

DAFTAR SINGKATAN

VANET	: Vehicular Ad Hoc Network
TORA	: Temporary Ordered Routing Algorithm
NS-2	: Network Simulation-2
SUMO	: Simulation of Urban Mobility
OSM	: OpenStreetMap
RREQ	: Paket Route Request
RREP	: Paket Route Reply
CBR	: Constant Bit Rate

BAB I

PENDAHULUAN

1.1 Latar Belakang

Salah satu hal yang menyebabkan kemacetan adalah tidak adanya komunikasi antar kendaraan sehingga akan menyebabkan kemacetan. VANET adalah teknologi komunikasi data untuk kendaraan yang bergerak dengan kecepatan tinggi dengan menggunakan *wireless* berbasis *ad hoc*. Pada VANET suatu node memiliki karakteristik mobilitas yang sangat tinggi dan terbatas pada pola pergerakannya. Hal tersebut membuat topologi VANET lebih dinamis pergerakannya [1]. Studi mengenai VANET menjadi salah satu tema yang menarik dibidang jaringan nirkabel. Karena sifatnya yang fleksibel dan dinamis, efisiensi menjadi standar penting yang perlu diperhatikan dalam menjalankan komunikasi antar node. Selain itu, pergerakan node menjadikan topologi selalu berubah-ubah dan tentunya mempengaruhi keandalan pertukaran data yang terjadi antar node sumber dan node tujuan [2]. Mengacu pada hal tersebut, pemilihan *routing protocol* yang tepat dapat dikatakan sebagai salah satu solusi untuk mencapai kondisi jaringan yg efisien. Selalu ada kebutuhan setiap node dalam VANET untuk mencari jalur (*route*) terbaik dalam komunikasi data dari sumber ke tujuan. Sehingga pengetahuan mengenai karakteristik *routing protocol* tentunya akan sangat membantu dalam memilih jenis *routing protocol* yang memberikan kinerja lebih baik dalam suatu kondisi yang ditinjau dalam suatu jaringan *mobile ad hoc*. Penelitian mengenai *routing protocol* pada VANET telah banyak diajukan untuk mendukung komunikasi antar kendaraan. Pada *VANET* terdapat *routing protocol* yang berfungsi untuk menentukan rute komunikasi kendaraan sesuai dengan karakteristik pada VANET dengan kondisi pergerakan kendaraan yang dinamis. Salah satunya adalah *protocol routing* TORA (*Temporally Ordered Routing Algorithm*) merupakan salah satu jenis protokol reaktif. Protokol *routing* TORA bersifat adaptif dan bebas dari kemungkinan looping sehingga cocok untuk kondisi jaringan yang berubah-ubah dan menggunakan transmisi *broadcast*

untuk melakukan pengiriman data dari node sumber menuju node tujuan sehingga menyediakan beberapa rute untuk sampai ke tujuan [3]. Sehingga *protocol routing* TORA sesuai diterapkan pada VANET yang memiliki pergerakan yang dinamis. Dengan adanya jarak pengiriman data antara node sumber ke node tujuan, sehingga memiliki waktu delay yang akan diukur berdasarkan nilai performansi *average end to end delay* pada *protocol routing* TORA.

1.2 Rumusan Masalah

Adapun rumusan masalah dari Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana performansi *average end to end delay* berdasarkan perubahan jumlah node dan perubahan kecepatan node pada protokol *routing* TORA?
2. Bagaimana hasil dari performansi *average end to end delay* disesuaikan dengan karakteristik perkotaan dengan menggunakan *protocol routing* TORA?

1.3 Tujuan

Tujuan dari penelitian ini adalah untuk mengetahui performansi *protocol routing* TORA berdasarkan parameter yang diujikan yaitu *average end to end delay*. Hasil pengujian diketahui bahwa *protocol routing* TORA memiliki pengaruh terhadap perubahan mobilitas pada perkotaan.

1.4 Batasan Masalah

Adapun batasan masalah dari Tugas Akhir ini adalah sebagai berikut:

1. Metode *Routing* yang digunakan adalah TORA.
2. Skenario penelitian adalah pada daerah Bandung Indah Plaza dengan luas area 1000 Km.
3. Komunikasi node hanya membahas komunikasi antar kendaraan V2V (*vehicle-to-vehicle*).
4. Tidak membandingkan dengan protokol lain.
5. Pengujian yang dilakukan hanya pada kendaraan roda 4 (mobil).
6. *Network simulator* yang digunakan adalah SUMO, NS-2, OSM.

7. Parameter yang akan dianalisis adalah *average end to end delay*.
8. Tidak membahas mengenai keamanan jaringan.
9. Tidak ada *obstacle* (gedung, pohon dan lain-lain).

1.5 Metodologi Penelitian

1. Studi Literatur

Pada tahap ini dilakukan pencarian teori-teori VANET dan konsep yang berkaitan dengan *protocol routing TORA* dan hal-hal lain yang berhubungan dengan topik. Penggunaan simulator SUMO dan NS-2 di dalamnya.

2. Analisa Kebutuhan

Pada tahap ini adalah dilakukan analisis kebutuhan sesuai dengan kebutuhan sistem yang akan dibangun. Analisa tersebut dilakukan agar protokol dan simulator yang dibuat dapat berjalan dengan baik.

3. Perancangan Sistem

Pada tahap ini melakukan perancangan sistem pada *Network Simulator* akan digunakan sebagai simulasi kendaraan yang akan menghasilkan *mobility* dan *traffic* dengan mengambil map sesuai dengan studikusus yang diambil pada *openstreetmap.org*. Untuk mengetahui nilai-nilai parameter untuk mengetahui performansi dari *protocol routing TORA*.

4. Analisis dan Penarikan Kesimpulan

Pada tahap ini dilakukan analisis dari pengujian protokol yang telah dibuat dan dilakukan pengujian terhadap performansi *routing protocol* berdasarkan nilai *average end to end delay*. Hal ini bertujuan untuk mengetahui performansi pengiriman paket data pada protokol TORA pada simulator NS-2.

5. Penyusunan Laporan Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan tugas akhir dan dokumentasi yang diperlukan untuk tugas akhir ini.

1.6 Sistematika Penulisan

Penulisan laporan tugas akhir ini dibagi dalam beberapa bagian. Tiap-tiap bagian menjelaskan langkah demi langkah dalam pengerjaan tugas akhir ini. Berikut ini adalah bagian dari laporan tersebut :

BAB I PENDAHULUAN

Pada bab ini terdiri dari latar belakang penelitian, rumusan masalah, tujuan, batasan masalah, metode penyelesaian dan sistematika penulisan yang berkaitan dengan tugas akhir.

BAB II LANDASAN TEORI

Pada bab ini terdiri dari teori dari berbagai sumber yang digunakan untuk simulasi dan analisis yang berhubungan dengan sistem pada tugas akhir. Teori yang dimaksud mencakup pengukuran nilai performansi *average end to end delay* pada VANET.

BAB III PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada bab ini terdiri dari perancangan dan pemodelan sistem, kebutuhan perangkat keras dan lunak, alur pengerjaan dan penyelesaian sistem, serta *output* yang diperlukan untuk mendapatkan performansi protokol routing *TORA*.

BAB IV PENGUJIAN DAN ANALISIS SISTEM

Pada bab ini terdiri dari hasil implementasi pada pengujian dan analisis simulasi yang diperoleh dari data hasil pengujian performansi *averang end to end delay* yang dilakukan.

BAB V KESIMPULAN DAN SARAN

Pada bab ini berisi tentang kesimpulan akhir dari penelitian yang telah dirancang dan dilakukan pada bab satu hingga bab empat. Selain tujuan, pada bab ini terdapat saran yang dimaksudkan untuk pengembang dari penelitian ini dikemudian hari.

BAB II

LANDASAN TEORI

2.1 VANET (*Vehicular ad-Hoc Network*)

VANET (*Vehicular Ad Hoc Network*) adalah suatu jaringan *ad-hoc* yang digunakan untuk berkomunikasi antara kendaraan satu dengan kendaraan lainnya yang disebut dengan *vehicle-to-vehicle communication (V2V)*. VANET bersifat *autonomous* dan *self-organizing wireless*, artinya jaringan ini tidak bergantung pada infrastruktur jaringan manapun. Dalam suatu node memiliki karakteristik mobilitas yang sangat tinggi dan terbatas pada pola pergerakannya. Hal tersebut membuat topologi jaringan VANET lebih dinamis dibandingkan dengan MANET. Sebagian besar node di dalam VANET bersifat *mobile* (selalu bergerak) dan pertukaran informasi terjadi di jalan raya [4]. *Protocol routing* pada VANET memiliki dua model yaitu *routing reactive* yang membentuk *table routing* hanya saat dibutuhkan dan *routing proactive* yang melakukan pemeliharaan *table routing* secara berkala pada waktu tertentu.

Pergerakan node pada VANET berubah-ubah setiap saat, sehingga setiap node akan terus memperbaharui *table routing* sesuai dengan node lain. Perubahan node pada VANET menjadi permasalahan dalam pengiriman paket sehingga dibutuhkan informasi jarak antara node, kecepatan dan data transmisi.

2.2 Karakteristik VANET

VANET memiliki beberapa karakteristik khusus yang membedakannya dari jaringan *ad hoc* lainnya. Karakteristik tersebut antara lain [5]:

1. Topologi yang sangat dinamis

Topology jaringan di dalam *VANET* dapat berubah-ubah (dinamis) dengan cepat dikarenakan pergerakan kendaraan dalam kecepatan tinggi. Contohnya pada 2 mobil yang bergerak pada kecepatan 20 m/s dengan *range* transmisi 250 m, maka konektivitas di antara keduanya hanya akan bertahan selama $250/20 = 12.5$ detik.

2. Putusnya koneksi jaringan

Terputusnya koneksi antara 2 mobil disebabkan akibat tingginya kecepatan kendaraan pada saat pertukaran informasi dilakukan.

3. Model mobilitas dan presiksi

Pola mobilitas dari kendaraan bergantung dengan ruang lingkup lingkungan, seperti tingkat kemacetan, kecepatan kendaraan, akselerasi dan deakselerasi, pola mengendara dari pengemudi.

4. Daya baterai dan kapasitas *storage*

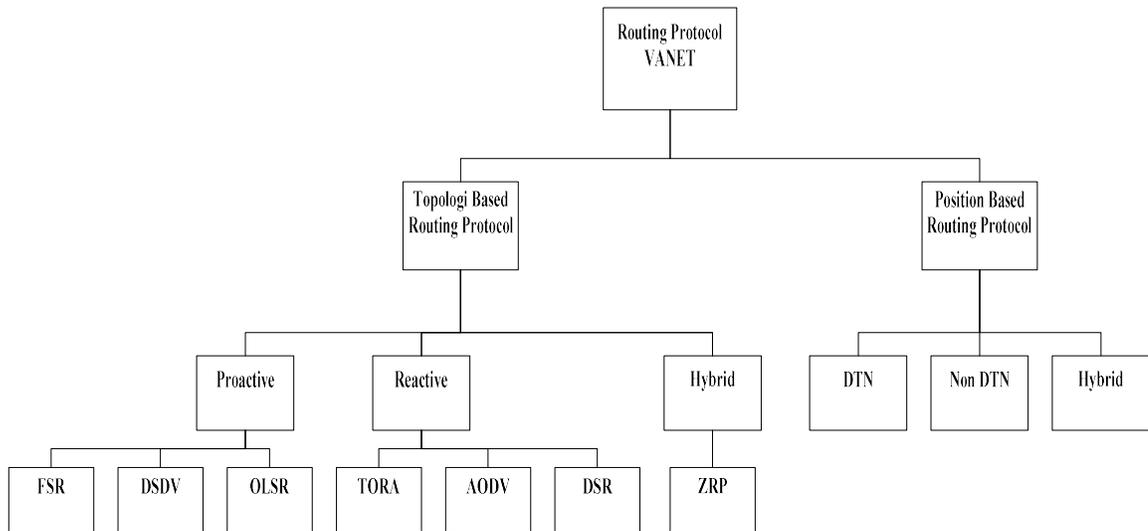
Kendaraan saat ini difasilitasi daya baterai dan kapasitas *storage* yang tidak terbatas. Hal inilah yang mendukung *VANET* sebagai sarana yang efektif dalam melakukan komunikasi dan penggunaan *routing protocol*.

5. Interaksi dengan *sensor on board* / posisi

Sensor on board digunakan untuk pengiriman posisi terkini dan pergerakan node yang akan membantu komunikasi yang lebih efektif dan keputusan rute.

2.3 Routing Protocol VANET

Routing adalah mekanisme penentuan *link* dari node pengirim ke node penerima yang bekerja pada layer 3 OSI (*Layer Network*). Protokol *routing* diperlukan karena untuk mengirimkan paket data dari node pengirim ke node penerima akan melewati beberapa node penghubung (*intermediate node*) dengan tujuan untuk mengatur komunikasi agar saling bertukar informasi, dimana protokol *routing* berfungsi untuk mencari *route link* yang terbaik dari link yang akan dilalui melalui mekanisme pembentukan tabel *routing*. Pemilihan route terbaik tersebut didasarkan atas beberapa pertimbangan seperti *bandwith link* dan jaraknya.



Gambar 2.1 Routing protokol pada VANET

Gambar 2.1 menjelaskan alur dari *routing protocol* yang terdapat pada VANET sesuai dengan karakteristik yang dimiliki berdasarkan *topologi-based routing* dan *position based routing*.

2.4 Jenis – jenis *Routing Protocol*

1. *Reactive Routing Protocol*

Reactive routing protocol atau disebut juga *on demand protocol* adalah salah satu jenis *routing* protokol yang tabel *routing* dibentuk jika ada permintaan pembuatan *route link* baru atau perubahan *link*. Saat node sumber membutuhkan routing ke node tujuan, node sumber melakukan proses *route discovery* dalam jaringan. Proses ini akan selesai jika rute telah ditemukan atau semua permutasi rute telah diperiksa. Setelah didapat rute maka akan dilakukan prosedur *routing maintenance* hingga node sumber tidak menginginkan lagi atau node tujuan tidak bisa diakses lagi.

2. *Proactive Routing Protocol*

Proactive routing adalah jenis *protocol* yang tabel routing dibentuk jika ada permintaan pembuatan route link baru atau perubahan link dan berusaha untuk menyediakan informasi routing yang konsisten serta *up-to-date* di setiap node. Setiap node diharuskan mempunyai satu atau lebih tabel untuk menyimpan informasi

routing. Setiap node merespon perubahan dalam topologi jaringan dengan menyalurkan update informasi tabel *routing* ke seluruh node di jaringan untuk memastikan konsistensi *routing*.

3. *Hybrid Routing Protocol*

Hybrid routing adalah merupakan kombinasi dari *distance vector* dan *link state routing*, dimana bekerja dengan cara berbagi pengetahuan dari seluruh jaringan dengan tetangga dan *link state routing* yang bekerja dengan *mengirim pada* setiap *router* pada jaringan tentang tetangga dekatnya. *Hybrid routing* adalah klasifikasi ketiga *routing* algoritma *hybrid* menggunakan *distance vector protocol* untuk matriks yang lebih akurat untuk menentukan jalur terbaik untuk jaringan tujuan, dan melaporkan informasi *routing* hanya bila terdapat perubahan dalam topologi jaringan. *Hybrid routing* memungkinkan konvergensi cepat namun memerlukan lebih sedikit pengolahan daya dan memori dibandingkan dengan *link state routing*.

2.5 **TORA (*Temporary Ordered Routing Algorithm*)**

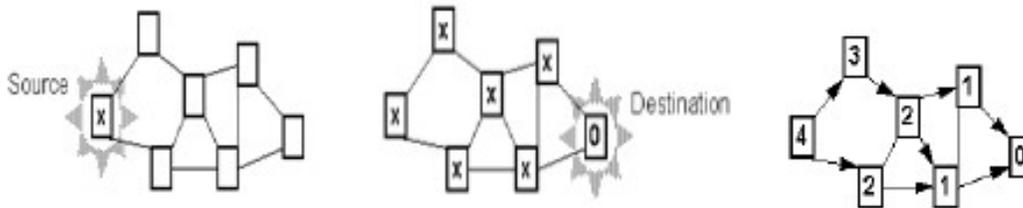
Temporally-Ordered Routing Algorithm (TORA) adalah *routing protocol* berbasis konsep *link reversal*. Keunggulan dan keunikan utama TORA adalah menyediakan *multi* rute ke node tujuan sehingga perubahan topologi jaringan tidak menjadi pengaruh besar kepada node dalam jaringan. *Multi* rute dapat dicapai karena terdapat *control message* dalam setiap kumpulan-kumpulan node, dimana setiap node dalam sebuah kumpulan tersebut hanya menjaga informasi routing tetangganya (*one hop neighbour*) saja. Protokol TORA mempunyai 3 peran utama, yakni membuat rute (*route creation*), mengelola rute (*route maintenance*) dan menghapus rute (*route erasure*). Optimalisasi rute (memilih rute yang terpendek) menjadi prioritas kedua pada TORA, karena rute yang panjang seringkali digunakan untuk menghindari overhead ketika menelusuri rute yang baru.

Tindakan yang diambil oleh TORA dapat diibaratkan seperti air yang mengalir menurun dari node sumber menuju node tujuan melalui jaringan pipa. Pipa menggambarkan hubungan antara node dalam jaringan, persimpangan dari pipa menggambarkan node, dan air yang mengalir dalam pipa menggambarkan paket data

yang mengalir ke arah tujuan. Setiap node memiliki ketinggian yang dihitung oleh *routing protocol* TORA. Ketinggian node ditetapkan lebih besar dari node tetangga manapun, sehingga jika pipa antara dua node tersumbat dan air tidak bisa lagi mengalir melalui jalur itu, maka air kemudian akan mengalir kembali keluar dari pipa yang tersumbat dan menemukan jalur alternatif lain menuju ke node tujuan.

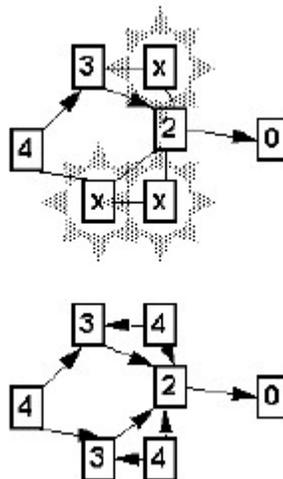
Ketika sebuah *node* membutuhkan rute ke tujuan tertentu, *node* tersebut akan mengirimkan *broadcast* sebuah paket permintaan rute yang berisi alamat tujuan. Paket ini di *broadcast* melalui jaringan hingga paket tersebut diterima oleh *node* tujuan atau *node* perantara yang sudah memiliki rute ke tujuan yang dibutuhkan. Setiap penerima paket permintaan kemudian *membroadcast* paket pembaruan daftar tinggi (*height*) yang berhubungan dengan tujuan (jika penerimanya adalah tujuan, ketinggiannya di *set* menjadi 0). Paket ini disebar kembali melalui jaringan, setiap *node* yang menerima *update* akan bernilai lebih besar dari ketinggian tetangga dari mana *update* diterima. Proses ini kemudian menghasilkan serangkaian jalur yang diarahkan dari *node* sumber ke *node* tujuan.

TORA bekerja berdasarkan *height* tiap node. *Height* dari suatu node menentukan path untuk tujuan yang dimaksud. TORA menggunakan DAG (*Directed Acyclic Graph*) yaitu jenis *graph direct* (langsung) tanpa adanya *directed cycles*. Jika *height* suatu node lebih besar dari tetangga A, link menuju A disebut *downstream*. Lainnya disebut *upstream*. Singkatnya, *height* dari setiap node dan link yang terhubung membentuk, jalur bebas loop, struktur *multipath routing* dimana semua node mengarah *downstream* [6].



Gambar 2.2 Proses permintaan *route routing* TORA

Gambar 2.2 menjelaskan bahwa ketika sebuah *node* menemukan rute ke tujuan yang tidak berlaku lagi, ketinggian akan kembali diperbarui sehingga nilai ketinggian maksimum kembali berkaitan dengan *node* tetangganya mengirimkan paket pembaruan.



Gambar 2.3 Proses permintaan *route* pada *protocol routing* TORA

Gambar 2.3 menjelaskan bahwa ketika sebuah *node* menemukan bagian dari jaringan secara fisik terpisah dari tujuan, *node* akan menghasilkan paket kembali dan mengulang *routing* serta menghapus rute yang tidak valid dari jaringan.

Keuntungan *Protocol* TORA :

1. TORA memiliki banyak opsi rute antara *source* node dan *destination* node. Tergantung dari kondisi topologi saat itu.
2. TORA tidak memerlukan *update* yang *periodic* maka dari itu *communication overhead* dan *bandwidth utilization* dapat diminimalisir.
3. TORA dapat mengawasi status dari tiap node yang ada pada jaringan.

Kekurangan dari *protocol routing* TORA yaitu lamanya waktu proses pengiriman dari node sumber ke node tujuan dikarenakan perubahan rute yang dilalui diakibatkan oleh kerusakan rute.

2.6 *Network Simulator (NS-2)*

Network Simulator merupakan perangkat lunak simulasi jaringan yang secara umum digunakan untuk riset dalam bidang penelitian jaringan dan menyediakan dukungan substansial untuk simulasi *routing*. Kelebihan dari *network simulator* diantaranya menjadi dukungan terhadap *multiple protocol* dan kemampuan untuk mempresentasikan data-data jaringan dalam bentuk grafik. *Network Simulator* dijalankan dengan menggunakan dua bahasa pemrograman, yaitu C++ dan *Object-oriented Tool Command Language (OTcl)*. Kelebihan dari bahasa C++ karena bahasa pemrograman tersebut lebih efisien dan sudah banyak dikenal. Mampu mendukung runtime simulasi yang cepat, meskipun melibatkan sejumlah packet dan sumber data dalam jumlah besar [7]. Hasil dari *network simulator* merupakan *file* berbentuk log data berekstensi “.tr”. *File log* ini dapat dihitung ataupun dianalisis menggunakan cara manual maupun menggunakan *file* lain yang disebut *awk script*.

Hasil yang dikeluarkan oleh ns-2 berupa *file trace*, harus diproses dengan menggunakan *tool* lain, seperti *Network Animator (NAM)*, perl, awk dan gnuplot. Hasil log *data* dapat dijelaskan dengan contoh sebagai berikut :

Tabel 2.1 Data file .tr

s	0.01335474	_1_	RTR	---	0	cbr	512	[0	0	0	0]	-----	[1: 0	5: 0	30 8]
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

Keterangan :

1. Aksi (s/r/d) = 8
 2. Waktu = 0.01335474
 3. Node sumber = 1
 4. Layer (AGT/RTR/LL/IFQ/MACPHY) = RTR (routing)
 5. Id paket = 0
 6. Tipe paket = cbr (Constant Bit Rate)
 7. Ukuran paket = 512
- [a b c d] = [0 0 0 0]
8. a = durasi header paket didalam layer mac
 9. b = mac sumber
 10. c = mac tujuan
 11. d = tipe mac didalam paket Flag
- [a:b c:d e f] = [1:0 5:0 30 8]
12. a = ip node sumber
 13. b = nomor port ip sumber
 14. c = ip node tujuan
 15. d = nomor port ip tujuan
 16. e = jumlah TTL ip header
 17. f = ip node selanjutnya

2.7 SUMO (*Simulation of Urban Mobility*)

SUMO (*Simulation of Urban Mobility*) merupakan simulator untuk mensimulasikan *traffic* dan *mobility* kendaraan dan lingkungan menyerupai keadaan *real*. SUMO termasuk simulator yang *open source* dan *release* pada tahun 2002, dimana ada dua alasan mengapa *open source* yaitu mendukung *traffic simulation* dengan *tool* gratis dengan algoritma masing-masing yang dapat diterapkan serta sumber terbuka dalam *traffic simulation*.

SUMO tidak hanya untuk *traffic simulation*, tetapi lebih dari aplikasi yang dapat membantu dalam mempersiapkan dan melakukan *traffic simulation*. Untuk