

# Analisis Penggunaan Algoritma *Useless Packet Transmission Avoidance* (UPTA) untuk Menghindari Transmisi Paket Tidak Berguna pada Multimedia di Jaringan dengan Tingkat *Best-Effort*

Yazid Herdianto, Wahyu Suadi dan Henning Titi Ciptaningtyas

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

E-mail: wahyu@if.its.ac.id

**Abstrak** - *Useless packet transmission* (UPT) adalah transmisi paket multimedia yang terjadi ketika nilai *packet loss rate* dari koneksi multimedia tersebut melebihi batas toleransi (*threshold*) yang bisa diterima untuk menghasilkan kualitas media yang baik. UPT bersifat merugikan karena *bandwidth* yang digunakan oleh UPT tersebut menjadi terbuang sia-sia. Pada jaringan dengan layanan *best-effort*, ketika trafik pada *router* sangat padat, nilai *packet loss rate* bisa sangat besar hingga melebihi *threshold*. Sebagai akibatnya, UPT sangat rentan terjadi. Pada penelitian ini dilakukan analisis terhadap manfaat penggunaan algoritma *useless packet transmission avoidance* (UPTA) dalam mengatasi masalah UPT. Dengan menggunakan OPNET, dilakukan simulasi untuk mengevaluasi performa dari UPTA menggunakan parameter-parameter uji yaitu *throughput*, waktu *upload*, *packet drop*, dan *packet loss*. Analisis dilakukan pada jaringan yang mengimplementasikan WFQ (*Weighted Fair Queueing*) sebagai algoritma *fair packet queueing*. Hasil evaluasi dari beberapa parameter pada simulasi menunjukkan bahwa dengan mengimplementasikan algoritma UPTA pada *router*, UPT dapat dihilangkan sehingga meningkatkan *throughput* hingga 1,9% dan mengurangi waktu *upload* hingga 1,2%.

**Kata Kunci** - MPEG-2, OPNET, *Useless Packet Transmission*, *Useless Packet Transmission Avoidance*, *fair queueing*, WFQ.

## I. PENDAHULUAN

Dalam jaringan internet dikenal adanya trafik adaptif dan nonadaptif. Trafik adaptif adalah trafik yang bisa menyesuaikan tingkat pengiriman datanya (*data rate*) sesuai dengan tingkat kepadatan *router*. Ketika trafik di *router* sedang padat, *data rate* akan dikurangi agar tidak banyak paket yang hilang. Berbeda dengan trafik adaptif, trafik nonadaptif mengirimkan paket-paketnya dengan *data rate* yang diinginkan tanpa terpengaruh dengan tingkat kepadatan *router*. Dari dua jenis protokol transportasi yang ada di internet, TCP (*Transmission Control Protocol*) bersifat adaptif sedangkan UDP (*User Datagram Protocol*) bersifat nonadaptif [1].

Di *router*, aplikasi multimedia yang berbasis UDP akan bersaing dengan aplikasi lain yang berbasis TCP untuk mendapatkan *bandwidth*. Ketika trafik sangat padat, TCP akan menurunkan *data rate*-nya jika ada paketnya yang dibuang oleh *router*. Sedangkan UDP tidak akan mengurangi *data rate*-nya walaupun ada paketnya yang dibuang oleh *router*. Akibatnya, terjadi pembagian *bandwidth* yang tidak merata pada masing-masing trafik dimana trafik UDP akan mendapatkan *bandwidth* yang lebih besar dari trafik TCP. Pembagian *bandwidth* yang tidak adil ini menyebabkan

munculnya istilah yang dikenal dengan *fairness problem*. Penelitian secara ekstensif telah dilakukan untuk mengatasi *fairness problem*. Beberapa algoritma *fair packet queueing* seperti WFQ (*Weighted Fair Queueing*) dan CSFQ (*Core-Stateless Fair Queueing*) telah diusulkan sebagai solusi yang efektif untuk mengatasi *fairness problem*. Beberapa bahkan telah diaplikasikan oleh vendor *router* pada produk-produknya [1].

Pada *router* yang trafiknya sangat padat, penggunaan algoritma *fair packet queueing* (contohnya WFQ) dapat menimbulkan permasalahan yang disebut dengan *useless packet transmission* (UPT) [1]. Pada algoritma WFQ, trafik dikelompokkan berdasarkan sumbernya masing-masing (per *flow*). Masing-masing *flow* akan mempunyai *queue* sendiri-sendiri dan WFQ akan mengalokasikan *bandwidth* secara merata ke masing-masing *queue*. Alokasi *bandwidth* yang diberikan secara adil oleh WFQ belum tentu sesuai dengan *data rate* dari masing-masing *flow*. Bahkan *flow* dari aplikasi multimedia bisa mendapatkan *bandwidth* yang lebih kecil dari *data rate*-nya. Hal ini akan menyebabkan banyak paket multimedia yang hilang di *router* sehingga meningkatkan *packet loss rate*.

Pada transmisi audio/video, *packet loss rate* harus dijaga supaya tetap dibawah *threshold* agar audio/video hasil transmisi mempunyai kualitas yang baik [2]. UPT terjadi ketika paket multimedia dikirimkan dalam keadaan *packet loss rate* melebihi batas toleransi (*threshold*). Transmisi paket yang terjadi dikatakan *useless* karena paket yang dikirimkan tidak memberikan kontribusi dalam menciptakan media komunikasi yang efektif. UPT sebaiknya dihindari karena *bandwidth* yang digunakan dalam UPT menjadi terbuang percuma.

Algoritma *useless packet transmission avoidance* (UPTA) telah diusulkan untuk mengatasi permasalahan UPT [1]. Pada penelitian ini, dilakukan uji coba untuk mengevaluasi performa UPTA. Uji coba dilakukan dengan melakukan simulasi *video streaming* dan pengiriman data menggunakan FTP (*File Transfer Protocol*) melalui *router* yang mengaplikasikan algoritma WFQ.

Pada bagian 2 akan dibahas mengenai permasalahan UPT dan pada bagian 3 akan dibahas tentang algoritma UPTA. Pada bagian 4 akan dibahas tentang metode analisis mulai dari simulasi hingga parameter yang akan dievaluasi, sedangkan hasilnya akan dibahas pada bagian 5. Kesimpulan akan dibahas pada bagian 6.

II. PERMASALAHAN USELESS PACKET TRANSMISSION

Di bawah ini merupakan asumsi dan hasil observasi terhadap aplikasi multimedia pada jaringan *best-effort* yang menjadi dasar untuk mendefinisikan permasalahan UPT [3]:

- Pada pengiriman paket-paket audio/video melalui jaringan, ketika *packet loss rate* melebihi batas toleransi (*threshold*), kualitas audio/video yang diterima menjadi sangat rendah (*unintelligible*) dan komunikasi data menjadi tidak efektif. Nilai *threshold* untuk aplikasi yang satu dengan yang lainnya bisa berbeda-beda. Selama *packet loss rate* masih dibawah *threshold*, audio/video dianggap masih *intelligible*.
- Jika aplikasi multimedia ditransmisikan melalui jaringan *best-effort* (tidak ada jaminan QoS), *packet loss rate* terkadang bisa melebihi *threshold*. Oleh karena itu, dalam transmisi audio/video, keseluruhan waktu yang diperlukan untuk mengirimkan audio/video bisa dideskripsikan sebagai urutan waktu yang terdiri dari interval *unintelligible* dan interval *intelligible*.

Dengan persamaan (1) dapat ditentukan nilai rasio *unintelligible*, yaitu perbandingan antara waktu interval *unintelligible* dengan keseluruhan waktu koneksi [3].

$$W = \frac{\Delta_U}{\Delta_I + \Delta_U} \tag{1}$$

Dimana :

- $U$  : interval *unintelligible*
- $I$  : interval *intelligible*
- $U$  : durasi interval  $U$
- $I$  : durasi interval  $I$

UPT adalah transmisi paket yang terjadi pada interval  $U$ . Transmisi paket-paket tersebut menyebabkan *bandwidth* terbuang secara percuma karena tidak memberikan kontribusi dalam komunikasi data yang bermanfaat bagi penerima. Jumlah *bandwidth* yang terbuang dapat dihitung dengan persamaan (2) [3].

$$\check{S} = y \frac{\Delta_U}{\Delta_U + \Delta_I} = yW \tag{2}$$

adalah nilai rata-rata *throughput* pada interval  $U$  dalam *bits per second* (bps).

III. ALGORITMA USELESS PACKET TRANSMISSION AVOIDANCE

Ada dua cara untuk menghindari UPT pada aplikasi multimedia. Yang pertama yaitu dengan menghindari terjadinya interval  $U$ , dan yang kedua yaitu dengan menghilangkan UPT pada interval  $U$  [3]. Cara pertama bisa dicapai dengan memberikan jaminan *quality of service* (QoS) pada aplikasi multimedia. Pada jaringan *best-effort*, dimana interval  $U$  terkadang tidak bisa kita hindari, algoritma UPT avoidance (UPTA) digunakan untuk menghilangkan UPT dari interval  $U$  tanpa mengubah nilai rasio *unintelligible* ( $w$ ). Dengan menghilangkan UPT, UPTA akan mencegah adanya *bandwidth* yang terbuang sehingga memberikan kesempatan pada aplikasi lain (contohnya aplikasi yang berbasis TCP) untuk mendapatkan lebih banyak *bandwidth*.

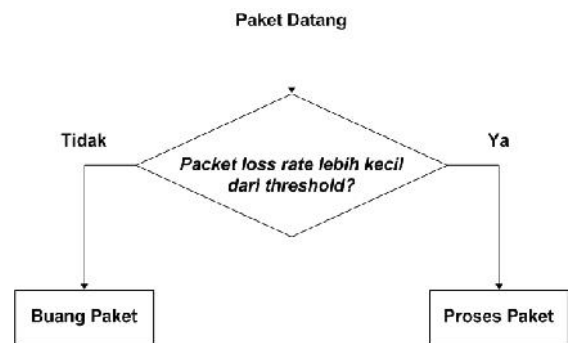
Pada gambar 1 ditunjukkan *flow chart* dari algoritma UPTA [3]. Dari *flow chart* dapat dilihat bahwa UPTA menghitung nilai *packet loss rate* setiap kali paket masuk ke *router*. Nilai *packet loss rate* kemudian akan dibandingkan

dengan nilai *threshold*. Jika *packet loss rate* melebihi *threshold*, maka paket akan dibuang oleh UPTA. Sebaliknya, jika *packet loss rate* kurang dari *threshold*, paket akan diproses secara normal. Pada UPTA, nilai *packet loss rate* didapatkan dari hasil perhitungan *packet arrival rate* dan *link fairshare*. *Packet arrival rate* adalah nilai estimasi rata-rata kedatangan paket per satuan waktu dan *link fairshare* adalah estimasi besaran *bandwidth* yang dialokasikan. Jika nilai *packet arrival rate* dan *link fairshare* diketahui, maka nilai *drop probability* (nilai *packet loss rate* ketika paket masuk ke *router*) bisa diketahui.

Estimasi nilai *packet arrival rate* ( $\lambda$ ) dihitung menggunakan *exponential averaging* dengan persamaan [3]:

$$\lambda_i^{new} = (1 - e^{-T/K}) \frac{l}{T} + e^{-T/K} \lambda_i^{old} \tag{3}$$

$l$  adalah besar ukuran paket,  $T$  adalah nilai *inter-arrival time* dari paket untuk *flow*  $i$  ( $T = t_{new} - t_{new-1}$ ), dan  $K$  adalah konstanta. Pada penelitian ini, sesuai dengan rekomendasi [3],  $K$  diberi nilai 100 ms.



Gambar 1. Flow chart algoritma UPTA

UPTA mengestimasi nilai *fairshare* menggunakan pendekatan *max-min fairness* [4]. Pada *max-min fairness*, *flow* yang mengirimkan paket dengan ukuran yang besar atau *flow* dengan *data rate* tinggi (memerlukan *bandwidth* yang sangat besar) akan mendapatkan *bandwidth* yang terbatas sehingga tidak merugikan *flow* lain. Bahkan *flow* tersebut akan mengalami kerugian karena *bandwidth* yang didapatkan lebih kecil dari yang diperlukan.

Perhatikan notasi-notasi di bawah ini:

- $C$  : kapasitas *output*.
- $N$  : jumlah *flow*.
- $n$  : *flow* ( $n = 1, 2, \dots, N$ ).
- $S$  : jumlah *unconstrained flow* ( $\lambda_i < C/N, i = 1, \dots, S$ ).
- $M$  : jumlah *flow* pada interval  $U$ .
- $m$  : *flow* pada interval  $U$  ( $m = 1, 2, \dots, M$ ).
- $\Lambda$  : vektor *arrival rate* dari semua *flow* ( $\Lambda = \lambda_1, \lambda_2, \dots, \lambda_N$ ).
- $r$  : *max-min fairshare* untuk *flow* pada interval  $U$ .
- $r'$  : *max-min fairshare* untuk *flow* pada interval  $I$ .

Diasumsikan pada vektor  $\Lambda$ , elemen disusun dari nilai yang terkecil ke yang terbesar,

$$\Lambda = \{ \lambda_1, \lambda_2, \dots, \lambda_{S-1}, \lambda_S, \lambda_{S+1}, \lambda_{S+2}, \dots, \lambda_{N-1}, \lambda_N \} \tag{4}$$

maka,  $r$  dan  $r'$  bisa didapatkan dengan persamaan [3]:

$$r = \frac{C - \sum_{i=1}^S \lambda_i}{N - S} \tag{5}$$

$$r' = \frac{C - \sum_{i=1}^S \}i}{N - S - M} \quad (6)$$

Untuk *flow* pada interval  $U$  ( $i \in m$ ), nilai *drop probability* didapatkan dengan persamaan [3]:

$$p_i = \max\left(0, 1 - \frac{r}{\}i}\right) \quad (i \in m) \quad (7)$$

Untuk *flow* pada interval  $I$  ( $i \notin m$ ), nilai *drop probability* didapatkan dengan persamaan [3]:

$$p_i = \max\left(0, 1 - \frac{r'}{\}i}\right) \quad (i \notin m) \quad (8)$$

Jika nilai  $p_i$  lebih besar dari *threshold* maka *flow*  $i$  adalah *flow* yang *unintelligible*.

IV. IMPLEMENTASI DAN UJI COBA

Pada penelitian ini, OPNET digunakan untuk memodelkan topologi jaringan dan mensimulasikan *video streaming* serta pengiriman data melalui *router* yang mengaplikasikan WFQ dan UPTA. Namun, sebelumnya juga dilakukan eksperimen untuk menentukan nilai *threshold* dari *packet loss rate* pada video yang akan dikirimkan pada simulasi *video streaming*.

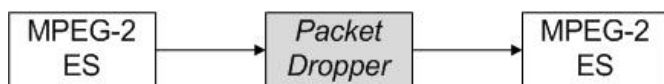
A. Penentuan Nilai Threshold dari Packet Loss Rate

Video yang digunakan pada simulasi ini adalah video *elementary stream* (ES) MPEG-2 yang berjudul *mobl\_120.m2v*. Parameter dari video ini diunjukkan oleh tabel 1. Komponen sistem eksperimen yang digunakan untuk menentukan *threshold* dari *packet loss rate* untuk *mobl\_120.m2v* ditunjukkan pada gambar 2.

Pada gambar 2 ditunjukkan bahwa sistem terdiri atas komponen utama yaitu *packet dropper*. *Packet dropper* yang mensimulasikan pengiriman paket melalui jaringan, membagi-bagi ES menjadi paket-paket berukuran 184 byte dan memberikan nilai *drop probability*  $p$  ( $0 < p < 1$ ) secara acak pada masing-masing paket. Paket-paket ES yang nilai  $p$ -nya melebihi *threshold* dibuang oleh *packet dropper*.

Tabel 1. Parameter dari video MPEG-2 *mobl\_120.m2v*

Parameter	Nilai
Bit Rate (Mbps)	12
Frame Rate (fps)	25
Frame Size (pixels)	704 x 576



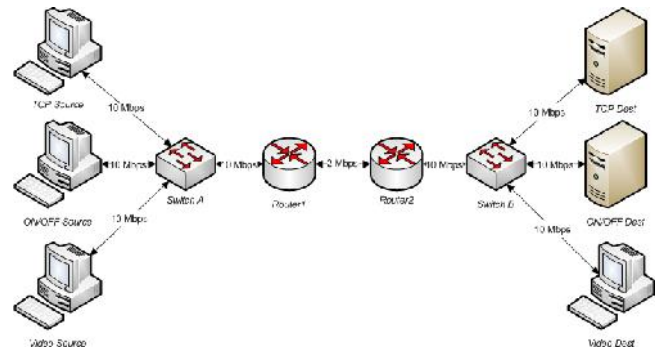
Gambar 2. Sistem eksperimen penentuan *threshold*

Pada eksperimen ini, *packet dropper* diimplementasikan dengan bahasa C. Eksperimen dilakukan sebanyak 20 kali dengan menggunakan nilai *threshold* yang berbeda-beda mulai dari 1% hingga 20%. Melalui serangkaian percobaan didapatkan bahwa pada uji coba dengan *threshold* kurang dari 12%, video ES yang dihasilkan oleh *system decoder* masih memiliki kualitas yang cukup baik. Dari sini dan berdasarkan uji coba pada [3] ditentukan bahwa nilai *threshold* yang akan digunakan pada algoritma UPTA adalah 12%.

B. Simulasi dan Parameter Uji

Topologi jaringan yang dimodelkan menggunakan OPNET dapat dilihat pada gambar 3. Transmisi dilakukan antara dua *subnet* yang dipisahkan oleh dua *router*. Konektor antara tiga *workstation* (sebagai pengirim data) dengan Router1 adalah kabel *ethernet* 10BaseT. Tiga *workstation* lainnya sebagai penerima data terhubung dengan Router2 juga dengan menggunakan konektor 10BaseT. *Buffer* pada Router1 diatur sehingga mempunyai kapasitas sebesar 65 KB. Router1 terhubung dengan Router2 melalui konektor PPP E1 dengan *data rate* 2.048 Mbps dan *propagation delay* sebesar 1 ms. Konektor lainnya pada jaringan mempunyai *propagation delay* sebesar 1  $\mu$ s.

Tiga *workstation* sebagai pengirim data terdiri dari TCP Source, ON/OFF Source, dan Video Source. TCP Source dan ON/OFF Source masing-masing mengirimkan data melalui FTP menuju TCP Dest dan ON/OFF Dest. Sedangkan Video Source mengirimkan video (*video streaming*) MPEG-2 (*mobl\_120.m2v*) menuju Video Dest. Video Source mengirimkan paket-paket berukuran 188 byte (ukuran paket pada MPEG-2 TS) dengan *data rate* 1 Mbps. TCP Source dan ON/OFF Source mengirimkan data juga dengan *data rate* sebesar 1 Mbps. Untuk menghindari *packet loss*, ukuran paket pada TCP Source dan ON/OFF Source ditetapkan sebesar 512 byte. ON/OFF Source mengirimkan data selama periode ON, dan berhenti pada periode OFF. ON/OFF Source melakukan transmisi data secara dinamis yang mempengaruhi tingkat kepadatan trafik pada *router*.



Gambar 3. Topologi jaringan

Interval  $U$  pada koneksi video dihasilkan dengan mengontrol periode ON dari ON/OFF Source. Ketika ON/OFF Source dimatikan, hanya ada dua *flow* di dalam jaringan sehingga trafik tidak padat. Tetapi ketika ON/OFF Source dinyalakan, ada tiga *flow* yang melalui Router1 sehingga trafik menjadi sangat padat.

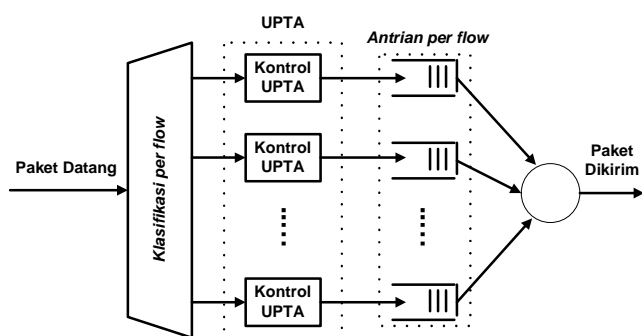
Algoritma UPTA diimplementasikan pada Router1 yang juga mengimplementasikan WFQ. Modifikasi dilakukan pada *process model* di dalam *router* yang menangani atau mengatur antrian paket dengan metode WFQ. Pada OPNET, *process model* yang dimaksud adalah *ip\_output\_iface*. Pada *ip\_output\_iface* ditambahkan prosedur untuk menghitung *arrival rate*, *fairshare*, dan nilai *drop probability* untuk setiap paket yang masuk ke Router1. Implementasi UPTA pada WFQ ditunjukkan pada gambar 4 [3].

Simulasi dilakukan untuk mendapatkan nilai dari parameter-parameter yang menjadi tolak ukur performa jaringan. Parameter-parameter tersebut adalah:

1. *throughput* koneksi TCP: untuk koneksi antara TCP Source dengan TCP Dest, nilai *throughput* diambil dari

jumlah trafik yang diterima pada TCP *Dest* dalam *bits per second* (bps).

2. waktu unggah koneksi TCP: dengan mengirimkan *file* dari TCP *Source* kepada TCP *Dest* dalam ukuran tertentu, waktu yang diperlukan untuk mengunggah *file* tersebut bisa diketahui dengan melihat pada detik keberapa potongan terakhir *file* sampai ke TCP *Dest*.
3. *packet drop* pada *router*: dengan mengetahui jumlah *packet drop* pada Router1, dapat diketahui apakah dengan mengimplementasikan UPTA, video menjadi semakin *unintelligible* atau tidak.
4. *packet loss* pada koneksi video: *packet loss* diukur dengan membandingkan jumlah paket yang dikirimkan oleh *Video Source* dengan jumlah paket yang diterima oleh *Video Dest*.



Gambar 4. Implementasi UPTA pada WFQ

### V. HASIL UJI COBA

Pada bagian ini akan ditunjukkan nilai dari parameter-parameter yang telah diujicobakan menggunakan simulasi pada OPNET. Masing-masing simulasi berdurasi 50 detik. Untuk setiap parameter, uji coba dilakukan dalam 5 skenario. Setiap skenario mempunyai interval *U* yang berbeda-beda dimana pada skenario 1 sampai dengan 5 masing-masing interval *U* berdurasi 10, 20, 30, 40, dan 50 detik.

#### A. Throughput koneksi TCP

Tabel 2 menunjukkan hasil pengukuran *throughput* untuk masing-masing skenario pada jaringan yang mengimplementasikan UPTA dan yang tidak. Hasil pengukuran menunjukkan bahwa penggunaan UPTA mampu meningkatkan nilai *throughput* hingga 1,9%.

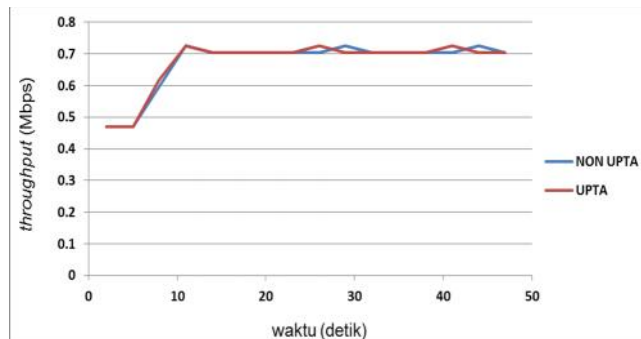
Tabel 2.  
Hasil pengukuran *throughput* pada 5 skenario

Skenario	WFQ Tanpa UPTA	WFQ Dengan UPTA
1	0,6720 Mbps	0,6733 Mbps
2	0,6253 Mbps	0,6280 Mbps
3	0,5787 Mbps	0,5840 Mbps
4	0,5320 Mbps	0,5400 Mbps
5	0,4867 Mbps	0,4960 Mbps

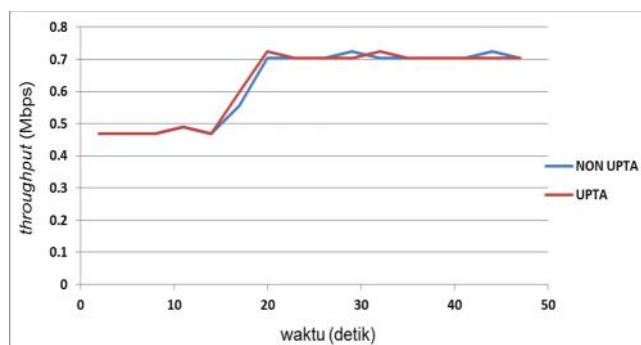
Pada gambar 5 sampai dengan 9 ditampilkan grafik nilai *throughput* pada masing-masing skenario. Dari grafik terlihat bahwa peningkatan *throughput* terjadi selama interval *U*. Sedangkan pada interval *I*, *throughput* kurang lebih sama. Hal ini menunjukkan bahwa UPTA hanya bekerja ketika UPT terjadi yaitu pada interval *U*.

#### B. Waktu unggah koneksi TCP

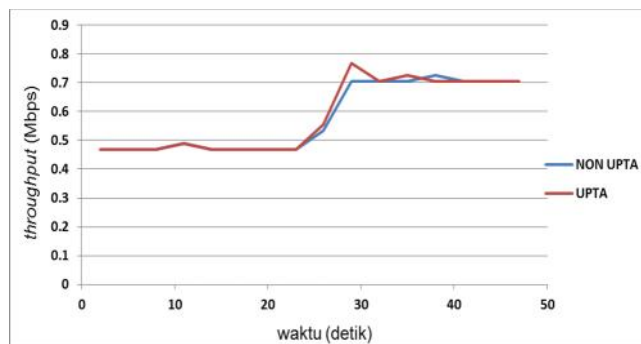
Uji coba untuk mengukur waktu unggah dilakukan dengan mengirimkan 5 *file* dengan ukuran yang berbeda-beda yaitu 15 KB, 200 KB, 1,5 MB, 2 MB, dan 2,5 MB. Pada gambar 10 sampai dengan 14 ditampilkan grafik hasil uji coba. Dari hasil pengukuran didapatkan bahwa UPTA mampu mengurangi waktu unggah *file* hingga 1,2%.



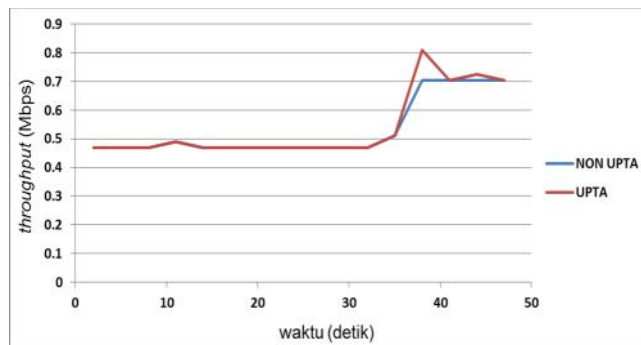
Gambar 5. Grafik *throughput* koneksi TCP pada Skenario 1



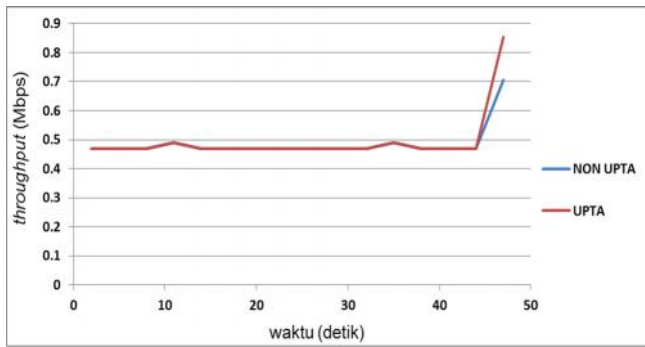
Gambar 6. Grafik *throughput* koneksi TCP pada Skenario 2



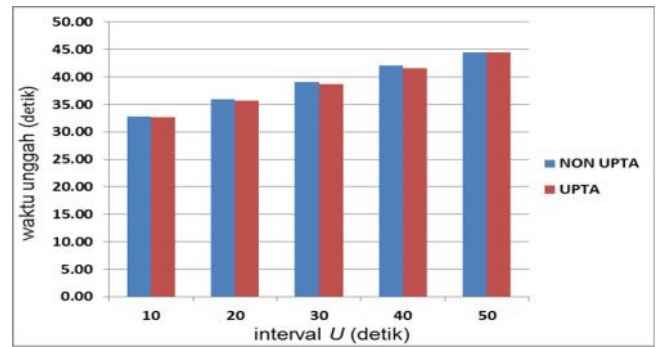
Gambar 7. Grafik *throughput* koneksi TCP pada Skenario 3



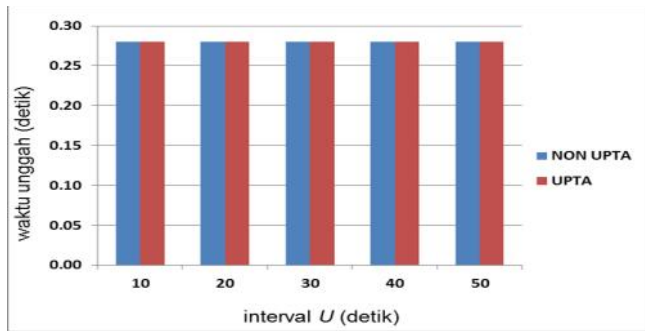
Gambar 8. Grafik *throughput* koneksi TCP pada Skenario 2



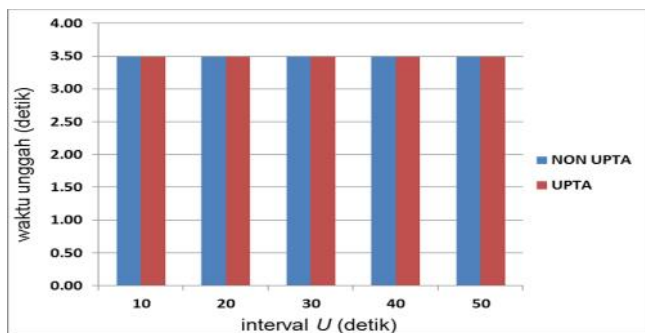
Gambar 9. Grafik throughput koneksi TCP pada Skenario 2



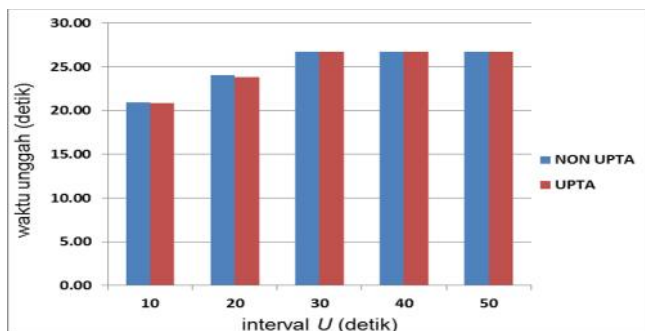
Gambar 14. Grafik waktu unggah file ukuran 2,5 M



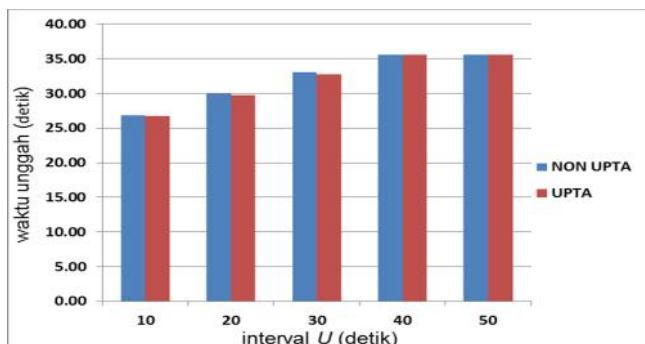
Gambar 10. Grafik waktu unggah file ukuran 15 KB



Gambar 11. Grafik waktu unggah file ukuran 200 KB



Gambar 12. Grafik waktu unggah file ukuran 1,5 MB



Gambar 13. Grafik waktu unggah file ukuran 2 MB

C. Packet Drop

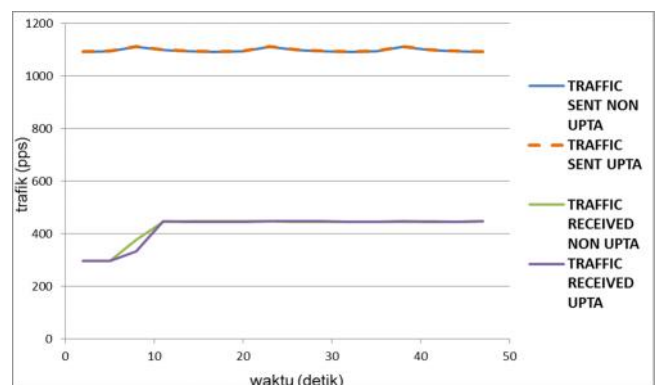
Pada tabel 3 ditunjukkan hasil pengukuran nilai rata-rata packet drop pada masing-masing skenario. Perbedaan packet drop yang paling besar Dari tabel dapat diketahui bahwa pada skenario 5 terjadi hingga 8%. Ini menunjukkan bahwa pada jaringan yang mengimplementasikan UPTA, semakin lama interval U maka akan semakin banyak pula UPT yang dihilangkan oleh algoritma UPTA. Namun, hal ini tidak akan mengurangi kualitas video secara signifikan karena prosentase penambahan packet drop cukup kecil.

Tabel 3. Jumlah Rata-rata Packet Drop pada Router1

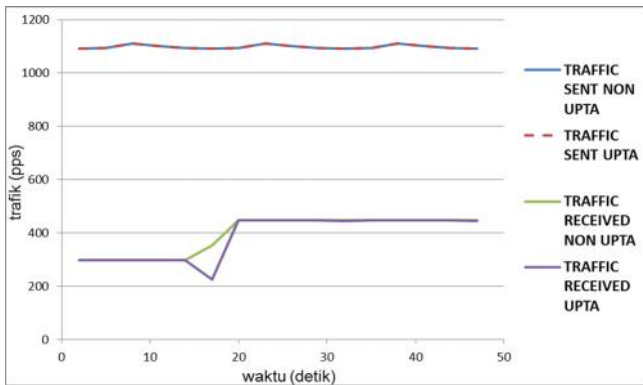
Skenario	WFQ Tanpa UPTA	WFQ Dengan UPTA
1	159,71 packets/s	162,35 packets/s
2	189,04 packets/s	197,02 packets/s
3	218,31 packets/s	231,50 packets/s
4	247,63 packets/s	265,54 packets/s
5	276,16 packets/s	300,15 packets/s

D. Packet Loss

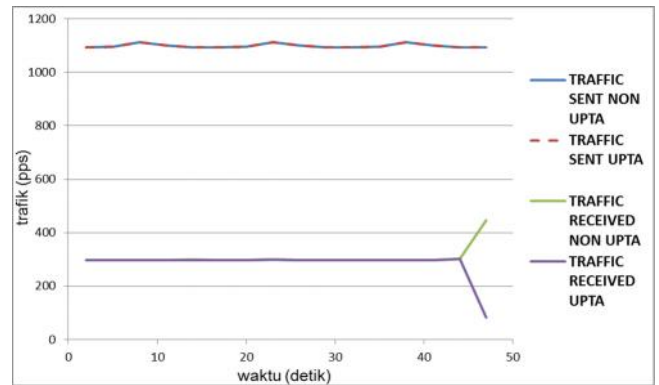
Pada gambar 15 sampai dengan 19, diperlihatkan grafik yang menunjukkan jumlah trafik yang dikirimkan (traffic sent) oleh Video Source dan trafik yang sampai (traffic received) ke Video Dest pada 5 skenario simulasi. Dari keseluruhan skenario dapat dilihat bahwa tidak terdapat perbedaan yang signifikan pada hasil statistik traffic received pada jaringan yang mengimplementasikan UPTA dan yang tidak. Grafik juga menunjukkan bahwa perbedaan packet loss paling besar ada di skenario 5 yaitu 7,4%. Hal ini menunjukkan bahwa UPTA tidak menyebabkan video yang diterima menjadi unintelligible.



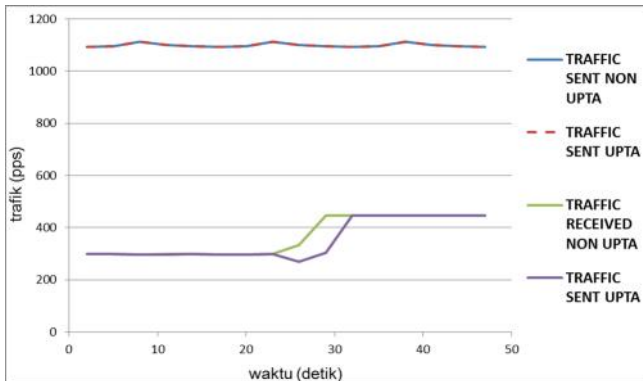
Gambar 15. Grafik traffic sent dan traffic received pada skenario 1



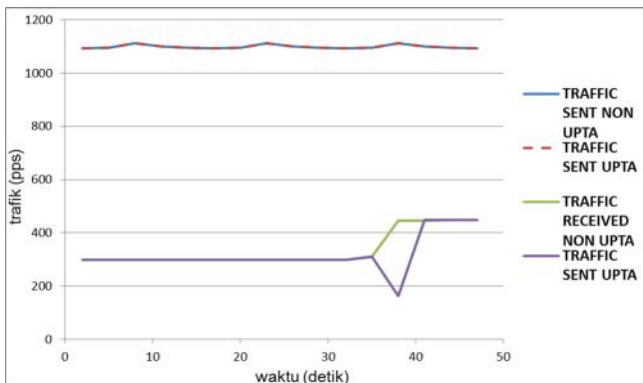
Gambar 16. Grafik traffic sent dan traffic received pada skenario 2



Gambar 19. Grafik traffic sent dan traffic received pada skenario 5



Gambar 17. Grafik traffic sent dan traffic received pada skenario 3



Gambar 18. Grafik traffic sent dan traffic received pada skenario 4

### VI. KESIMPULAN

Pada makalah ini telah dijelaskan permasalahan UPT. UPT muncul sebagai efek samping dari penggunaan algoritma *fair packet queueing* pada router dengan trafik yang sangat padat. Pada makalah ini juga telah dilakukan analisis untuk mengetahui pengaruh penggunaan algoritma UPTA pada router untuk menghilangkan UPT. Dari data-data hasil percobaan dapat disimpulkan bahwa dengan mengimplementasikan algoritma UPTA, *throughput* dari aplikasi non multimedia mengalami peningkatan sehingga data yang dikirimkan oleh aplikasi tersebut lebih cepat sampai ke tujuan. Walaupun dengan mengimplementasikan algoritma UPTA dapat menyebabkan lebih banyak paket yang terbuang, audio/video yang sampai ke penerima tidak akan mengalami penurunan kualitas yang signifikan karena penambahan jumlah paket yang terbuang relatif sedikit.

### DAFTAR PUSTAKA

- [1] Wu, Jizhong, and Mahbub Hassan, "The Issue of Useless Packet Transmission for Multimedia over the Internet". *Proceeding of Computer Communication 26*. Sydney, Australia, (2002).
- [2] Kurose, James F, and Keith W Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison Wesley Longman, (2000).
- [3] Wu, Jizhong, "Techniques to Avoid Useless Packet Transmission in Multimedia over Best-Effort Networks". *Thesis Report*. Sydney, Australia., (2003).
- [4] Jaffe, J.M. "Bottleneck Flow Control." *Proceeding of IEEE Transactions On Communications*, Vol. COM-29, No. 7, (1981).