Mengenal OMNet++ dan Algoritma Dijkstra

Ricky
Teknik Komputer, Fakultas Teknologi Informasi
Institut Teknologi Batam
Batam, Indonesia
1922022@student.iteba.ac.id

Muhamad Arie
Teknik Komputer, Fakultas Teknologi Informasi
Institut Teknologi Batam
Batam, Indonesia
1922021@student.iteba.ac.id

***Abstract*—Omnet++ merupakan aplikasi dasar berbasis open source yang dituliskan menggunakan c++ untuk mendesain simulasi dan topologi jaringan. Disini kami akan mencoba mendesain sebuah topologi jaringan berisi 6 node dan dengan menggunakan algoritma Dijkstra, kita akan mencari jarak tersingkatnya.**

***Index Terms*—Simulation, Networking, Collision, Detection, Error, Correction**

I. Pengenalan

Tugas ini bertujuan untuk memberikan pengalamanan kepada mahasiswa untuk menggunakan OMNeT++. Tutorial ini memandu mahasiswa dalam membuat dan bekerja dengan contoh model simulasi, dan menunjukkan beberapa fitur OMNeT ++ yang umum digunakan. Tutorial ini didasarkan pada simulasi contoh Tictoc, yang dapat ditemukan di direktori samples/tictoc dari instalasi OMNeT++, sehingga mahasiswa dapat langsung mencoba contoh tersebut. Dan juga terdapat simulasi contoh Aloha yang dapat ditemukan di direktori samples/aloha. Versi OMNeT: 5.6.2. Lokasi file sumber: tictoc1.ned, txc1.cc, omnetpp.ini

II. Teori dan Tools

Simulasi jaringan adalah teknik di mana program perangkat lunak memodelkan perilaku jaringan dengan menghitung interaksi antara entitas jaringan yang berbeda (router, sakelar, node, titik akses, tautan, dll.). Kebanyakan simulator menggunakan simulasi kejadian diskrit - pemodelan sistem di mana variabel status berubah pada titik waktu tertentu. Perilaku jaringan dan berbagai aplikasi serta layanan yang didukungnya kemudian dapat diamati di lab uji; berbagai atribut *environment* juga dapat dimodifikasi dengan cara yang terkontrol untuk menilai bagaimana jaringan / protokol akan berperilaku dalam kondisi yang berbeda.

Emulasi jaringan memungkinkan pengguna untuk memperkenalkan perangkat dan aplikasi nyata ke dalam jaringan uji (disimulasikan) yang mengubah aliran paket sedemikian rupa untuk meniru perilaku jaringan langsung. Lalu lintas langsung dapat melewati simulator dan dipengaruhi oleh objek dalam simulasi.

Metodologi tipikal adalah paket nyata(*live packet*) dari aplikasi langsung dikirim ke server emulasi (tempat jaringan virtual disimulasikan). Paket sebenarnya **dimodulasi** menjadi paket simulasi. Paket simulasi didemodulasi menjadi paket nyata setelah mengalami efek kehilangan, kesalahan, penundaan, jitter, dll., Dengan demikian mentransfer efek jaringan ini ke paket nyata. Dengan demikian, seolah-olah paket nyata mengalir melalui jaringan nyata tetapi pada kenyataannya mengalir melalui jaringan yang disimulasikan. Emulasi digunakan secara luas dalam tahap desain untuk memvalidasi jaringan komunikasi sebelum penerapan.

Tools atau alat yang kami gunakan disini adalah *fresh installed OMNet++* dengan samples bawaan yang sudah disediakan didalam folder instalasi.

A. OMNeT++

OMNeT++ adalah kerangka kerja simulasi jaringan peristiwa diskrit modular berorientasi objek. Memiliki arsitektur generik, sehingga dapat (dan telah) digunakan di berbagai domain masalah:

- Permodelan jaringan komunikasi kabel dan nirkabel
- Permodelan protokol
- permodelan jaringan antrian
- Permodelan multiprosesor dan sistem perangkat keras terdistribusi lainnya
- Memvalidasi arsitektur perangkat keras
- Mengevaluasi aspek kinerja sistem perangkat lunak yang kompleks
- Secara umum, pemodelan dan simulasi sistem apa pun di mana pendekatan kejadian diskrit cocok, dan dapat dengan mudah dipetakan ke dalam entitas yang berkomunikasi dengan bertukar pesan.

B. Algoritma Dijkstra

Dijkstra adalah algoritma yang digunakan untuk mencari lintasan terpendek pada sebuah graf berarah. Contoh penerapan algoritma ini adalah lintasan terpendek yang menghubungkan antara dua kota ataupun titi berlainan tertentu.

III. Pengerjaan

Bagian ini akan membahas sedikit tentang cara kami mengerjakan, mulai dari metodologi hingga langkah-langkahnya:

A. Metodologi

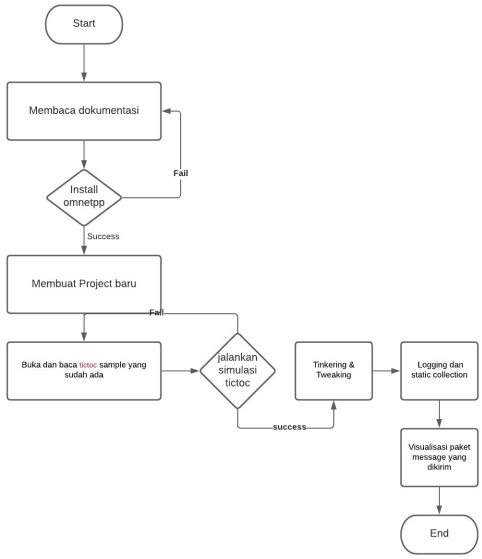
Pada dasarnya tidak ada metode spesifik yang kami terapkan di *Hands on* ini, kami hanya mengikuti *official documentation* dari *OMNet++*.

B. Langkah Instalasi

1) Buka *official site OMNet++*, lalu kunjungi bagian *Introduction*. Baca sekilas pengenalan dengan *OMNet++*.
2) Download terlebih dahulu *OMNet++* disini
3) *Extract zip* yang sudah didownload ke direktori yang diinginkan(nantinya akan menjadi direktori instalasi *OMNet++*
4) Buka direktori hasil *extract* sebelumnya lalu *double-click file* `mingwenv`, jalankan perintah `./configure` lalu tunggu hingga selesai. Setelah itu jalankan perintah `make` dan tunggu hingga selesai.
5) Setelah itu kita dapat membuka IDE dengan mengetikkan perintah `omnetpp` di terminal `mingwenv`. Dan proses instalasi selesai

untuk lebih lengkapnya tentang proses instalasi dapat mengikuti petunjuk disini

C. Flowchart

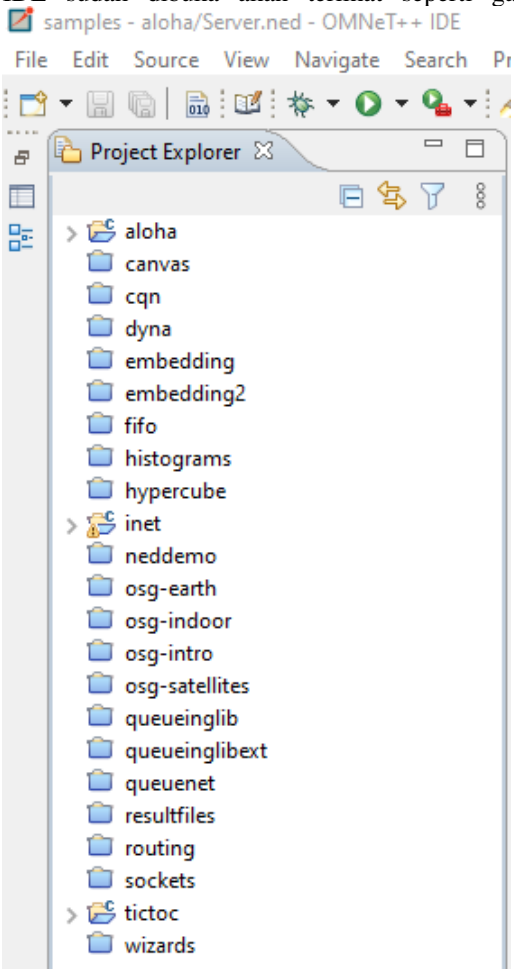


IV. Pembahasan

Bagian ini akan menjelaskan secara singkat mengenai tiap bagian-bagian dari *Tutorial Learn OMNeT++ with TicToc*. Namun kita tidak perlu mengikuti seluruh bagian yang dijelaskan pada halaman ini, seperti contohnya membuat *sample* dan menambahkan *NED file*. Dikarenakan saat kita menginstal sudah disediakan seluruh contohnya pada direktori *samples/* beserta penjelasannya.

A. Part 1: Getting Started

Model pertama yang akan dikerjakan pada tutorial ini adalah model *tic-toc*, dimana terdapat sebuah node yang saling mengirim dan menerima paket secara berulang. Ketika IDE sudah dibuka akan terlihat seperti gambar berikut,

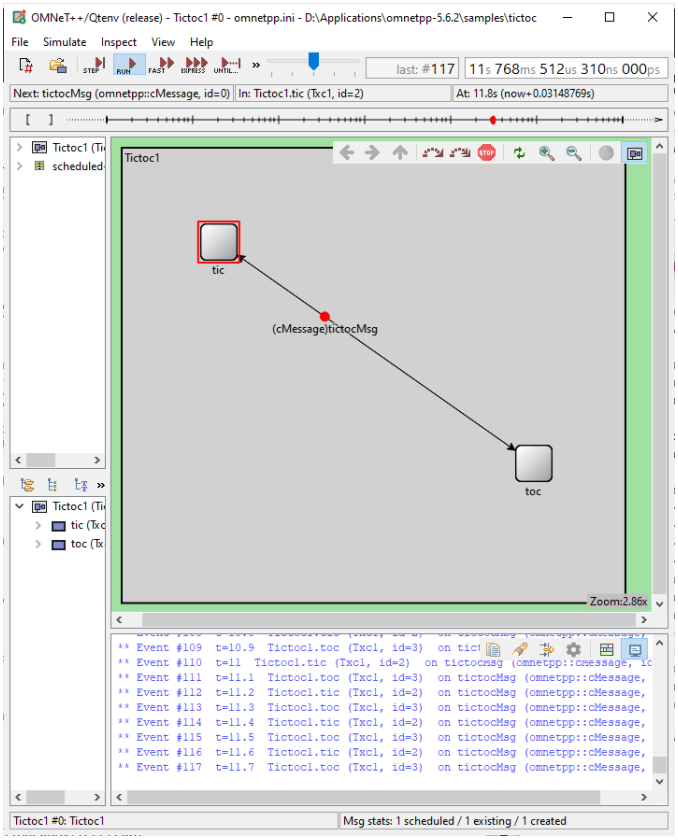


lalu buka *tictoc* folder, didalamnya terdapat banyak sekali file. Pilih `tictoc1.ned`.

B. Part 2: Running the Simulation

Untuk menjalankan simulasinya, kita dapat menekan tombol Run

Akan terlihat sebuah *console window* yang akan menjalankan proses *Compiling*. Tunggu beberapa saat lalu akan muncul *window* baru.

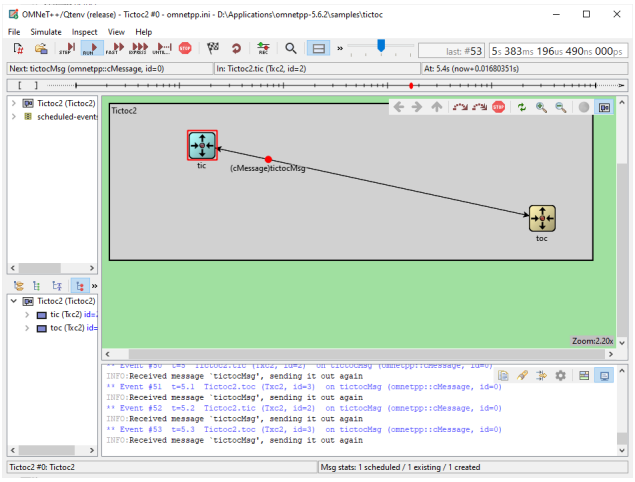


Dibagian ini juga kita dapat melakukan beberapa hal lainnya, seperti *Debugging* dan *Logging*, dapat juga memvisualisasikan menggunakan *Sequence Chart*.

C. Part 3: Enhancing the 2-node TicToc

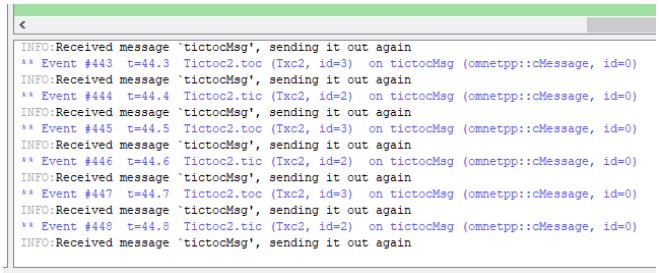
Pada bagian ini kita dapat meningkatkan *2-node TicToc* menggunakan *icon, logging, state variables, parameters, timers, etc*.

Untuk melihat contoh implementasi *icon*, dapat membuka dan menjalankan *file* `tictoc2.ned`.



Files: `tictoc2.ned`, `txc2.cc`

untuk melihat contoh implementasi icon, silahkan buka *file* `tictoc3.ned`. Jalankan simulasi dan klik kanan pada *node*, klik menu `Open Component Log for 'tic'`. Akan muncul log seperti gambar berikut

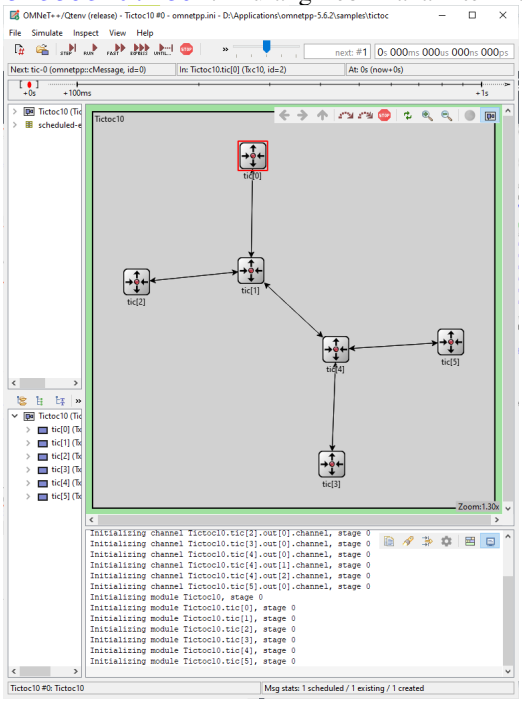


selain itu terdapat beberapa *file sample* yang dapat kita coba:

- Parameters; *Files:* `tictoc4.ned`, `txc4.cc`
- Timeout, cancelling timers; *Files:* `tictoc8.ned`, `txc8.cc`
- etc

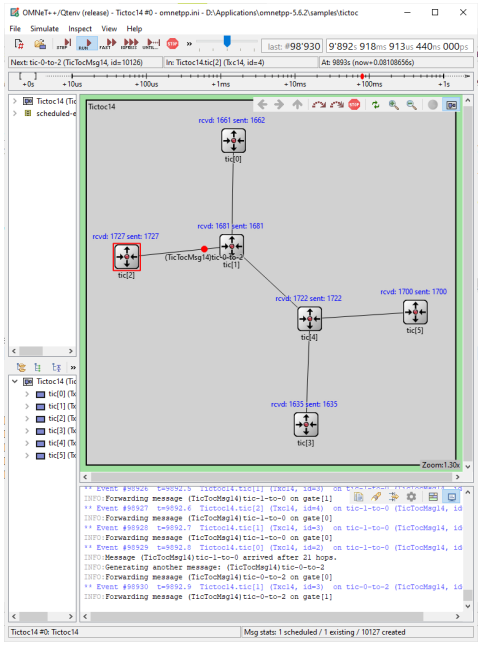
D. Part 4: Turning it Into a Real Network

Bagian ini akan menampilkan contoh implementasi lebih dari *2 nodes*. Dapat membuka dan menjalankan simulasi *file* `tictoc10.ned` . Kurang lebih akan terlihat seperti ini.



E. Part 5: Adding Statistics Collection

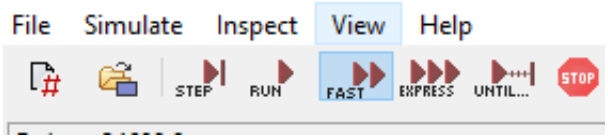
Dari simulasi tersebut kita dapat mengumpulkan statistik data antar node juga. Seperti yang sudah diimplementasi di `txc14.cc`.



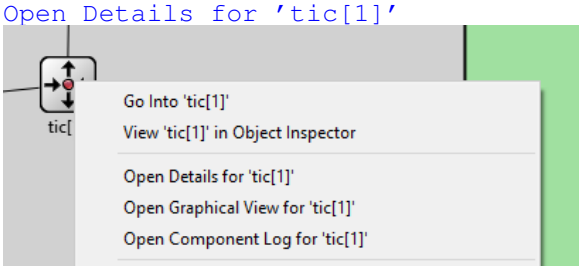
Files: `tictoc14.ned`, `tictoc14.msg`, `txc14.cc`

Kita dapat menampilkan Histogram data dengan cara sebagai berikut:

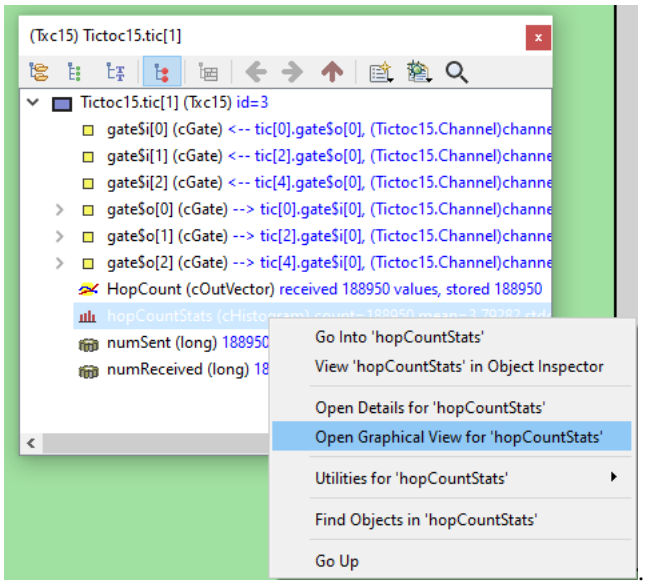
1) Buka dan jalankan simulasi *sample file* `tictoc15.ned` dengan mode `Fast` seperti gambar berikut



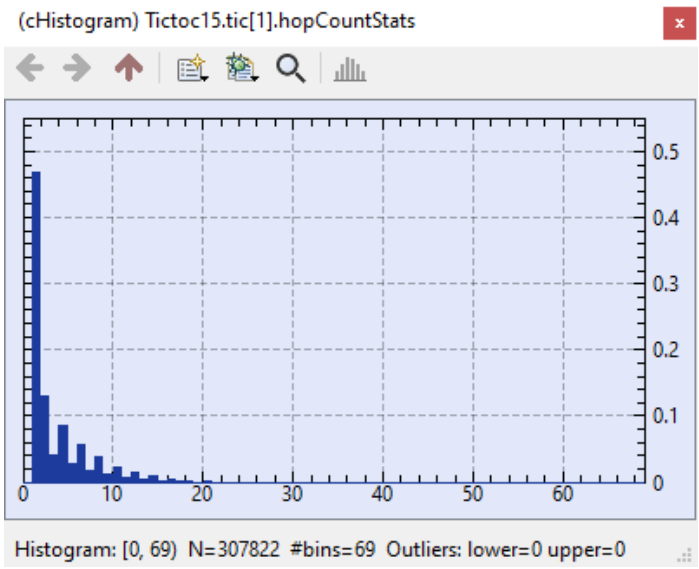
2) Klik kanan pada `tic[1]`, lalu `Open Details for 'tic[1]'`



3) Setelah itu klik kanan lagi pada `hopCountStats`, lalu pilih `Open Graphical View for 'hopCountStats'`.



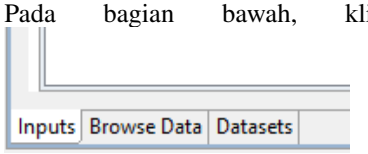
Maka akan muncul Histogramnya seperti ini



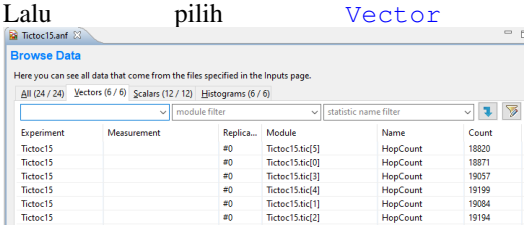
F. Part 6: Visualizing the Results With the IDE

Hasil dari *static collection* dapat kita visualisasikan dalam berbagai model *chart*. Berikut cara-caranya:

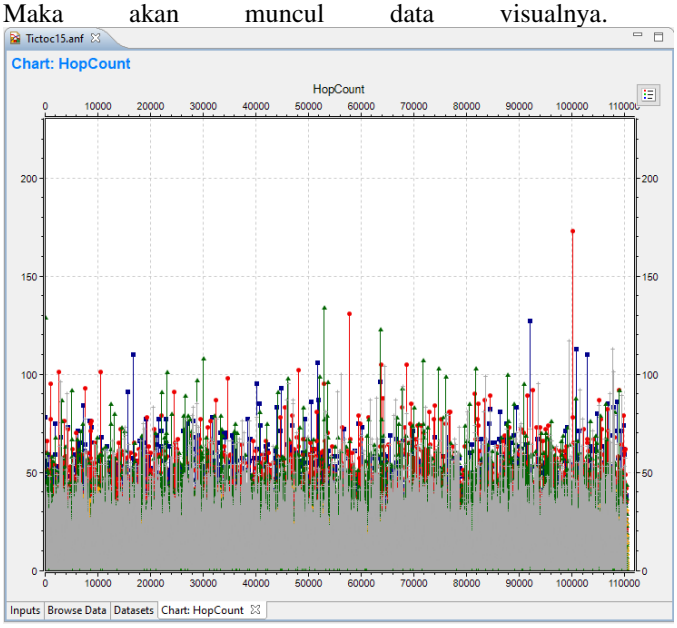
1) Buka *sample file* `tictoc/results/Tictoc15-#0.sca`
2) Pada bagian bawah, klik `Browse Data`



3) Lalu pilih `Vector` data



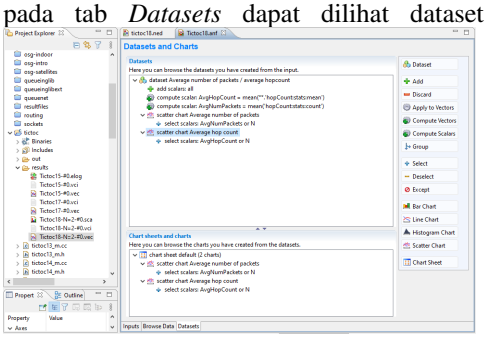
4) Lalu *Select all*, klik kanan lalu `Plot`. Maka akan muncul data visualnya.



5) Pada *background chart*, klik kanan `Properties`. Pada tab `Lines` kita dapat memilih *Line type* dan *Symbol type*. Pilih `Dots`
6) Selanjutnya *Display Legend* dengan cara, pilih menu `Legend` *Display Legend*. *Position above* dan *Anchoring north*. Klik *Ok*

G. Part 7: Parameter Studies

Kita dapat menganalisis data yang sudah di *record* dengan membuka *file* yang berektensi *.anf* , lalu pada tab *Datasets* dapat dilihat dataset yang terdaftar.



V. Kesimpulan

Kita dapat simpulkan bahwa simulasi jaringan berguna untuk meniru jaringan nyata sehingga kita dapat merencanakan rancangan, melakukan *debugging*, dan juga analisis jaringan.

References

[1] https://doc.omnetpp.org/omnetpp/InstallGuide.pdf
[2] https://youtu.be/u4T02tZuKrQ
[3] https://docs.omnetpp.org/tutorials/tictoc
[4] https://doc.omnetpp.org/omnetpp/UserGuide.pdf