

Pemodelan dan Verifikasi Formal Protokol EE-OLSR dengan UPPAAL CORA

Rachmat Wahid Saleh Insani*¹, M. Reza M.I Pulungan²

¹ Prodi S2/S3 Ilmu Komputer, FMIPA UGM, Yogyakarta

³Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

e-mail: *rachmat.wahid.s@gmail.com, pulungan@ugm.ac.id

Abstrak

Sistem Information and Communication Technology (ICT) adalah suatu bagian hidup dari masyarakat yang sangat penting. Sistem ICT terus berkembang menjadi sistem yang kompleks dan besar. Protokol komunikasi adalah contoh sistem ICT yang digunakan oleh seluruh masyarakat pengguna Internet. Protokol OLSR adalah protokol komunikasi jaringan wireless yang bersifat proaktif, table-driven dan berbasis pada algoritma link-state. Protokol EE-OLSR adalah varian dari protokol OLSR yang dinyatakan mampu meningkatkan penggunaan energi tanpa adanya pengurangan pada performa.

Proses verifikasi protokol umumnya dilakukan dengan cara simulasi dan pengujian langsung. Namun proses tersebut tidak mampu memverifikasi bahwa tidak ada subtle error atau design flaw pada suatu protokol. Model checking merupakan metode algoritmik yang dijalankan secara fully-automatic untuk melakukan verifikasi pada sistem. UPPAAL adalah model checker tool untuk memodelkan, simulasi, dan verifikasi suatu sistem yang dimodelkan pada Timed Automata.

UPPAAL CORA adalah model checker tool untuk memverifikasi protokol yang telah dimodelkan ke bahasa pemodelan Linearly Priced Timed Automata, apakah protokol memenuhi properti energy efficient yang telah diformulasikan menggunakan bahasa spesifikasi formal dalam sintaks Weighted Computation Tree Logic. Teknik Model Checking terhadap protokol tersebut menghasilkan bukti bahwa protokol EE-OLSR memenuhi properti energy efficient hanya ketika lalu lintas pengiriman paket terjadi.

Kata kunci—pemodelan, verifikasi, EE-OLSR, UPPAAL CORA.

Abstract

Information and Communication Technology systems is a most important part of society. These systems are becoming more and more complex and are massively encroaching on daily life via the Internet and all kinds of embedded systems. Communication protocols are one of the ICT systems used by Internet users. OLSR protocol is a wireless network communication protocol with proactive, and based on link-state algorithm. EE-OLSR protocol is a variant of OLSR that is able to prolong the network lifetime without losses of performance.

Protocol verification process generally be done by simulation and testing. However, these processes unable to verify there are no subtle error or design flaw in protocol. Model Checking is an algorithmic method runs in fully automatic to verify a system. UPPAAL is a model checker tool to model, verify, and simulate a system in Timed Automata.

UPPAAL CORA is model checker tool to verify EE-OLSR protocol modelled in Linearly Priced Timed Automata, if the protocol satisfy the energy efficient property formulated by formal specification language in Weighted Computation Tree Logic syntax. Model Checking Technique to verify the protocols results in the protocol is satisfy the energy efficient property only when the packet transmission traffic happens.

Keywords— modeling, verification, EE-OLSR, UPPAAL CORA.

1. PENDAHULUAN

Mobile Ad Hoc Network (MANET) adalah jaringan komputer bersifat spontan, yang berkomunikasi melalui suatu media nirkabel. Setiap *node* dapat bergabung atau memisahkan diri kapan saja, serta bebas bergerak sesuai keinginan. MANET tidak memiliki infrastruktur terpusat, seluruh *node* yang berpartisipasi berfungsi sebagai *node* akhir dan *router*. Jaringan MANET memiliki sebuah topologi yang dinamis, dimana setiap *node* dapat bergerak secara acak, kondisi penyebaran radio berubah dengan cepat sepanjang waktu, serta *bandwidth* yang terbatas [1]. Sejak *mobile host* ditenagai oleh baterai, penggunaan energi baterai secara efisien sangatlah penting. Umur baterai dapat mempengaruhi performa komunikasi jaringan secara keseluruhan [2].

Optimized Link State Routing (OLSR) adalah protokol *routing* yang didasarkan pada *link-state algorithm* dan bersifat proaktif, protokol ini banyak digunakan dan dikembangkan sehingga berbagai macam studi terus dilakukan untuk memperbaiki serta meningkatkan kinerja protokol OLSR [3]. OLSR tradisional memberikan keunggulan dalam menemukan rute diantara dua *node* di dalam jaringan dalam waktu yang singkat dikarenakan polanya yang proaktif, namun OLSR dapat menghabiskan banyak sumber daya dalam proses pemilihan *node* MPR serta dalam pertukaran informasi Topology Control. Energy Efficient OLSR (EE-OLSR) adalah suatu protokol yang dinyatakan dapat meningkatkan tingkat efisiensi energi untuk memperpanjang umur jaringan tanpa adanya pengurangan pada performa [2].

Model Checking adalah suatu teknik verifikasi yang menyelidiki seluruh kemungkinan *system state* dengan cara *brute force* [4]. Model Checking juga merupakan sebuah metode algoritmik yang dijalankan secara fully-automatic untuk melakukan verifikasi pada sebuah sistem [5]. UPPAAL adalah suatu *model checker tool* yang digunakan untuk memodelkan, simulasi, dan melakukan verifikasi terhadap suatu sistem yang dimodelkan pada Timed Automata [6]. UPPAAL CORA adalah suatu varian dari UPPAAL untuk menganalisa Cost Optimal Reachability. UPPAAL CORA dikembangkan oleh tim yang telah mengembangkan UPPAAL sebelumnya. UPPAAL CORA menggunakan Linearly Priced Timed Automata (LPTA) sebagai bahasa pemodelannya. Penelitian ini dilakukan dengan memodelkan dan memverifikasi protokol EE-OLSR tersebut menggunakan *tool model checker* UPPAAL CORA.

2. METODE PENELITIAN

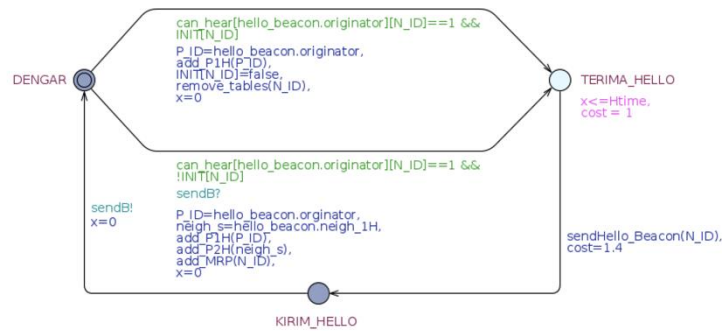
2.1 Priced Timed Automata

Priced Timed Automata (PTA), memiliki suatu definisi formal yaitu dimana X adalah kumpulan *clock*, dan Priced Timed Automata pada X adalah sebuah *annotated directed graph* dengan *vertex* yang dapat dibedakan satu sama lain, yang disebut dengan *initial location*. Untuk membentuk suatu PTA, ditambahkan catatan pada *edge* dengan *cost* yang dibutuhkan untuk melakukan transisi, serta *location* dengan *cost rate* yang menunjukkan berapa *cost* yang dihabiskan ketika protokol berada di *location* tersebut.

Definisi 1. Secara formal, suatu PTA $A = (L, E, J, \mathcal{L}, \mathcal{C})$

Dimana L adalah himpunan *location*, E adalah sekumpulan *edge*, J menetapkan invariant pada *location*, \mathcal{L} merupakan *labelling function*, dan \mathcal{C} adalah *cost* pada *location* dan *edge*.

Suatu contoh model Priced Timed Automata pada pemrosesan pesan HELLO ditunjukkan oleh Gambar 1.



Gambar 1 Contoh Priced Timed Automata

2.2 Weighted Computation Tree Logic

Properti dari Priced Timed Automata diformalisasikan oleh rumus Weighted Computation Tree Logic (WCTL) [7].

Definisi 2. Jika diberikan AP sebagai himpunan *atomic proposition*, sintaks *logic discrete* WCTL adalah suatu *grammar*, yakni

$$\varphi ::= \sigma | \pi | \neg\varphi | \varphi \vee \varphi | \exists 0\varphi | \varphi \exists U\varphi | \varphi \forall U\varphi | z_i \cdot \varphi$$

2.3 UPPAAL CORA

UPPAAL CORA adalah sebuah cabang dari UPPAAL untuk menganalisis Cost Optimal Reachability. Jika diberikan suatu model PTA, UPPAAL CORA memberikan informasi *optimal path* menuju suatu *state* yang memenuhi beberapa kondisi tujuan. Disini makna *optimal* adalah *path* dengan akumulasi *cost* yang paling rendah.

Penggunaan *tool model checker* UPPAAL CORA adalah untuk memverifikasi protokol EE-OLSR yang telah dimodelkan ke dalam bahasa pemodelan PTA, apakah suatu protokol dapat memenuhi properti *energy efficient* yang telah diformulasikan menggunakan bahasa spesifikasi formal melalui sintaks Weighted Computation Tree Logic (WCTL). UPPAAL CORA dapat menemukan *path* yang memiliki *cost* (pemakaian energi) paling minimal pada setiap proses di protokol EE-OLSR yang dimodelkan ke dalam bahasa pemodelan PTA.

2.4 Energy Efficient Optimized Link State Routing Protocol (EE-OLSR)

Protokol EE-OLSR merupakan varian dari protokol OLSR. Optimized Link State Routing Protocol (OLSR) adalah protokol routing yang berbasis pada algoritma *link-state* dan bersifat proaktif. Protokol ini melakukan pertukaran pesan secara berkala untuk menjaga informasi topologi jaringan di setiap *node*. OLSR memadatkan ukuran informasi yang dikirimkan didalam pesan kemudian mengurangi jumlah transmisi ulang untuk menyebarkan pesan tersebut di dalam jaringan.

Protokol OLSR bekerja dengan mengirimkan pesan kontrol ke seluruh *node* didalam jaringan melalui sekumpulan *node* MPR. *Node* MPR adalah sekumpulan *node* tetangga yang berjarak 1 *hop* yang bertugas untuk melakukan *forwarding* pesan yang diterima dari *node* MPR Selector. *Node* MPR Selector adalah *node* yang memilih sekumpulan *node* tersebut sebagai MPR.

Energy Efficient-OLSR (EE-OLSR) adalah suatu modifikasi dari protokol *routing* OLSR yang ditujukan untuk memperpanjang umur *node* di dalam jaringan. Modifikasi tersebut berada pada mekanisme pemilihan MPR yang berdasarkan pada konsep *Willingness*, untuk memperpanjang umur hidup jaringan tanpa kehilangan performa dalam hal *throughput*, *end-to-end delay* atau *overhead*, pengecualian pada pemakaian energi yang disebabkan oleh *overhearing* yang dapat memperpanjang umur *node* tanpa mengurangi fungsi OLSR, serta ditambahkan suatu *energy-aware metric* pada proses pemilihan MPR dan pemilihan rute antara

sumber dengan tujuan, yakni Minimum Drain Rate (MDR). Pesan kontrol yang disebarakan ke seluruh jaringan di protokol OLSR ada dua tipe, yakni pesan HELLO dan pesan Topology Control.

Pesan HELLO adalah pesan yang ditujukan agar *node* dapat mengetahui informasi *node* tetangga yang berjarak 1 *hop* darinya. Setiap *node* didalam jaringan mengirimkan pesan HELLO setiap 2 detik. Pesan HELLO tidak boleh di-*forward* oleh *node* penerima. Pesan HELLO berisi informasi mengenai: Message Type, tipe pesan, Originator Address, alamat ip dari *node* yang membuat pesan, Message Valid Time, jangka waktu hingga pesan ini dinyatakan tidak valid. One-hop neighbors of the Originator, alamat *IP node* tetangga dari *node* pembuat pesan.

Pesan TC adalah pesan kontrol yang ditujukan untuk mengetahui informasi rute serta informasi topologi. Setiap *node* melakukan *broadcast* pesan TC setiap 5 detik. Pesan TC harus disebarakan ke seluruh *node* didalam jaringan. Setiap pesan TC berisikan informasi, yakni: Message Type, tipe pesan, Message Originator, alamat *IP* pembuat pesan, Message Sequence Number, angka *integer* yang menunjukkan apakah pesan yang diterima adalah yang terbaru atau tidak, Time to Live, jumlah *node* didalam jaringan yang belum menerima pesan, Hops, jumlah *hop* dari pembuat pesan ke *node* penerima pesan, Validity Time, jangka waktu hingga pesan ini dinyatakan tidak *valid*, dan Advertised Neighbors Main Address, alamat *IP* dari *node-node* tetangga 1 *hop* dari pembuat pesan.

Protokol EE-OLSR memiliki sejumlah modifikasi pada mekanismenya, yakni Energy Aware Packet Forwarding, Mekanisme Pengaturan EA-Willingness, dan Pengecualian Overhearing. Energy Aware Packet Forwarding merupakan mekanisme yang terjadi pada proses pemilihan *next hop* untuk penyampaian paket data, dimana dilakukan suatu *energy-aware metric* yang ditujukan untuk memisahkan proses pemilihan MPR dari mekanisme pemilihan rute. *Energy-aware metric* tersebut adalah Minimum Drain Rate (MDR). Pengaturan Energy Aware Willingness adalah suatu mekanisme yang melibatkan pertimbangan energi pada pemilihan MPR. Pada saat suatu *node* melakukan transmisi paket *unicast* ke *node* lain, transmisi tersebut akan di-*overhear* oleh seluruh *node* tetangga dari *node* pengirim. Sehingga, *node-node* tetangga tersebut menghabiskan sejumlah energi walaupun paket tersebut tidak ditujukan pada mereka. Pengecualian Overhearing adalah suatu mekanisme dimana *node* yang berada pada jaringan dengan protokol EE-OLSR akan mengubah statusnya menjadi non-aktif ketika terjadi transmisi paket *unicast* di lingkungan sekitar *node* tersebut, sehingga *node* sekitar dari pasangan *node* yang sedang dalam transmisi pengiriman paket tidak menghabiskan energi karena *overhear* paket tersebut.

2.5 Asumsi Pemodelan

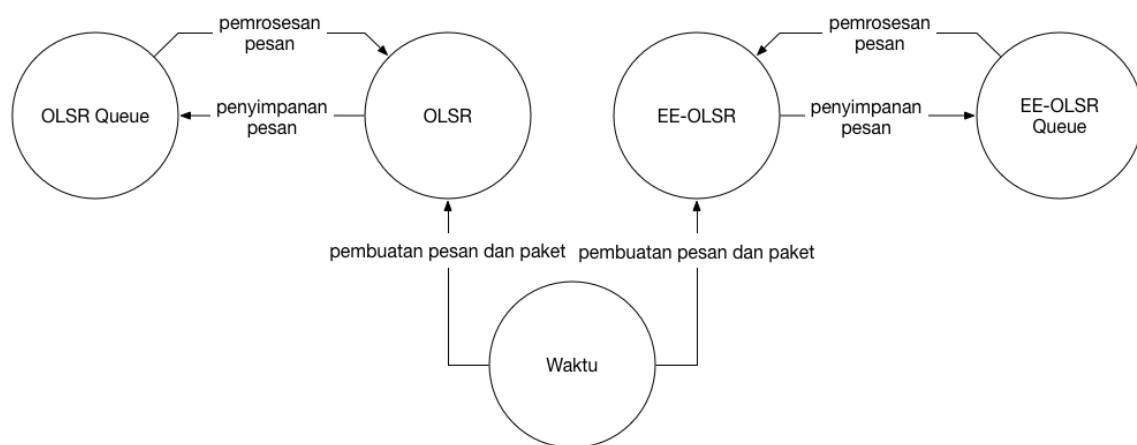
Setiap model merupakan sebuah abstraksi dari *real system*. Oleh karena itu, model yang dibangun harus mampu dijadikan sebuah asumsi skenario yang baik. Penelitian yang melibatkan Model Checking untuk membuat suatu model protokol komunikasi harus menyertakan suatu asumsi yang kuat terhadap sistem yang dibangun. Abstraksi yang diambil juga harus menitikberatkan pada protokol itu sendiri.

Sejumlah batasan juga diberikan agar model yang dibangun dapat terhindar dari keterbatasan di bidang Model Checking, yakni *state space explosion*. Penelitian sebelumnya membuktikan bahwa Model Checking terhadap protokol yang berjalan di MANET memiliki keterbatasan, yakni jumlah *node* yang dimodelkan tidak lebih dari 10 *node* [8]. Asumsi lain yang diperlukan adalah *link* antar *node* bersifat *bi-directional*. Secara umum, komunitas MANET membuat asumsi tersebut dengan tujuan untuk menyederhanakan protokol komunikasi dengan tidak menyertakan *link* yang *uni-directional* [9]. Penelitian ini bertujuan untuk memverifikasi apakah protokol EE-OLSR mampu memiliki umur *node* yang lebih panjang dari protokol OLSR tradisional melalui teknik Model Checking. Oleh karena itu keseluruhan jaringan terdiri dari sekumpulan *node* yang statis, dimana *link failure* tidak muncul. Kekurangan

yang sangat penting di sistem MANET bahkan dapat ditemukan pada skenario dengan jumlah node yang tidak banyak [10].

2.6 Hubungan antar Model

Pada penelitian ini, model yang dibuat terdiri dari 5 model, yakni model Waktu, OLSR, OLSR Queue, EE-OLSR, dan EE-OLSR Queue. Hubungan antar model ditunjukkan oleh Gambar 2. Model Waktu merepresentasikan kapan waktu yang diperlukan untuk mulai melakukan pembuatan pesan dan paket. Model EE-OLSR merepresentasikan perilaku *node* yang menggunakan jaringan protokol EE-OLSR. Model EE-OLSR Queue berfungsi untuk menyimpan informasi pesan maupun paket yang diterima oleh node dengan jaringan protokol EE-OLSR. Model OLSR menunjukkan perilaku dari *node* yang berada pada jaringan protokol OLSR. Model OLSR Queue berfungsi untuk menyimpan informasi pesan dan paket yang diterima oleh *node* protokol OLSR, serta membuat paket.



Gambar 2 Hubungan antar model

Model Waktu merepresentasikan suatu jadwal pengiriman pesan kontrol, yakni pesan HELLO dan pesan TC. Pengiriman tersebut dilakukan sesuai dengan *interval* masing-masing pesan. Model *node* EE-OLSR merepresentasikan perilaku utama dari *node* yang beroperasi di jaringan protokol EE-OLSR. Model EE-OLSR Queue bertugas untuk menyimpan pesan maupun paket yang diterima oleh *node* yang berjarak 1-hop dari pembuat pesan. Setiap pesan yang masuk akan disimpan di dalam suatu antrian. Model *node* OLSR merepresentasikan perilaku utama dari *node* yang beroperasi di jaringan protokol OLSR. Model OLSR Queue bertugas untuk menyimpan pesan maupun paket yang diterima oleh *node* pada protokol OLSR yang berjarak 1-hop dari pembuat pesan.

Secara detail, proses ini dimulai dengan seluruh *node* melalui model EE-OLSR dan OLSR melakukan pembuatan pesan HELLO ketika *clock* di model Waktu mencapai 2 detik. Model Waktu ditunjukkan oleh Gambar 3, model EE-OLSR ditunjukkan oleh Gambar 4, sedangkan model EE-OLSR Queue ditunjukkan oleh Gambar 5. Kemudian, *node* melakukan *broadcast* pesan tersebut ke seluruh *node* tetangga yang berjarak 1 hop darinya, proses *broadcast* pesan ini menghabiskan sejumlah energi, yakni energi *transmit*. Pesan HELLO yang diterima oleh suatu *node*, akan disimpan di dalam Queue. *Node* menghabiskan sejumlah energi ketika menerima pesan, dan dinyatakan dengan energi *receive*. Apabila *node* tersebut sedang berada dalam keadaan *idle*, maka EE-OLSR Queue dan OLSR Queue akan mengirimkan pesan tersebut menuju model EE-OLSR dan OLSR untuk diproses. *Node* akan melakukan pemeriksaan terhadap tipe pesan. Apabila tipe pesan adalah pesan HELLO, *node* akan melakukan *update* terhadap *routing table* bahwa *node* pembuat pesan tersebut berjarak 1 hop darinya. Pesan HELLO hanya dapat diterima oleh *node* yang berjarak 1 hop dari *node* pembuat

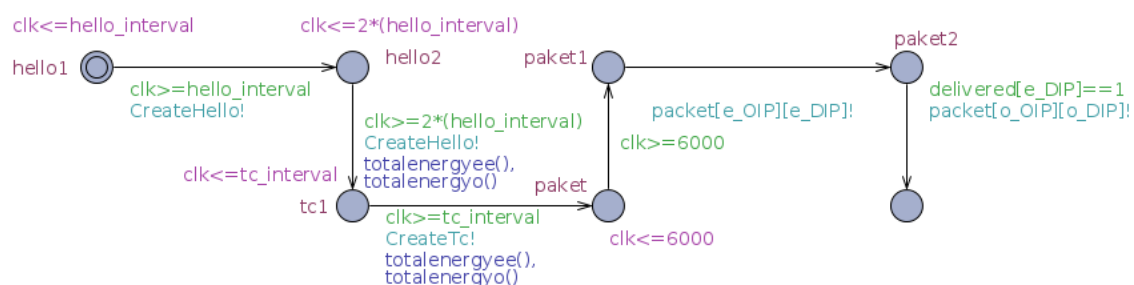
pesan. Setelah *node* melakukan update terhadap *routing table*, maka pesan tidak akan di-*forward* dan kemudian pesan dihapus.

Ketika *node* dalam keadaan *idle*, *node* akan melakukan pengiriman pesan TC setiap 5 detik. Pesan TC dikirimkan ke *node* yang berjarak 1 hop dari *node* pembuat pesan. Kemudian pesan TC tersebut di-*forward* kembali oleh *node* penerima ke *node* tetangga 1 hop darinya. Pesan TC akan di-*forward* hingga pesan tersebut tersebar di seluruh *node* didalam jaringan. Pesan TC akan disimpan di dalam Queue dari *node* penerima pesan. Pesan TC diproses ketika *node* tersebut sedang berada dalam keadaan *idle*, kemudian *node* melakukan pemeriksaan terhadap tipe pesan, apabila tipe pesan adalah pesan TC, maka pesan mulai diproses sesuai dengan tipe pesan. *Node* akan menghapus pesan apabila *IP node* penerima sama dengan *IP node* pembuat pesan. Jika *IP node* tidak sama dengan *IP node* pembuat pesan, dan jika informasi *hops* didalam pesan lebih kecil dari entri *hops* di *routing table*, maka *node* melakukan *update* terhadap *routing table*. Jika pesan telah di-*forward* sebelumnya, yang ditunjukkan oleh informasi *forwarded* = 1, dan entri *hops* didalam pesan lebih besar sama dengan jumlah seluruh *node* dikurangi 1, serta informasi Time to Live pesan kurang dari sama dengan 1 maka pesan TC tersebut akan di-*forward* menuju *node* tetangga yang berjarak 1 hop dari *node* tersebut. Pesan TC akan berhenti di-*forward* apabila pesan tersebut pernah di-*forward* oleh *node* tersebut, atau entri *hops* didalam pesan lebih besar dari jumlah *node* di jaringan selain *node* pengirim, artinya pesan TC telah mengunjungi seluruh *node* di jaringan. Pesan TC juga tidak di-*forward* apabila Time to Live pada pesan kurang dari 1.

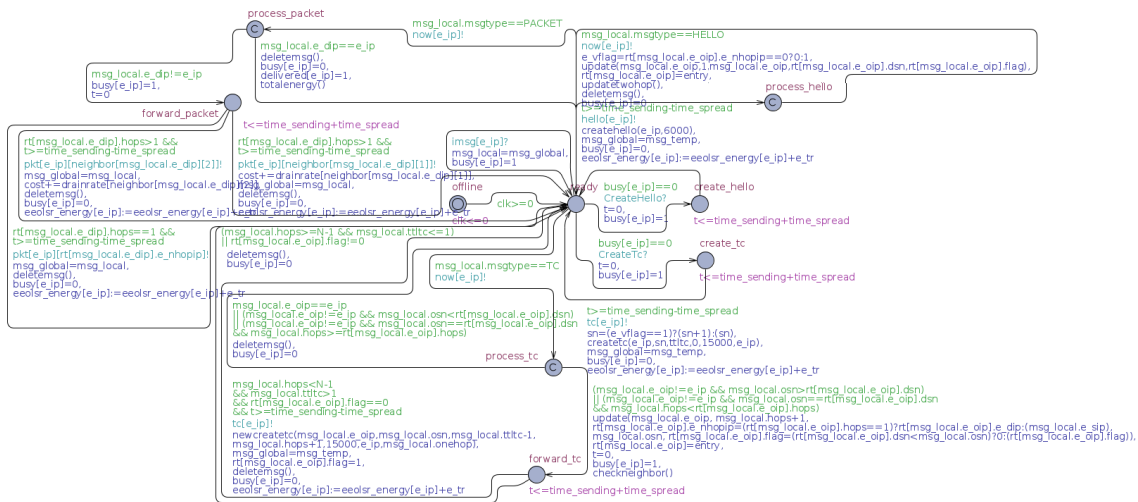
Suatu paket mulai dikirim ketika *clock* telah mencapai angka 7000 milidetik. Ketika *clock* telah mencapai angka tersebut, seluruh *node* di jaringan telah mengetahui topologi seluruh *node* lainnya. Paket akan dibuat oleh model EE-OLSR Queue dan OLSR Queue. Setiap model tersebut melakukan *assign IP* pengirim sebagai *ipk* dan *IP* penerima paket sebagai *ipt*, informasi tersebut disimpan di paket. Kemudian, EE-OLSR Queue dan OLSR Queue memasukkan paket ke dalam antrian pesan. Apabila *node* sedang berada dalam keadaan *idle* maka kedua model Queue akan mengirimkan paket menuju model EE-OLSR dan OLSR. Apabila *node* yang menerima paket memiliki *IP* yang tidak sama dengan *ipt* di informasi paket, maka *node* tersebut bukanlah *node* tujuan, artinya paket akan di-*forward*.

Model EE-OLSR akan mencari *path* yang memiliki *node* dengan *drain rate* terendah untuk dijadikan sebagai *node* berikutnya yang harus menerima *forwarding* paket, informasi *node* ini terdapat di *routing table* yang disebut sebagai *nhopip*. Mekanisme ini dilakukan oleh UPPAAL CORA, karena *tool* ini menentukan dan mengambil transisi dengan nilai *cost* terendah secara otomatis. Sedangkan model OLSR memperoleh informasi *nexthopip* melalui pembaharuan *routing table* pada saat pesan TC diterima.

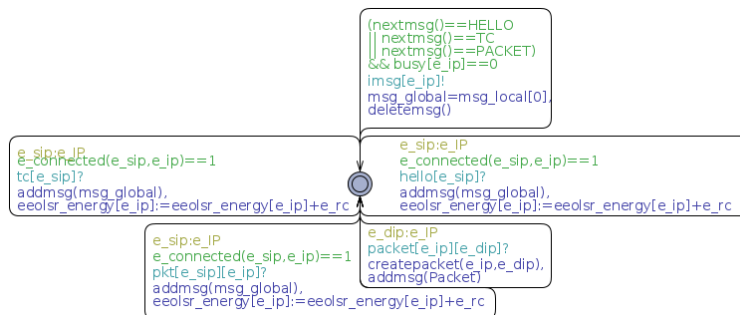
Kemudian, model EE-OLSR dan OLSR bersinkronisasi dengan model Queue dari *node* tetangga yang berjarak 1 hop dari *node* penerima paket yang memiliki *IP* yang sama dengan entri *nhopip* di *routing table* *node* tersebut. Apabila *IP node* sama dengan *dip* di informasi paket, maka paket berhasil dikirim.



Gambar 3 Model Waktu



Gambar 4 Model EE-OLSR



Gambar 5 Model EE-OLSR Queue

3. HASIL DAN PEMBAHASAN

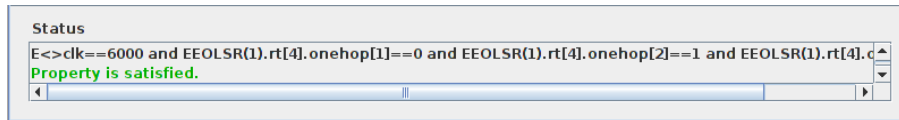
Pada tahap ini dilakukan serangkaian percobaan untuk membuktikan bahwa model yang telah dibuat berhasil melakukan sejumlah urutan proses utama yang ada di protokol. Proses-proses tersebut antara lain seperti:

Neighbor Detection, pada proses ini, seluruh *node* mengetahui informasi *IP* dari node tertangganya. Populating 2-hop Neighbor. *Node* mengetahui informasi *IP* dari node tetangga yang berjarak 2 hop darinya. Hasil verifikasi proses Neighbor Detection tersebut ditunjukkan oleh Gambar 6.



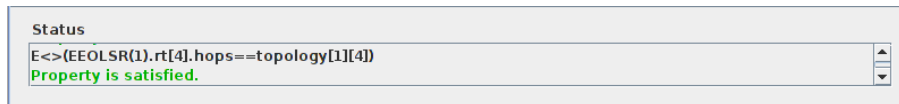
Gambar 6 Hasil verifikasi Neighbor Detection

Topology Discovery, *node* mengetahui informasi *node* tetangga dari seluruh *node* di jaringan. Hasil verifikasi proses Topology Discovery ditunjukkan oleh Gambar 7.



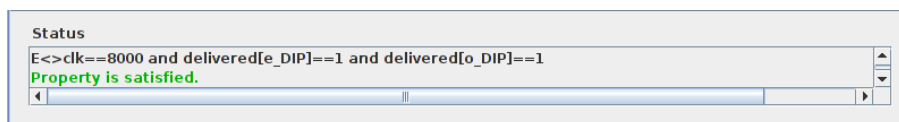
Gambar 7 Hasil verifikasi Topology Discovery

Optimal Route. Setiap node memperoleh informasi path terpendek antara *node* tersebut dengan *node* lain. Hasil verifikasi properti Optimal Route ditunjukkan oleh Gambar 8.



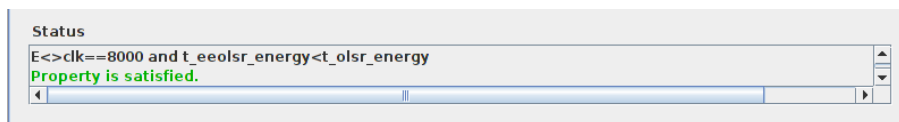
Gambar 8 Hasil verifikasi Optimal Route

Packet Delivery, suatu paket yang dikirim oleh suatu *node* dapat diterima di node yang ditujukan untuk paket tersebut. Hasil verifikasi properti Packet Delivery ditunjukkan oleh Gambar 9.



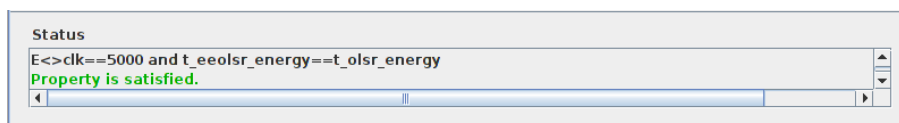
Gambar 9 Hasil verifikasi Packet Delivery

Energy Efficient. Energi yang dihabiskan oleh seluruh *node* di protokol EE-OLSR lebih kecil daripada energi yang dihabiskan oleh seluruh node di protokol OLSR. Hasil verifikasi properti *energy efficient* tersebut ditunjukkan oleh Gambar 10.

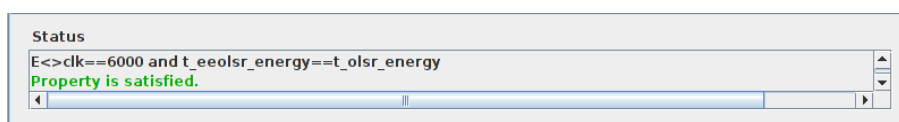


Gambar 10 Hasil verifikasi Energy Efficient

Penelitian ini juga melakukan percobaan perhitungan energi ketika *node* hanya melakukan pertukaran pesan kontrol, yakni pesan HELLO dan TC. Hasil verifikasi energi pada saat itu, kedua protokol menghabiskan energi dengan nilai yang sama. Hasil verifikasi proses energy efficient pada pemrosesan pesan kontrol tersebut ditunjukkan oleh Gambar 11 dan 12.



Gambar 11 Hasil verifikasi energi ketika pertukaran pesan HELLO



Gambar 12 Hasil verifikasi energi ketika pertukaran pesan TC

Tabel 1 menunjukkan hasil dari verifikasi-verifikasi yang telah dilakukan pada penelitian ini.

Tabel 1 Daftar hasil verifikasi protokol

Waktu (milidetik)	Proses pada protokol	Hasil verifikasi
2000	Neighbor Detection	<i>Satisfy</i>
4000	Populating 2 Hop Neighbor	<i>Satisfy</i>
5000	Topology Discovery	<i>Satisfy</i>
7000	Optimal Route	<i>Satisfy</i>
7000	Packet Delivery	<i>Satisfy</i>
7000	Energy Efficient	<i>Satisfy</i>

4. KESIMPULAN

Verifikasi protokol EE-OLSR yang dilakukan dengan memodelkan protokol menjadi tiga Priced Timed Automata dan diverifikasi secara otomatis menggunakan *tool model checker* UPPAAL CORA. Berdasarkan hasil verifikasi, protokol EE-OLSR mampu memenuhi properti *energy efficient* ketika lalu lintas pengiriman paket dilakukan. Namun, properti tersebut tidak dipenuhi apabila *node* masih melakukan pemrosesan pesan. Energi yang dihabiskan oleh kedua protokol pada saat pemrosesan pesan kontrol bernilai sama.

5. SARAN

Pada penelitian yang selanjutnya diharapkan mampu melakukan verifikasi protokol lain dengan properti lainnya. Pembahasan terhadap penggunaan PTA untuk memodelkan dan memverifikasi suatu protokol diharapkan dapat digunakan untuk penelitian terhadap protokol-protokol lainnya, terutama protokol yang memiliki ketergantungan aspek waktu dan *cost* berdasarkan spesifikasinya.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Allah swt, kedua orang tua, keluarga, dan para sahabat yang telah memberi dukungan pada penulis terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] Wibling, O., Parrow, J. dan Pears, A., 2004. Automated Verification of Ad Hoc Routing Protocols. *Formal Techniques for Networked and Distributed Systems*, 325, pp.343–358.
- [2] Rango, F. De, Fotino, M. dan Marano, S., 2008. EE-OLSR: Energy Efficient OLSR routing protocol for Mobile ad-hoc Networks. *MILCOM 2008 - 2008 IEEE Military Communications Conference*, pp.1–7.
- [3] Jacquet, P. dan Clausen, T., 2001. Optimized link state routing protocol for ad hoc networks. ... , 2001. *Ieee Inmic ...*, 2001, pp.62–68. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=995315.

-
- [4] Baier, C. dan Katoen, J.-P., 2008. *Principles Of Model Checking*, The MIT Press. Available at: <http://mitpress.mit.edu/books/principles-model-checking>.
- [5] Berard, B. et al., 2010. *System and Software Verification: Model Checking Techniques and Tools*, Springer Publishing Company.
- [6] Larsen, K.G., Pettersson, P. & Yi, W., 1997. Uppaal in a nutshell. *International Journal on Software Tools for Technology Transfer*, 1(1-2), pp.134–152.
- [7] Brihaye, T., Bruyère, V. dan François Raskin, J., 2004. Model-Checking for Weighted Timed Automata, in *In Proceeding of FORMATS-FTRTFT'04, Lect. Notes Comput. Sci.* 3253 , 277–292, Springer, pp. 277–292.
- [8] Höfner, P. dan Mciver, A., 2013. Statistical Model Checking of Wireless Mesh Routing Protocols. In *NASA Formal Methods*. pp. 322–336.
- [9] Prakash, R., 2001. A routing algorithm for wireless ad hoc networks with unidirectional links. *Wireless Networks*, 7(6), pp.617–625.
- [10] Fehnker, A., van Glabbeek, R., Hofner, P., McIver, A., Portmann, M., dan Tan, W. L., 2011. Modelling and Analysis of AODV in UPPAAL. In *1st International Workshop on Rigorous Protocol Engineering*.