaringan sekaligus secara bersamaan sesuai dengan metode *redundant*. Metode *redundant* sangat berpengaruh dalam hal *throughput* karena memaksimalkan kedua antarmuka tersebut dalam pengiriman data. Ini membuktikan bahwa jaringan MPTCP dengan metode *redundant* memberikan solusi kepada jaringan *singlepath* TCP dalam keadaan *bufferbloat* dengan memberikan hasil *throughput* diatas *singlepath* TCP dan *delay* dibawah jaringan *singlepath* TCP.

## 5. Kesimpulan

### 5.1 Kesimpulan

MPTCP dengan metode *redundant*, penggunaan dua antarmuka jaringan LTE dan WiFi sekaligus secara bersamaan memberikan efek yang signifikan pada jaringan *bufferbloat* dibandingkan dengan *singlepath* TCP.

Dalam penelitian ini, hasil pengujian dari ketiga skenario pada penelitian ini dapat ditarik kesimpulan bahwa ukuran *buffer* yang sudah ditentukan sangat berpengaruh pada hasil *throughput* dan *delay*. Untuk ketiga antarmuka (LTE, WiFi, MPTCP), pada ukuran *buffer* yang sama menunjukkan bahwa MPTCP dengan metode *redundant* lebih baik dalam hal *throughput* dan *delay* yang dihasilkan dari *singlepath* TCP. Hasil *throughput* yang dihasilkan pada LTE sebesar 1.63 Mbit/s, untuk WiFi sebesar 1.08 Mbit/s sedangkan *Multipath* TCP sebesar 2.59 Mbit/s. Juga pada *delay* yang dihasilkan pada LTE sebesar 4.6 ms, untuk antarmuka WiFi sebesar 4.09 ms, dan untuk antarmuka MPTCP sebesar 3.00 ms.

## 5.2 Saran

Di penelitian selanjutnya, peneliti akan melakukan penambahan parameter pengujian seperti *congestion control* dan konsumsi energi yang dihasilkan dari MPTCP agar penelitian tentang *bufferbloat* MPTCP semakin baik di masa mendatang.

## Daftar Pustaka

- [1] S. Chen, Z. Yuan, and G.-M. Muntean. An energy-aware multipath-tcp-based content delivery scheme in heterogeneous wireless networks. In *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*, pages 1291–1296. IEEE, 2013.
- [2] S. Deng, R. Netravali, A. Sivaraman, and H. Balakrishnan. Wifi, lte, or both?: Measuring multi-homed wireless internet performance. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 181–194. ACM, 2014.
- [3] P. Dong, J. Wang, J. Huang, H. Wang, and G. Min. Performance enhancement of multipath tcp for wireless communications with multiple radio interfaces. *IEEE Transactions on Communications*, 64(8):3456–3466, 2016.
- [4] S. Ferlin-Oliveira, T. Dreiholz, and Ö. Alay. Tackling the challenge of bufferbloat in multi-path transport over heterogeneous wireless networks. In *Quality of Service (IWQoS), 2014 IEEE 22nd International Symposium of*, pages 123–128. IEEE, 2014.
- [5] A. Frommgen, T. Erbschäuber, A. Buchmann, T. Zimmermann, and K. Wehrle. Remp tcp: Low latency multipath tcp. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–7. IEEE, 2016.
- [6] J. Gettys and K. Nichols. Bufferbloat: Dark buffers in the internet. *Queue*, 9(11):40, 2011.
- [7] I. A. Halepoto, F. C. Lau, and Z. Niu. Management of buffer space for the concurrent multipath transfer over dissimilar paths. In *Digital Information, Networking, and Wireless Communications (DINWC), 2015 Third International Conference on*, pages 61–66. IEEE, 2015.
- [8] H. Im, C. Joo, T. Lee, and S. Bahk. Receiver-side tcp countermeasure to bufferbloat in wireless access networks. *IEEE Transactions on Mobile Computing*, 15(8):2080–2093, 2016.
- [9] H. Jiang, Y. Wang, K. Lee, and I. Rhee. Drwa: A receiver-centric solution to bufferbloat in cellular networks. *IEEE Trans. Mob. Comput.*, 15(11):2719–2734, 2016.
- [10] F. Kaup, M. Wichtlhuber, S. Rado, and D. Hausheer. Can multipath tcp save energy? a measuring and modeling study of mptcp energy consumption. In *Local Computer Networks (LCN), 2015 IEEE 40th Conference on*, pages 442–445. IEEE, 2015.
- [11] H. Kim and S. Choi. The effect of routing path buffer size on throughput of multipath tcp. In *Information and Communication Technology Convergence (ICTC), 2016 International Conference on*, pages 1261–1263. IEEE, 2016.
- [12] J. Kim, B.-H. Oh, and J. Lee. Receive buffer based path management for mptcp in heterogeneous networks. In *Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on*, pages 648–651. IEEE, 2017.
- [13] J. S. Lee and J. Lee. Multipath tcp performance improvement in mobile network. In *Ubiquitous and Future Networks (ICUFN), 2015 Seventh International Conference on*, pages 710–714. IEEE, 2015.



- [14] Y.-s. Lim, Y.-C. Chen, E. M. Nahum, D. Towsley, R. J. Gibbens, and E. Cecchet. Design, implementation, and evaluation of energy-aware multi-path tcp. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, page 30. ACM, 2015.
- [15] NS-3. Network Simulator 3 DCE, year = 2018, url = <https://www.nsnam.org/overview/projects/direct-code-execution/>, access date = 2018-15-4.
- [16] T. Sparks. How To: Network / TCP / UDP Tuning, year = 2003, url = [https://wwwx.cs.unc.edu/sparkst/howto/network\\_tuning.php](https://wwwx.cs.unc.edu/sparkst/howto/network_tuning.php), accessdate = 2018-19-8.
- [17] UCLouvain. MultiPath TCP - Linux Kernel implementation, year = 2018, url = <https://multipath-tcp.org/pmwiki.php/Users/ConfigureMPTCP>, access date = 2018-28-5.
- [18] J. Wang, J. Wen, C. Li, Z. Xiong, and Y. Han. Dc-vegas: a delay-based tcp congestion control algorithm for datacenter applications. *Journal of Network and Computer Applications*, 53:103–114, 2015.
- [19] M. Zhang, M. Mezzavilla, J. Zhu, S. Rangan, and S. Panwar. The bufferbloat problem over intermittent multi-gbps mmwave links. *arXiv preprint arXiv:1611.02117*, 2016.

## Lampiran