Terbit online pada laman web jurnal : **http://jurnal.iaii.or.id**

JURNAL RESTI

(Rekayasa Sistem dan Teknologi Informasi)

Vol. 2 No. 3 (2018) 728 – 735 ISSN : 2580-0760 (media online)

Kontrol Bandwidth Dinamis Berbasis Algoritma Logika Fuzzy pada Jaringan Wireless Ad-Hoc

Yedidio Purwodwiyogo[a], Alif Subardono[b]

[a] Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Teknologi Jaringan, yedidio9932@gmail.com

[b] Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Teknologi Jaringan, masalif@gmail.com

Abstract

Ad-Hoc network is a wireless network consisting of a collection of nodes that are connected spontaneously and dynamically. With the character of the Ad-Hoc network, this network is often used in the development of IoT devices (Internet of Things) or emergency conditions. But in the implementation of the Ad-Hoc network there are challenges that need to be addressed, one of which is the very limited bandwidth that can be provide. The existence of these things will lead to new problems, when network traffic is very busy, it will reduce the fairness level of bandwidth distribution. Therefore, the implementation of fuzzy logic is proposed as a reliable decision maker in managing bandwidth. From the results found in this study prove that the results of the obtained throughput performance is not found to be a significant increase, but the jitter and packet loss produces a better value than when the bandwidth management system is applied.

Keywords: Bandwidth, Fuzzy logic, Ad-Hoc

Abstrak

Jaringan *Ad-Hoc* adalah jaringan *wireless* yang terdiri dari kumpulan *node* yang terhubung secara spontan dan bersifat dinamik. Dengan karakter jaringan *Ad-Hoc* tersebut, jaringan ini sering digunakan pada pengembangan perangkat IoT (*Internet of Things*) hingga kondisi darurat. Namun dalam implementasi jaringan *Ad-Hoc* terdapat tantangan yang perlu dihadapi, salah satunya adalah lebar *bandwidth* yang sangat terbatas. Adanya hal tersebut akan mengakibatkan permasalahan baru, ketika *traffic* jaringan sangat sibuk akan berkurangnya tingkat *fairness* pembagian *bandwidth*. Maka dari itu diajukan implementasi logika *fuzzy* sebagai penentu keputusan yang handal dalam memanajemen *bandwidth*. Dari hasil yang didapati pada penelitian ini membuktikan bahwa hasil performa *throughput* yang diperoleh tidak didapati peningkatan yang signifikan, namun pada nilai jitter dan nilai packet loss menghasilkan nilai yang lebih baik dari ketika sistem manajemen *bandwidth* diterapkan.

Kata kunci: *Bandwidth*, Logika *Fuzzy*, *Ad-Hoc*

1. Pendahuluan

Jaringan *Ad-Hoc* adalah jaringan *wireless* yang terdiri dari kumpulan *mobile node* yang bersifat dinamik dan spontan. Jaringan ini terhubung antar satu *node* dengan *node* lain tanpa adanya konfigurasi awal atau kontrol tersentral. Sebagai hasil dari tipe karakteristik jaringan ini, terdapat peramasalahan yang akan dihadapi, salah satu dari permasalah tersebut adalah ketersediaan *bandwidth* yang terbatas. Dari permasalah tersebut tingkat fairness pada jaringan *Ad-Hoc* ini ditakutkan akan menurun ketika sedang pada kondisi trafik yang sibuk. Hal tersebut juga didukung dengan adanya paket *User Datagram Protocol* (UDP) yang sering digunakan pada komunikasi jaringan. Yang mana paket UDP tidak terdapat fungsi *flow bandwidth control*, sehingga dari sisi pengirim akan mengirimkan paket UDP sebanyak mungkin tanpa mengetahui kondisi yang terjadi pada jaringan tersebut.

Dalam memberikan layanan jaringan yang baik dibutuhkan manajemen kualitas jaringan yang sesuai dengan kebutuhan layanan yang diaplikasikan pada jaringan tersebut. Maka dari itu dibutuhkan suatu aturan dalam manajemen jaringan yang dapat menjamin tersedianya layanan tersebut. Seperti membagi lebar bandwidth untuk setiap layanan yang disediakan serta memberikan skala prioritas untuk layanan tersebut. Dengan penerapan manajemen jaringan sebagai mana yang diusulkan sebelumnya, akan memastikan layanan yang ada pada jaringan tersebut terkirim dengan baik ke pengguna layanan jaringan.

Diterima Redaksi : 25-09-2018 | Selesai Revisi : 16-11-2018 | Diterbitkan Online : 16-12-2018

Manajemen lebar *bandwidth* dapat menjadi jalan keluar atas permasalahan ketersediaan yang terbatas. Namun dibandingkan memberikan lebar *bandwidth* untuk setiap layanan secara tetap, akan lebih baik ketika lebar *bandwidth* secara dinamis dapat berubah sesuai kondisi trafik [3] yang berlangsung pada jaringan *Ad-Hoc*. Tentunya dengan berubahnya lebar *bandwidth* ini juga harus memastikan bahwa layanan lain tidak terganggu ketersediaannya. Sebagai alternatif dalam manajemen *bandwidth* secara dinamis diperlukan suatu kontrol yang dapat menetukan lebar *bandwidth* yang diperlukan. Terdapat berbagai cara dalam menerapkan konsep manajemen *bandwidth* secara dinamis, sebagai contohnya dapat menerapkan teori kecerdasan buatan. Untuk mendukung usulan alternatif tersebut teori kecerdasan buatan perlu diaplikasikan untuk menetukan *output* yang diberikan dalam kondisi tertentu secara langsung, yang mana teori logika *fuzzy* dapat dijadikan sebagai landasan menerapkan manajemen *bandwidth* secara dinamis.

Logika *fuzzy* sendiri merupakan suatu metode dalam menentukan nilai yang berada pada titik abu-abu (nilainya berada pada antara 0 – 1), yang mana logika fuzzy ini menggantikan kebenaran *boolean* dengan tingkat kebenaran. Dalam penerapannya, logika *fuzzy* pada sistem manajemen kualitas jaringan akan mengelola lebar *bandwidth* disesuaikan dengan kondisi kebutuhan jaringan pada saat itu juga. Hal tersebut dapat terjadi ketika parameter kualitas jaringan dijadikan tolak ukur dalam penentuan keputusan. Oleh karena itu pada penelitian ini akan menerapkan logika *fuzzy* pada jaringan *Ad-Hoc*, dengan harapan dapat meningkatkan *fairness* pada jaringan *Ad-Hoc*.

Terdapat juga rumusan masalah yang ditentukan pada penelitian ini yaitu; Apakah dengan implementasi logika *fuzzy* dapat meningkatkan *throughput* dan *fairness* pada jaringan *Ad-Hoc*? Serta bagaimana perbandingan performa logika *fuzzy* dengan sistem manajemen *bandwidth* konvensional pada jaringan *Ad-Hoc*?

Dari latar belakang yang telah disebutkan, penelitian ini ditujukan untuk menerapkan logika *fuzzy* pada manajemen *bandwidth* secara dinamis pada jaringan *Ad-Hoc*, mengetahui perbandingan kinerja logika *fuzzy* dengan sistem manajemen *bandwidth* yang konvensional dan juga meningkatkan *throughput* dan *fairness* pada pada jaringan *Ad-Hoc*.

2. Tinjauan Pustaka

Logika *Fuzzy* merupakan salah satu kecerdasan buatan yang merupakan bukan hal baru dalam memberikan keputusan terhadapat suatu kondisi tertentu. Hal ini dapat memberikan hasil yang lebih optimal dalam suatu proses yang sedang berjalan, dikarenakan sumber daya yang digunakan pada suatu proses tersebut digunakan sesuai dengan kebutuhan pada kondisi yang memiliki sifat kedinamisan yang tinggi. Maka dari itu terdapat sejumlah penelitian mengenai strategi manajemen bandwidth yang dinamis menggunakan logika *fuzzy* untuk mengoptimalkan penggunaan sumber daya bandwidth yang digunakan.

Salah satu penelitian yang telah dilakukan, dalam penerapan manajemen bandwidth dilakukan dengan mengelompokkan trafik menjadi dua jenis yaitu; *Restricted Time-Bounded Service* (RTBS) untuk layanan suara dan video, dan *Loose Time-Bounded Service* (LTBS), untuk layanan data. Yang mana setiap dari jenis layanan tersebut memiliki karakteristik trafik, kebutuhan *bandwidth*, dan kemungkinan terjadinya *blocking* yang dapat diterima. Dalam penelitian ini menyebutkan bahwa skema pembagian *bandwidth* yang agresif akan memberikan manajemen *bandwidth* yang lebih baik, mengingat kondisi trafik yang sering mengalami fluktuasi yang tidak terduga, dan juga dipastikan diminimalkan kemungkinan *blocking* pada RTBS. Untuk mengatasi masalah tersebut pada penelitian ini mengajukan lima strategi manajemen *bandwidth*; *Early Request First Strategy* (ERFS) yang merupakan konsep sederhana dari *first-come-first-serve* memastikan tercapainya *fairness*, namun terdapat peningkatan kemungkinan terjadinya *blocking*, *Short Request First Strategy* (SRFS) mengutamakan layanan terpendek pada *buffer* untuk diproses, hal ini akan menyebabkan layanan dengan waktu terlama dan interval waktu *shareable slot* yang digunakan oleh LTBS akan semakin panjang. Konsekuensinya adalah permintaan RTBS akan terabaikan dan permintaan yang lebih dulu datang akan ditunda telebih dahulu yang menyebabkan berkurangnya *fairness*, *Adaptive* SBLR konsep utama dalam strategi ini adalah layanan terpendek akan ditempatkan pada *Borrowed channel* dan layanan terpanjang akan ditempatkan pada *Reserved channel*. Strategi ini akan mencapai kemungkinan *blocking* yang lebih kecil dari pada SRFS. *Fuzzy ERFS/SRFS Strategy* (FESS) merupakan metode *switching* antara ERFS dan SRFS, seperti yang telah diketahui bahwa keduanya memiliki kelebihan masing-masing. Ketika RTBS trafik mengalami peningkatan kemungkinan *blocking* SRFS akan digunakan, selain itu akan menggunakan ERFS untuk meningkatkan *fairness* jaringan, *Enhanced* FESS (EFESS) pada strategi ini menggunakan dua strategi secara bersamaan. Ketika kapasitas *shareable bandwidth* dapat mengatasi RTBS trafik, ERFS dapat diaplikasikan pada *borrowed slot.* Ketika kebutuhan RTBS menggunakan keseluruhan kapasitas *shareable bandwidth*, *reserved slot* digunakan untuk melayani paket dengan strategi ERFS dan *borrowed slot* melayani paket dengan waktu lanyanan terpendek. Hasil yang didapat dari kelima strategi ini adalah dengan mengaplikasikan logika *fuzzy* dapat memastikan utilisasi *bandwidth* dan *fairness* yang tinggi dan kemungkinan *blocking* yang rendah [2].

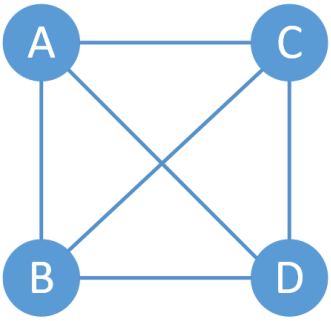
Dalam pengujian strategi sistem manajemen *bandwidth* dinamis, terdapat pula strategi dengan menganalisis *buffer* pada trafik jaringan dengan *single output link*. Konsep utama dalam implemetasi logika *fuzzy* pada strategi ini adalah dengan dengan membandingkan *buffer* dari sumber yang berbeda, yang kemudian dari nilai yang didapat akan menjadi tolak ukur dalam menetukan *buffer* dari sumber mana yang perlu mendapat lebar *bandwidth* lebih. Di waktu yang sama pada melayani *buffer* tersebut, dilakukan juga penghitungan ulang terhadap nilai lebar bandwidth. Hal tersebut dilakukan secara berkala dalam interval waktu tertentu. Dari hasil yang didapat pada penelitian dengan strategi ini dapat disimpulakan bahwa mekanisme manajemen *bandwidth* ini meningkatkan performa *buffered network* dan memberikan hasil QoS yang lebih baik dari alternatif yang ditawarkan ketika sumber trafik memiliki intensitas perubahan yang tinggi pada interval waktu yang pendek [3].

Penelitian lain dilakukan pada jaringan yang dibagi menjadi beberapa jaringan *virtual*, yang ditujukan untuk memungkinkan pengguna jaringan dapat mengelompokkan trafik. *Dynamically Adaptive Virtual Network for a Customized Internet* (DaVinci) merupakan suatu teknik yang memungkinkan terbentuknya beberapa *virtual network* dengan membagi setiap node fisik dan link fisik menjadi beberapa node virtual dan link virtual. Masalah yang dijumpai pada DaVinci, utilisasi dari setiap link *virtual* yang telah terbentuk tidak semua terpakai secara maksimal. Logika *fuzzy* sendiri membutuhkan 3 variabel sebagai input yaitu; rata-rata utilisasi link *virtual*, rata-rata panjang antrian dan rata-rata paket *delay*. Dari setiap variabel tersebut dibuat fungsi keanggotaan yang didaftarkan dengan nama *LOW*, *MEDIUM* dan *HIGH*. Pada percobaannya dibuat 2 *virtual* dengan kondisi variabel input fuzzy yang berbeda. Kemudian dari input ynag masuk akan bandingkan dengan *rule* yang telah dibuat. Pada dasarnya *rule* tersebut memastikan ketika utilisasi pada link virtual pada kondisi *HIGH* serta *delay* atau panjang antrian tidak sama dengan *LOW* maka sisa *bandwidth* pada link *virtual* lain akan akan dipindahkan ke link virtual yang membutuhkan. Hasil dari uji coba ini adalah rata-rata delay yang dihasilkan lebih kecil dari pada DaVinci yang tidak menggunakan fuzzy, begitu pula pada nilai utilisasi yang diperoleh lebih besar [1].

Dalam penerapan teori kecerdasan buatan pada manajemen *bandwidth*, strategi lain diajukan ketika diimplementasikan pada jaringan Ad Hoc. Strategi ini menggunakan variabel *price* yang diperoleh dari pesan ACK yang dikirimkan oleh *node destination* ke *node sender*. Variabel *price* tersebut akan digunakan oleh *node sender* untuk menentukan lebar *bandwidth* yang dibutuhkan. Skema kontrol ini dinilai cukup adil dalam memungkinkan suatu *flow* untuk menambah nilai *price* pada *route* yang dipakai. Pada jaringan *wireless* node meneruskan paketnya dengan melakukan *broadcast* ke tetangganya. Yang mana paket tersebut akan tersimpan sementara di *buffer* untuk diterukan ke *node* tujuan. Pada buffer harus ditentukan nilainya untuk dapat menetukan *network delay* dan *throughput*. Nilai *price* dari suatu router ditentukan dari besaran antrian, yang mana besaran antrian tersebut menjelaskan perbedaan antara *network load* dan *wireless bandwidth*. Konsekuensinya adalah nilai *price* akan terus ditambahkan hingga utilisasi dari router lebih besar dari *threshold*, selain itu nilai *price* akan dikurangikan. Dari konsep tersebut *Fuzzy Source Controller* (FSC) yang digunakan untuk menetukan *source rate* pada jaringan *Ad Hoc*, disimpulkan bekerja lebih baik pada jaringan *wireless multi-hop*. Hasil yang diperoleh pada skema ini menghasilkan respon yang baik dari pemakaian *buffer* dan *transmission rate*, sehingga diperoleh utilisasi yang lebih baik [4].

3. Metodologi Penelitian

3.1 Perancangan topologi



Gambar 1. Topologi jaringan Ad-Hoc

Topologi jaringan dalam proyek akhir ini dirancang untuk dapat menjalankan 3 skenario yaitu: tanpa menerapkan manajemen *bandwidth*; manajemen *bandwidth* dengan memberikan nilai lebar *bandwidth* yang tetap; manajemen *bandwidth* dengan menerapkan logika *fuzzy* sebagai kontrol lebar *bandwidth*. Topologi ini terdiri atas 4 buah Raspberry Pi 3 Model B yang terhubung secara wireless dengan mode *Ad-Hoc*. Sistem routing pada jaringan ini tidak diterapkan, sehingga sifat jaringan ini adalah *peer-to-peer*.

Dalam menentukan alamat IP, setiap node berada pada satu jaringan yang sama yaitu 10.0.0.0/24. Untuk lebih jelasnya dapat dilihat pada Tabel 1.

Tabel 1.Tabel Software dan Hardware Pendukung

Nama Perangkat	Alamat IP
Node A	10.0.0.1/24
Node B	10.0.0.1/24
Node C	10.0.0.1/24
Node D	10.0.0.1/24

3.2 Metode logika fuzzy

Berikut merupakan bagian-bagian yang ada pada sistem logika *fuzzy* :

1. Pembentukan variabel

Veriabel *input* yaitu besar *packet* TCP, yang kemudian akan menghasilkan nilai *P* dari hasil perhitungan sebagai berikut:

$$P = 1 - \frac{besar\ packet\ TCP}{kapasitas\ buffer} \quad (1)$$

Dari hasil perhitungan tersebut akan varibel *P* selalu menghasilkan nilai dengan interval [-1,1]. Dengan begitu [-1,1] dapat dikatakan merupakan semesta pembicaraan dari *P*.

2. *Fuzzy*fikasi

Pada himpunan *fuzzy* ini variabel *P* memiliki 3 himpunan *fuzzy* yaitu sangat negatif, nol dan sangat positif. Himpunan *fuzzy* untuk variabel *P* ditunjukan dalam gambar 2.

Gambar 2. Himpunan *fuzzy* untuk variabel *P*

Pada himpunan *fuzzy* sangat negatif memiliki domain [-1, 0] dengan derajat keanggotaan tertinggi (=1) terletak pada nilai antara $-1\ -(-0{,}5)$ dengan persamaan :

$$f_{V.Neg.}[P] = \begin{cases} 1; \\ (0-P)/(0-(-0{,}5)); \\ 0; \end{cases} \quad (2)$$

Pada himpunan *fuzzy* nol memiliki domain [-0,5, -0,5] dengan derajat keanggotaan tertinggi (=1) terletak pada nilai 0 dengan persamaan :

$$f_{Zero}[P] = \begin{cases} 0; \\ (P-(-0{,}5))/(0-(-0{,}5)); \\ (0{,}5-P)/(0{,}5-0); \end{cases} \quad (3)$$

Pada himpunan *fuzzy* sangat positif memiliki domain [0, 1] dengan derajat keanggotaan tertinggi (=1) terletak pada nilai antara $0{,}5 - 1$ dengan persamaan :

$$f_{V.Pos.}[P] = \begin{cases} 0; \\ (P-0)/(0{,}5-0); \\ 1; \end{cases} \quad (4)$$

3. Inference Engine

Setelah menentukan derajat keanggotaan dari setiap himpunan *P*. Pada bagian ini akan ditentukan bagaimana mengatur nilai α dari himpunan yang telah dibentuk. Pada cara kerja dari *inference engine* ini dibutuhkan *fuzzy rule* untuk menalarkan keluaran yang diharapkan. *Fuzzy rule* tersebut ditulis sebagai berikut:

R1: Jika *P very negative*, maka α *small*
R2: Jika *P* mendekati *zero*, maka α *medium*
R3: Jika *P very positive*, maka α *large*

Dari *rule* yang telah ditentukan dapat dijelaskan dengan contoh sebagai berikut. Jika *P* berada pada *domain very negative* maka konten 2 melebihi dari konten 1, yang mengimplikasikan kebutuhan untuk meningkatkan layanan pada konten 2 dengan kata lain nilai dari α perlu diturunkan. Untuk *rule* 2 dan *rule* 3 juga memiliki penjelasan yang kurang lebih sama.

4. De*fuzzy*fikasi

Pada program logika fuzzy ini menerapkan metode Sugeno yang mana menghasilkan output berupa konstanta. Maka dari itu pada variable α dengan semesta pembicaraan dari nilai 0 sampai dengan 40, yang mana nilai tersebut merupakan nilai bandwidth yang ditentukan berdasarakan hasil performa *throughput* yang konsisten. Variable α juga memiliki 3 himpunan yaitu *small*, *medium* dan *large* dimana dalam pembagian domain untuk setiap himpunan diserakan pada *library* Scikit-Fuzzy.

Nilai keluaran dari hasil defuzzyfikasi akan dijadikan untuk menentukan besaran *bandwidth* yang diberikan pada trafik UDP. Hal ini dilakukan karena pada protokol UDP tidak terdapat fungsi *flow control*, sehingga mangakibatkan trafik UDP lebih mendominasi dari pada trafik TCP

3.3 Proses pengambilan data

Dalam pengambilan data, dibagi menjadi dua bagian yaitu; dengan menggunakan *software* Iperf3 dan D-ITG. Dalam penelitian ini kedua *software* tersebut akan membuat trafik UDP dan TCP yang mana proses tersebut akan berlangsung selama 60 detik. Selama kurun waktu tersebut akan terdapat tiga trafik dalam jaringan *Ad-Hoc* yang telah dirancang, yaitu; UDP, TCP1 dan TCP2. Ketiga trafik tersebut juga berjalan dalam interval waktu yang berbeda dan waktu mulai yang berbeda, upaya ini dilakukan untuk dapat melihat bagaimana karakter kerja dan sifat kedinamisan dari protokol UDP dan TCP ketika dikirimkan. Secara detail dalam proses pengambilan data akan dijelaskan sebagai berikut:

a. Iperf3

Iperf3 merupakan salah satu software yang ditujukan untuk membuat trafik dalam suatu jaringan. Hal ini ditujukan untuk memperoleh data yang kemudian dapat digunakan untuk menyatakan kemampuan dari jaringan yang diuji. Kelebihan dari Iperf3 ini dapat menampilkan kemampuan jaringan dalam interval waktu yang dibutuhkan, untuk kasus ini interval waktu ditentukan setiap satu detik. Proses trasnmisi paket dalam jaringan ini akan terdapat tiga *node* yang

menjadi *server* (Node A, Node B dan Node C) dan satu *node* yang menjadi *client* (Node D). Tugas dari *node client* ini akan mengirimkan paket UDP ke *node server* selama 60 detik dengan ukuran paket sebesar 8 Kbit dan mengirimkan paket TCP ke *node server* selama 25 detik dengan ukuran paket sebesar 128 Kbit, dimana ukuran paket yang disebutkan merupakan nilai *default*. Dalam kurun waktu 60 detik akan dijalankan terlebih dahulu trafik UDP, kemudian setelah 10 detik berjalan akan ditambahkan trafik TCP1 dan 25 detik berjalan trafik TCP2 ditambahkan ke dalam jaringan. Berikut merupakan contoh perintah untuk trafik UDP pada sisi client:

iperf3 –c <ip_server> -t <waktu_traffic> -u –b 0

dan berikut merupakan perintah dari trafik TCP:

iperf3 –c <ip_server> -t <waktu_traffic>

pada paket UDP "–b 0" ditujukan agar paket UDP tidak dibatasi oleh *bandwidth* yang secara default bernilai 1Mbps.

b. D-ITG

D-ITG juga merupakan *software traffic generator* yang memiliki flexibilitas yang lebih dari pada iperf3, namun yang membedakan adalah pada D-ITG tidak dapat menampikan hasil setiap interval waktu dan hanya menampilkan rata-rata dari trafik yang dibuat. Secara garis besar metode pembuatan trafik pada jaringa *Ad-Hoc* ini identik dengan apa yang ada pada Iperf3, yaitu terdapat tiga trafik UDP, TCP1 dan TCP2 yang dijalankan pada waktu interval dan waktu mulai yang berbeda. Sama juga dengan Iperf3, D-ITG juga membutuhkan node yang menjadi *client* dan *server*. Berikut merupakan contoh perintah pada sisi *client* untuk trafik UDP:

./ITGSend –T UDP –a <ip_server> -c 8192 –C 100000 –t 60000 –l sender.log –x receiver.log

Dan berikut merupakan contoh perintah untuk trafik TCP:

./ITGSend –T TCP –a <ip_server> -c 131072 –C 100000 –t 25000 –l sender.log –x receiver.log

Perintah diatas ditujukan untuk membuat trafik UDP dengan besar paket 8 Kbit dan banyak paket per detik sebesar 100000 pps selama 60 detik, begitu pula pada trafik TCP dengan besar paket 128 Kbit dan banyak paket perdetik sebesar 100000 pps selama 25 detik.

3.4 Detail skenario

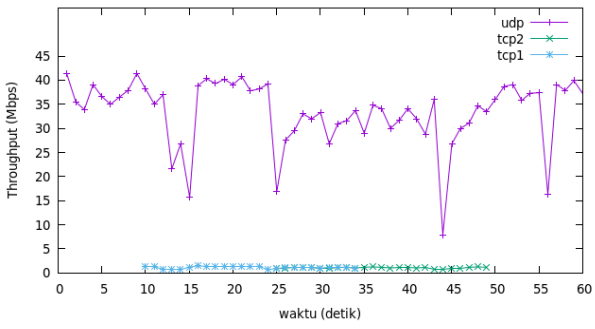
Pada penelitian ini, untuk dapat mengambil kesimpulan dari menerapkan logika *fuzzy* pada manajemen *bandwidth* terdapat tiga skenario yang diusulkan. Tiga skenario tersebut merupakan; tanpa menerapkan manajemen *bandwidth*, manajemen *bandwidth* dengan memberikan nilai lebar *bandwidth* yang tetap, manajemen *bandwidth* dengan menerapkan logika *fuzzy* sebagai kontrol lebar *bandwidth*.

Pada skenario yang pertama yaitu tanpa manajemen *bandwidth*, sehingga trafik yang telah dibuat oleh *software* Iperf3 dan D-ITG akan mengalir sesuai dengan karakteristik dari protokol UDP dan TCP. Kemudian pada penerapan manajemen *bandwidth* secara konvensional, hanya trafik dengan protokol UDP saja yang diberikan besaran *bandwidth*-nya dengan besar nilai 13.3 Mbps yang didapat dari membagi tiga dari dari nilai maksimal *throughput* yaitu sebesar 40 Mbps. Hal tersebut dilakukan karena pada program logika *fuzzy* hanya memberikan perintah pada untuk membatasi nilai bandwidth pada trafik UDP saja. Penerapan logika *fuzzy* sebagai manajemen *banwidth* yang mana merupakan skenario ketiga, dilakukan dengan menjalankan *script* program berbasis python pada sisi *client*. Setiap pengambilan data dilakukan sebanyak 4 kali untuk data yang didapat dari D-ITG untuk mendapatkan nilai rata-rata QoS yang dapat dikatakan *valid*. Sedangkan pada iperf3 dikarenakan diubutuhkan data aktual yang mana hanya dilakukan sekali, hal ini dilakukan agar memperoleh pola karakteristik dari protokol UDP dan TCP ketika digunakan pada jaringan *Ad-Hoc*.

4. Hasil dan Pembahasan

Pada bagian ini dibagi menjadi dua bagian, antara lain yaitu; hasil data yang diperoleh dari *software traffic generator* Iperf3 dan D-ITG. Pada Iperf3 data berupa hasil throughput berdasarkan interval 1 detik, sedangkan pada D-ITG hasil yang data yang diperoleh berupa nilai rata-rata *jitter*, *packet loss* dan *throughput* dari 4 kali pengambilan data.

4.1 Hasil Pengujian Iperf3

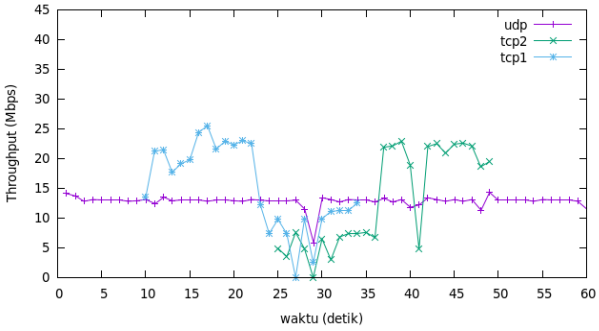


Gambar 3. Hasil *throughput* tanpa manajemen *bandwidth*

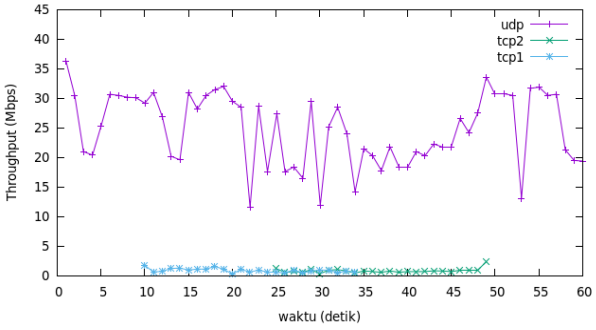
Dari hasil pengujian tanpa adanya manajemen bandwidth, dapat dikatakan bahwa pada hasil trafik UDP sangat mendominasi dibandingkan dengan trafik TCP1 dan TCP2. Hal ini disebabkan pada protokol UDP tidak terdapat adanya fungsi *flow control* yang pada akhirnya mengakibatkan paket UDP akan dikirimkan tanpa melihat kondisi trafik pada jaringan yang digunakan. Nilai *throughput* yang diperoleh trafik

UDP ini tidak begitu stabil dengan ditandainya nilai yang bersifat fluktiatif, dengan nilai tertingginya mampu sampai lebih dari 40 Mbps dan nilai terendah hingga 10 Mbps pada detik ke-44. Dari hasil trafik TCP1 dan TCP2 nilai *throughput* yang diperoleh cukup stabil di nilai kurang dari 3 Mbps.

Hasil dari pada skenario penerapan manajemen *bandwidth* konvensional untuk membatasi throughput dari trafik UDP agar tidak lebih dari 13.3 Mbps, membuktikan bagaimana kerja fungsi flow control yang ada pada ptotokol TCP. Trafik TCP1 dan TCP2 pada detik ke-25 yang mana trafik TCP2 mulai masuk pada jaringan, trafik TCP1 secara langsung menurukan besar *bitrate* agar pada trafik TCP2 mendapatkan *throughput* yang dapat dikatakan cukup adil dengan trafik TCP1. Kemudian pada detik ke-35 dimana trafik TCP1 telah selesai trafik TCP2 terdapat kenaikan nilai *bitrate* yang sama dengan trafik TCP1 ketika hanya terdapat satu trafik protokol tcp pada jaringan.



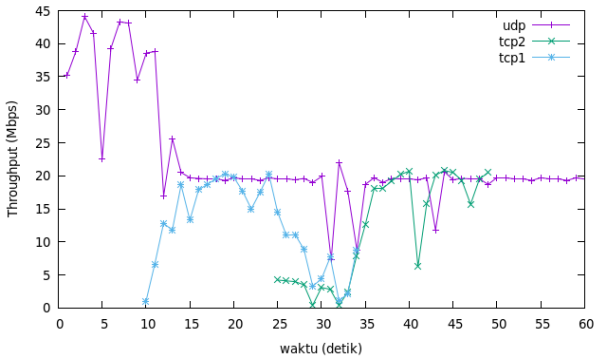
Gambar 4. Hasil *throughput* dengan manajemen *bandwidth* konvesional



Gambar 5. Hasil *throughput* dengan logika *fuzzy*

Pada hasil yang diperoleh pada skenario yang mengimplementasikan logika *fuzzy* sebagai manajemen *bandwidth*, tingkat kestabilan nilai *throughput* yang diperoleh dikatakan lebih buruk dari pada tanpa adanya manajemen *bandwidth*. Hal ini disebabkan adanya antrian dalam membaca nilai lebar paket tcp, yang mana kondisi trafik pada jaringan tidak sesuai dengan keluaran yang dihasilkan. Penyebab utamanya adalah waktu yang cukup lama dalam mengubah rule pada program Tcconfig. Maka dari itu *throughput* pada trafik TCP1 dan TCP2 hampir tidak memiliki perbedaan dengan yang dihasilkan pada skenario tanpa menerapkan manajemen *bandwidth*.

Sebagai jalan keluar dari permasalahan yang terjadi pada skenario sebelumnya, dilakukan penambahan pengembangan terhadap *script* logika *fuzzy*. Hal tersebut dilakukan dengan cara memfilter paket yang dijadikan nilai masukan pada logika *fuzzy*. Pada Gambar 4.4 merupakan hasil yang diperoleh setelah memastikan hanya paket dengan ukuran lebih dari 1500 bit saja yang menjadi nilai inputnya. Dari hasil yang diperoleh ketika trafik TCP1 masuk, dengan segera trafik UDP diturunkan lebar *bandwidth*nya. Hal ini terjadi dikarenakan hasil dari memfilter paket yang dijadikan nilai masukan, berbeda dengan program sebelumnya yang membaca secara keseluruhan varisasi dari besar paket yang diterima. Dengan stabilnya besar *bandwidth* untuk trafik UDP, maka throughput yang dihasilkan pada trafik TCP1 dapat meningkat. Ketika trafik TCP2 masuk ke jaringan terdapat lonjakan dan turunan *throughput* pada setiap trafik, dilanjukan dengan meningkatnya throughput trafik TCP2 dan kestabilan throughput UDP setelah trafik TCP1 berakhir. Namun terdapat kelemahan pada sistem ini yaitu ketika trafik TCP2 berakhir, *throughput* trafik UDP tidak mengalami peningkatan. Hal ini dikarnakan sistem filter pada nilai masukan yang mencegah paket dengan ukuran kecil untuk diproses oleh logika *fuzzy*. Dengan ini lebar bandwidth pada trafik UDP tidak dapat ditingkatkan.



Gambar 6. Hasil *throughput* dengan logika *fuzzy* dengan memfilter paket

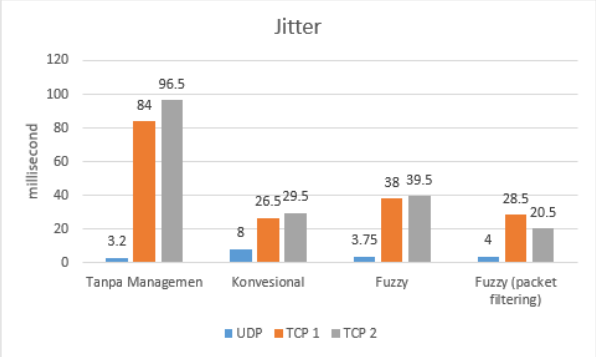
4.2 Hasil Pengujian D-ITG

Hasil yang diperoleh pada uji coba dengan D-ITG ini, terdapat 3 data yang dibahas. Data tersebut merupakan data nilai rata-rata *jitter*, nilai rata-rata *packet loss* dan nilai rata-rata *throughput*, yang mana dibuat perbandingan antara skenario satu dengan yang lain.

a. Hasil *jitter*

Dari hasil yang diperoleh pada trafik UDP untuk setiap skenario, nilai *jitter* termasuk dalam kategori bagus dengan nilai terendah sebesar 3,2 ms dan nilai terbesar 8 ms. Pada trafik TCP1 dan TCP2 memperlihatkan bahwa nilai *jitter* untuk skenario tanpa manajemen

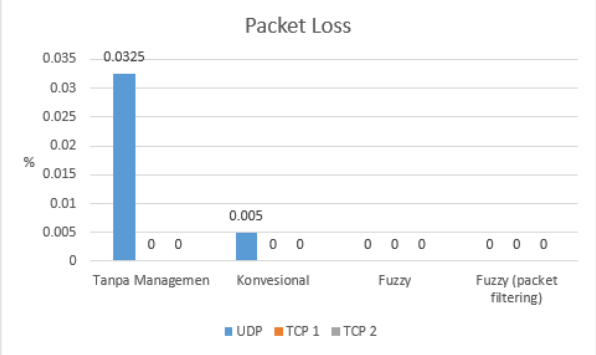
bandwidth menghasilkan nilai yang cukup tinggi yaitu 84 ms dan 96,5 ms, hasil tersebut termasuk dalam kategori sedang dan dapat dikatakan bahwa terjadi antrian paket yang cukup tinggi.



Gambar 7. Perbandingan nilai rata-rata *jitter*

Skenario dengan memanajemen *bandwidth* secara konvensional membuktikan bawah dengan menetapkan lebar bandwidth pada trafik UDP mampu untuk mengurangi terjadinya antrian paket yang ditandai dengan menurunnya nilai jitter pada trafik TCP1 dan TCP2 yang cukup signifikan. Pada penerapan logika *fuzzy* juga memberikan nilai yang baik walaupun terdapat kenaikan yang dihasilkan sebesar kurang lebih 10 ms dari pada skenario sebelumnya dan menghasilkan nilai throughput yang terbilang rendah, hasil throughput dapat dilihat pada Gambar 9. Pada bagian fuzzy dan filtrasi paket ini dapat dikatakan menghasilkan nilai jitter yang lebih baik dari pada skenario lainnya, hal tersebut ditandainya nilai *jitter* yang lebih rendah.

b. Hasil *packet loss*

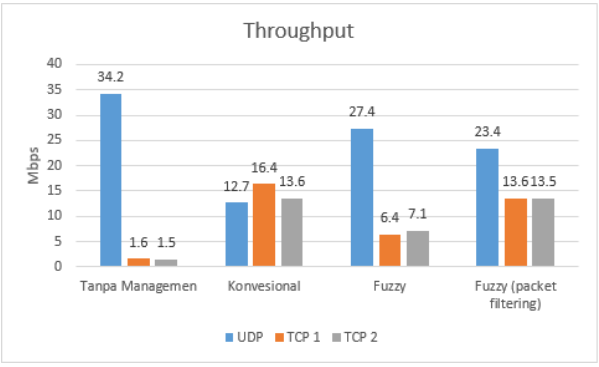


Gambar 8. Perbandingan nilai rata-rata *packet loss*

Pada Gambar 8 menjelaskan bahwa pada dasarnya nilai *packet loss* dari semua skenario terbilang sangat baik. Hal ini didukung dengan topologi jaringan *Ad-Hoc* yang memungkinkan setiap *node* dapat berkomunikasi dengan *node* lainya dengan cara *peer-to-peer*, sehingga meminimalisir terjadingan *packet loss*. Untuk skenario tanpa manajemen *bandwidth* sekalipun menghasilkan *throughput* yang tinggi dan lebih dominan dari trafik lainnya, hal tersebut tidak memastikan menghasilkan nilai yang lebih baik dibandingkan dengan skenario 2. Pada penerapan logika *fuzzy* dapat dikatakan cukup berhasil dalam memastikan tidak adanya *packet loss*

c. Hasil *throughput*

Hasil *throughput* dari D-ITG memiliki karakteristik yang sama dengan hasil yang diperoleh dari Iperf3. Dimana pada penerapan tanpa adanya manajemen *bandwidth* hasil performa *throughput* dari trafik UDP sangat mendominasi, sehingga menurunkan performa *throughput* dari trafik TCP1 dan TCP2. Pada penerapan manajemen *bandwidth* konvensional terlihat bahwa trafik UDP diberi limitasi yang ditandai dengan besar throughput yang tidak melebihi 13,3 Mbps. Begitu pula dengan hasil yang diperoleh pada trafik TCP1 dan TCP2 dimana kedua trafik dapat membagi beban trafik sendiri sehingga dapatkan hasil performa throughput yang tidak terlalu berbeda. Penerapan logika fuzzy sendiri kurang begitu terlihat perbedaan dengan hasil yang diperoleh tanpa menerapkan manajemen *bandwidth* dimana masih terlihat mendominasinya trafik UDP dari pada trafik lain. Hal tersebut disebabkan adanya keterlambatan program logika *fuzzy* dalam membaca kondisi trafik ketika trafik TCP masuk. Ketika ditambahkan baris perintah untuk memfilter paket dengan ukuran lebih dari 1500 bit pada *script* program logika *fuzzy*, terlihat bahwa terdapat peningkatan hasil *throughput* pada trafik TCP1 dan TCP2. Bahkan total *throughput* yang diperoleh lebih tinggi dari pada scenari lain. Dengan ini setelah melakukan perbaikkan pada script program logika *fuzzy* didapati hasil yang lebih baik. Namun pada penerapan memfilter besar paket yang diproses akan berdampak ketika hanya trafik UDP yang ada pada jaringan, dimana lebar *bandwidth* pada trafik tersebut tidak dapat diperbesar.



Gambar 9. Perbandingan nilai rata-rata *throughput*

5. Kesimpulan

5.1 Simpulan

Berdasarkan hasil analisis pengujian kontrol *bandwidth* dinamis berbasis algoritma logika *fuzzy* pada jaringan *wireless ad-hoc*, maka dapat diambil kesimpulan sebagai berikut.

1. Penerapan logika *fuzzy* sebagai system manajemen *bandwidth* dinamis menghasilkan performa *throughput* yang hampir tidak ada bedanya dengan kondisi tanpa menambahkan manajemen *bandwidth* pada jaringan, yang mana didapati trafik UDP yang sangat mendominasi dari trafik TCP.
2. Hasil perbandingan performa QoS pada setiap skenario yang diperoleh, dapat dikatakan bahwa dengan menerapakan sistem manajemen *bandwidth* konvensional atau manajemen *bandwidth* dinamis dengan logika *fuzzy* menghasilkan nilai packet loss dan jitter yang lebih baik dari pada tanpa adanya sistem manajemen *bandwidth*.

5.2 Saran

Berikut adalah saran yang dapat digunakan untuk mengembangkan penelitian lebih lanjut terkait manajemen *bandwidth* dinamis berbasis algoritma logika *fuzzy*.

1. Menerapkan manajemen *bandwidth* dinamis berbasis algoritma logika *fuzzy* pada jaringan Ad-Hoc yang menerapkan system *routing.*
2. Sistem manajemen *bandwidth* dinamis berbasis algoritma logika *fuzzy* dipasangkan pada perangkat yang memiliki kemampuan proses yang lebih baik dari pada Raspberry Pi 3 Model b.

Daftar Rujukan

[1] Asmuss, J., & Lauks, G., 2013. *A Fuzzy Logic Based Approach to Bandwidth Allocation in Network Virtualization*. Science and Information Conference (hal. 7). London: Science and Information.

[2] Sheu, S. T., & Chen, M. H., 1999. *A Fuzzy Bandwidth Allocation Controller to Support Real-time Traffic over Wireless Network*. IEEE Conference (hal. 5). Tamsui: IEEE.

[3] Slonowsky, D., Jiang, Q., & Srinivasan, R., 2002. *A Fuzzy Dynamic Bandwidth Re-Allocator*. *IEEE Canadian Conference* (hal. 8). Toronto: IEEE Canadian.

[4] Yi, C., Hao, L., Ge, G., & Ruimin, H., 2007. *A Resource Allocation Control for Wireless Ad Hoc Networks*. IEEE Conference (hal. 4). Wuhan: IEEE.