JURNAL KEJURUTERAAN
http://www.eng.ukm.my

Penspesifikasian dan Pengesahsahihan Formal Protokol CSMA/CD Menggunakan Z

Zarina Shukur[1], Nursyahidah Alias[1], Bahari Idrus[1] dan Mohd Hazali Mohamed Halip[2]

[1]Jabatan Sains Komputer
[2]Jabatan Komputeran Industri
Fakulti Teknologi dan Sains Maklumat,
Universiti Kebangsaan Malaysia
43600 UKM Bangi, Selangor,
Malaysia
E-mail: zs@ftsm.ukm.my

Received Date: 23th April 2007 Accepted Date: 27th May 2009

ABSTRAK

Kertas ini membincangkan tentang penspesifikasian dan pengesahsahihan formal ke atas protokol CSMA/CD. Penspesifikasian dilakukan ke atas nod dalam rangkaian dan situasi dalam implementasi bas bagi protokol CSMA/CD. Satu jenis asas, empat jenis bebas, satu pemboleh ubah sejagat, dua skema keadaan dan sembilan skema operasi yang mewakili protokol CSMA/CD telah berjaya dibangunkan menggunakan bahasa formal Z. Kemudian, pengesahsahihan dilakukan ke atas spesifikasi tersebut dengan menggunakan teknik pembuktian teorem yang disokong oleh alatan pembukti Z/EVES. Sembilan teorem telah dibangunkan berdasarkan kepada sembilan operasi yang telah dispesifikasikan. Dari kajian didapati bahasa Z berupaya untuk menspesifikasikan protokol komunikasi. Selain itu, penggunaan alatan sokongan dalam proses pembuktian dapat menjimatkan masa dan tenaga, dan kecenderungan ralat berlaku adalah kecil.

Kata kunci: Spesifikasi formal, Z, Z/EVES, protokol rangkaian.

ABSTRACT

This paper discusses the formal specification and validation for CSMA/CD protocol. The Z specification language is used to specify a node in a network and a situation in a bus implementation for CSMA/CD protocol. One basic type, four free types, one global variables, two state schemas and nine operation schemas that represent CSMA/CD protocol have been specified by using the Z language. The specification has been validated by using theorem proving techniques supported by Z/EVES theorem prover. Nine theorems have been identified based on the nine specified operations. This study has shown that, Z has the ability to specify a communication protocol. Beside that, the usage of support tools during a proving process can save time dan energy, and reduce error-prone.

Keywords: Formal specification, Z, Z/EVES, network protocol.

PENGENALAN

Protokol ialah set peraturan yang mentakrifkan cara sesuatu servis yang terlibat dalam penghantaran dan penerimaan data dilaksanakan. Set peraturan perlulah dinyatakan dengan betul untuk menjamin penghantaran dan penerimaan data yang betul. Data tersebut mungkin data penting seperti data yang terlibat dalam aplikasi yang dilaksanakan oleh pengguna seperti pembayaran bil, kad kredit secara atas talian atau dokumen yang penting seperti surat rasmi.

Kebiasaannya dalam protokol rangkaian, teknik tidak formal banyak digunakan dan dapat mereka bentuk protokol yang besar dengan jayanya, tetapi terdapat ralat yang tidak dijangka dan tidak dikehendaki dalam kelakuan bagi kebanyakan protokol (Bochmann & Sunshine 1983). Ini kerana teknik tidak formal dalam amalan kejuruteraan perisian tidak menyokong penilaian seperti pengesahbetulan atau pengesahsahihan pada peringkat awal oleh pengguna atau pelanggan (Quemad 2004). Sebaliknya, teknik formal dapat digunakan untuk memberi kemudahan dalam mereka bentuk protokol yang betul (Bochmann & Sunshine 1983) bermula daripada fasa awal kitar hayat pembangunan.

Teknik formal meliputi penspesifikasian formal dan pengesahsahihan formal. Penggunaan kaedah formal dalam spesifikasi adalah untuk memahami keperluan pelanggan, mengenalpasti kekaburan, ketidakkonsistenan dan ketidaklengkapan dalam spesifikasi pada peringkat awal fasa pembangunan, dan memudahkan untuk diselenggara (Wing 1990; Giunchiglia & Traverso 2000).

Pengesahsahihan formal boleh dilakukan menggunakan teknik pembuktian teorem. Pembuktian teorem ialah proses untuk mendapatkan bukti bagi ciri sistem daripada aksiom yang telah didefinisikan. Langkah dalam pembuktian memerlukan aksiom dan peraturan taakulan yang telah didefinisikan untuk mendapatkan kesimpulan bagi pembuktian tersebut. Pembuktian boleh dilakukan secara manual dan menggunakan alatan pembukti teorem (Clarke & Wing 1996; NASA ARC 2002). Pembuktian manual berkemungkinan menghasilkan kesilapan lebih tinggi. Pembukti teorem sangat berguna kerana kesilapan yang berlaku adalah kecil (Sremack tanpa tarikh).

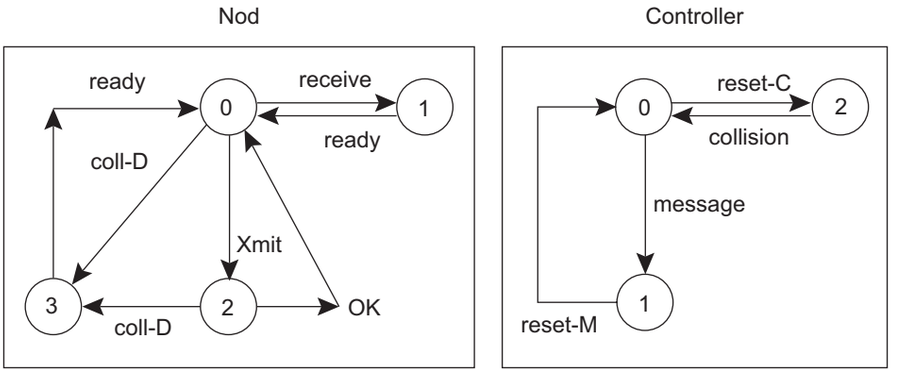
Teknik formal biasanya digunakan untuk menjamin tahap keselamatan perisian. Namun tidak semua perisian sesuai menggunakan teknik ini kerana faktor kos dan matlamat. Hanya perisian yang memerlukan jaminan elemen selamat dan teguh perlu diberi perhatian. Dari segi perisian berasaskan protokol, terdapat beberapa kajian yang menggunakan teknik formal seperti yang disenaraikan dalam (Zarina dan Nursyahidah 2007a) bagi protokol *authentication,* (Zarina dan Nursyahidah 2007b) bagi protokol *Secure Electronic Transaction* dan (Ainita dan Zarina, 2009) bagi protokol e-Voting. Manakala satu contoh penggunakan teknik formal ke atas protokol komunikasi lain iaitu SACK adalah seperti yang kami bincangkan dalam (Zarina et al. 2007c).

Kajian ini menjelaskan berkenaan penspesifikasian dan pengesahsahihan formal terhadap protokol CSMA/CD. Kajian seumpama ini penting dilakukan khasnya untuk memperoleh pensijilan paras tertinggi jaminan keselamatan iaitu E6 bagi suatu produk perisian oleh *Information Technology Security Evaluation Criteria* (ITSEC 1991). Kertas ini diorganisasikan kepada lima bahagian. Bahagian dua membincangkan tentang pendekatan dan metod yang digunakan dalam kajian ini. Bahagian tiga menjelaskan tentang hasil kajian ini. Manakala perbincangan tentang hasil kajian ini dijelaskan dalam bahagian empat. Bahagian lima ialah kesimpulan bagi kajian ini.

PENDEKATAN DAN METOD

Dalam kajian ini protokol yang telah dipilih untuk disahsahih ialah protokol komunikasi CSMA/CD. CSMA/CD (IEEE/ANSI 1985) dikenali sebagai piawaian 802.3 ialah protokol yang menyediakan kaedah capaian medium untuk digunakan dalam rangkaian setempat. Spesifikasi formal bagi CSMA/CD dibangunkan menggunakan bahasa spesifikasi Z (Spivey 1998). Z ialah bahasa spesifikasi formal yang dibangunkan di Universiti Oxford pada akhir 1970 (Bowen 2005) dan telah dipiawaikan pada tahun 2002 (ISO/IEC 2002). Bahasa Z berasaskan kepada set teori dan logik tertib pertama.

Spesifikasi Z bagi protokol CSMA/CD yang dibangunkan dalam kajian ini adalah berdasarkan kepada model System of Communicating Machines (SCM) bagi protokol tersebut yang dibangunkan oleh Lundy dan Miller (1993). Model SCM bagi CSMA/CD ini dapat dilihat seperti Rajah 1.



Rajah 1. Spesifikasi bagi nod rangkaian dan Controller dalam model SCM Sumber: Lundy dan Miller 1993

Rajah 1 menunjukkan spesifikasi bagi nod rangkaian dan *Controller*. Nod rangkaian mewakili satu mesin keadaan. Manakala *Controller* mewakili situasi dalam implementasi bus, iaitu panjang mesej digunakan untuk memastikan setiap nod dalam rangkaian dapat mengesan mesej tersebut dan isyarat akan disebarkan sehingga ke hujung pengakhiran bus. *Controller* digunakan untuk mengosongkan medium pada masa tertentu. *Controller* hanya digunakan untuk tujuan pemodelan dan tidak diperlukan dalam piawaian CSMA/CD.

Nod rangkaian mempunyai lima operasi iaitu:

- Operasi *Xmit*: operasi yang menunjukkan nod sedang menghantar mesej.
- Operasi *OK*: operasi yang menunjukkan nod selesai menghantar mesej.
- Operasi *coll-D*: operasi yang menunjukkan nod mengesan pelanggaran.
- Operasi *ready*: operasi yang menunjukkan nod bersedia untuk menghantar mesej.
- Operasi *receive*: operasi yang menunjukkan nod sedang menerima mesej.

Operasi bagi nod menyatakan perubahan yang berlaku dalam penimbal nod dan isyarat yang dijana oleh nod tersebut. Manakala Controller pula mempunyai empat operasi iaitu:

- Operasi *message*: operasi yang menunjukkan penghantaran mesej sedang berlaku dalam medium.
- Operasi *reset-M*: operasi yang menunjukkan penghantaran mesej dalam medium telah selesai.
- Operasi *collision*: operasi yang menunjukkan pelanggaran sedang berlaku dalam medium.
- Operasi *reset-C*: operasi yang menunjukkan pelanggaran dalam medium telah selesai.

Operasi bagi *Controller* menyatakan perubahan yang berlaku dalam medium dan isyarat yang disebarkan oleh medium tersebut.

Spesifikasi Z CSMA/CD yang telah dibangunkan, kemudiannya disahsahih secara formal menggunakan teknik pembuktian teorem. Pengesahsahihan formal yang dilakukan ke atas spesifikasi menggunakan teknik pembuktian teorem dikenali sebagai *proof of correctness* (Wing 1998). *Proof of correctness* menggunakan logik formal untuk membuktikan dengan nilai input yang diberi, nilai output adalah benar (Haggarty 2002). Bagi pembuktian pada paras spesifikasi, sesuatu spesifikasi formal dikatakan benar jika dibuktikan lengkap dan konsisten. Kelengkapan dan kekonsistenan bagi spesifikasi Z dapat dinyatakan dengan membuktikan aspek keadaan awal, prakeadaan dan sifat luaran atau ciri spesifikasi tersebut (Barden et al. 1994). Setiap aspek tersebut diwakilkan dalam bentuk teorem untuk dibuktikan. Kertas ini mendemonstrasikan pembuktian ke atas aspek prakeadaan sahaja.

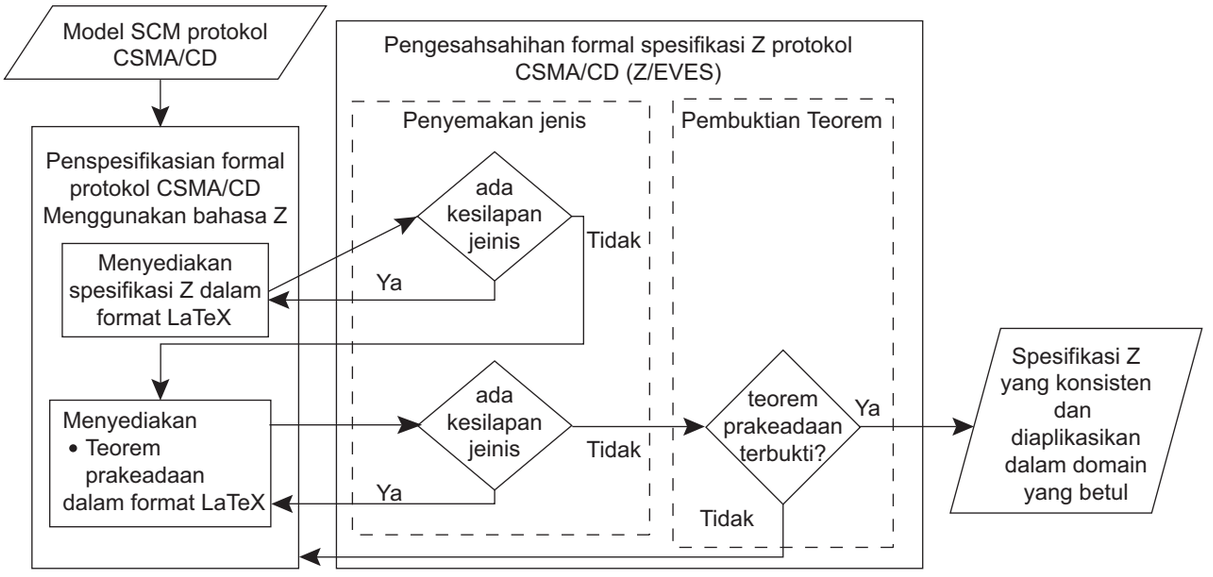
Dalam kajian ini, pembuktian dilakukan menggunakan alatan pembukti teorem Z/ Eves. Z/Eves dipilih untuk digunakan kerana ia adalah salah satu pembukti teorem yang telah diperakui, bersifat mesra pengguna, dan disenggara secara profesional. Alatan ini juga boleh dilarikan dalam pelbagai persekitaran seperti Unix, SunOS, Linux dan Windows (WetStone Technologies 1999).

Rajah 2 menunjukkan pendekatan yang digunakan dalam kajian ini bagi proses penspesifikasian dan pengesahsahihan protokol CSMA/CD.

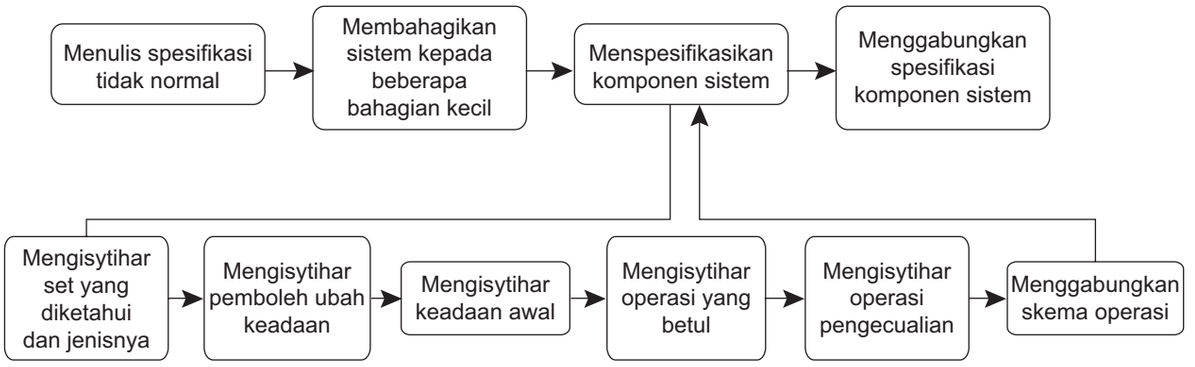
Merujuk Rajah 2, penspesifikasian formal protokol CSMA/CD melibatkan penyediaan spesifikasi Z dan teorem yang perlu dibuktikan menggunakan bahasa spesifikasi formal Z dalam format LaTeX. Teorem yang perlu dibuktikan ialah teorem yang akan digunakan dalam proses pengesahsahihan formal iaitu teorem prakeadaan. Teorem tersebut juga perlu dispesifikasi dalam notasi Z. Spesifikasi Z disediakan dalam format LaTeX menggunakan perisian penyunting PCTeX. Perisian tersebut dapat memudahkan penulisan notasi-notasi matematik dalam dokumen. Proses dan langkah-langkah yang terlibat dalam pembangunan spesifikasi Z boleh diperincikan seperti yang ditunjukkan dalam Rajah 3 (Sommerville 1995).

Manakala, pengesahsahihan formal spesifikasi Z protokol CSMA/CD melibatkan proses penyemakan jenis dilakukan ke atas spesifikasi Z yang telah dihasilkan daripada proses penspesifikasian formal. Penyemakan jenis dilakukan menggunakan penyemak jenis yang turut disediakan oleh Z/EVES. Penyemakan jenis perlu dilaksanakan ke atas spesifikasi Z yang telah dibangunkan sebagai syarat sebelum pembuktian teorem boleh dilakukan menggunakan alatan tersebut. Dengan menggunakan Z/EVES, kesilapan jenis dalam spesifikasi dapat disemak. Jika terdapat kesilapan jenis, spesifikasi tersebut perlu diperbetulkan.

Setelah spesifikasi bebas daripada kesilapan jenis, pengenalpastian dan pembangunan



Rajah 2. Pendekatan proses penspesifikasian dan pengesahsahihan protokol CSMA/CD



Rajah 3. Proses dan langkah-langkah yang terlibat dalam membangunkan Z

teorem prakeadaan (Saaltink 1997) dilakukan. Teorem tersebut dibangunkan dalam spesifikasi Z yang sama dan perlu melalui proses penyemakan jenis menggunakan Z/EVES seperti yang telah dinyatakan di atas. Jika tiada kesilapan jenis, pembuktian teorem prakeadaan dilakukan. Jika teorem prakeadaan berjaya dibuktikan, ia menunjukkan spesifikasi Z tersebut dikatakan konsisten dan diaplikasikan dalam domain yang betul. Manakala jika teorem tersebut tidak berjaya dibuktikan, ia menunjukkan terdapat ketidakkonsistenan dalam spesifikasi Z atau teorem atau kedua-duanya. Maka, spesifikasi Z atau teorem atau kedua-duanya perlu dianalisis dan ketidakkonsistenan tersebut perlu diperbetulkan. Kemudian, proses penyemakan jenis serta pembuktian teorem diulang semula.

HASIL DAN PERBINCANGAN

Bahagian ini membincangkan tentang pembangunan spesifikasi Z CSMA/CD, pembangunan teorem yang perlu dibuktikan, penyemakan jenis dan pembuktian yang dilaksanakan menggunakan alatan Z/EVES.

Spesifikasi Z CSMA/CD

Spesifikasi Z CSMA/CD terdiri daripada jenis, pembolehubah sejagat, keadaan sistem, keadaan awal dan operasi-operasi yang diwakilkan dalam bentuk perenggan. Jenis dalam Z dapat dibahagikan kepada tiga jenis iaitu jenis asas, jenis bebas dan jenis majmuk.

Jenis Asas

Nod dalam rangkaian diwakilkan sebagai NODE_ID dan data yang perlu dihantar oleh setiap nod diwakilkan sebagai DATA. Kedua-dua elemen ini diwakilkan sebagai jenis asas dalam spesifikasi Z seperti berikut:

$[NODE\_ID, DATA]$

Jenis Bebas

Terdapat tiga isyarat yang dijana oleh nod dan disebarkan oleh medium iaitu:

- Isyarat penghantaran atau penerimaan sedang berlaku,
- Isyarat pelanggaran sedang berlaku atau
- Isyarat kosong.

Isyarat kosong bermaksud tiada penghantaran, penerimaan atau pelanggaran sedang berlaku di dalam medium. Ketiga-tiga isyarat tersebut diwakilkan sebagai jenis bebas bentuk pertama seperti berikut:

$signalValue ::= tranceive \mid collosion \mid clear$

Mesej yang dihantar mempunyai alamat nod yang menghantar dan alamat nod yang menerima. Nod yang menghantar mesej diwakili sebagai SA manakala yang menerima mesej diwakili sebagai DA. Kedua-dua nod tersebut diwakilkan sebagai jenis bebas menggunakan takrifan abreviasi seperti berikut:

$DA == NODE\_ID$
$SA == NODE\_ID$

Mesej yang dihantar oleh nod diwakilkan sebagai paket yang mempunyai maklumat tentang alamat nod yang menerima, alamat nod yang menghantar dan data seperti yang diwakilkan jenis bebas menggunakan takrifan abreviasi seperti berikut:

$PACKET == DA \times SA \times DATA$

Pembolehubah sejagat

Keadaan pelanggaran di dalam medium diwakili oleh ungkapan undefined dan keadaan tiada paket di dalam medium diwakili oleh ungkapan emptyPaket. Kedua-dua ungkapan tersebut diwakilkan dalam spesifikasi sebagai pembolehubah sejagat seperti berikut. Perwakilan di bawah menunjukkan kedua-dua pemboleh ubah tersebut adalah pemboleh ubah yang berlainan tetapi mempunyai jenis yang sama.

$undefined: PACKET$
$emptyPacket: PACKET$
$undefined \neq emptyPacket$

Keadaan bagi CSMA/CD

Keadaan bagi protokol CSMA/CD diwakili oleh dua skema iaitu skema *Node* dan skema *Controller*. Skema *Node* mempunyai pemboleh ubah keadaan yang berikut:

- *Id* mewakili nod dalam rangkaian.
- *msg* mewakili penimbal dalam nod yang menyimpan mesej untuk dihantar.
- *inbuf* mewakili penimbal dalam nod yang menyimpan mesej yang diterima.

Semua pemboleh ubah tersebut dinyatakan dalam skema *Node* seperti berikut:

Node

$Id: NODE\_ID$
$msg: PACKET$
$inbuf: PACKET$

Manakala, skema *Controller* mempunyai pemboleh ubah keadaan yang berikut:

- *medium* mewakili medium iaitu bus.
- *signal* mewakili isyarat yang dijana oleh nod dan disebarkan oleh medium.

Semua pemboleh ubah tersebut dinyatakan dalam skema *Controller* seperti berikut:

Controller

$medium:PACKET$
$signal: NODE\_ID \rightarrow signalValue$

Keadaan Awal

Secara praktikalnya, keadaan awal bagi CSMA/CD ialah penimbal nod iaitu *msg* dan *inbuf* adalah kosong. Penimbal kosong bermaksud tiada data dalam penimbal tersebut. Namun, dari aspek teknikal, keadaan ini, iaitu spesifikasi yang mewakili skema *Node* boleh diabaikan (Spivey 1998). Ini kerana paket ialah pemboleh ubah berjenis *tuple* yang diwakili oleh ungkapan *paket* == DA x SA x DATA. Keadaan awal bagi *tuple* ialah tiada komponen di dalamnya. Dalam penulisan spesifikasi Z tiada cara untuk menyatakan jika pemboleh ubah yang berjenis *tuple* nilainya bersamaan dengan tiada komponen. Maka, pengawalan bagi skema *Node* diabaikan.

Operasi-Operasi

Berdasarkan kepada model SCM bagi CSMA/CD (Lundy & Miller 1993), lima skema operasi bagi *Node* dan empat skema operasi bagi Controller perlu dibangunkan dalam spesifikasi Z CSMA/CD. Sebahagian operasi mungkin akan mengubah keadaan sistem. Skema yang mewakili operasi yang menyebabkan perubahan kepada sistem akan menggunakan notasi Δ. Manakala, skema yang mewakili operasi yang tidak menyebabkan perubahan kepada sistem akan menggunakan notasi Ξ. Pemboleh ubah dengan simbol ' (perdana) menunjukkan ia adalah pemboleh ubah keadaan selepas sesuatu operasi dilaksanakan.

Operasi: *Transmit*

Operasi *Transmit* ialah operasi yang menunjukkan nod sedang menghantar mesej. Syarat bagi operasi ini berlaku ialah: "penimbal nod mestilah mempunyai mesej untuk dihantar", dan "medium yang digunakan untuk menghantar data kepada nod penerima tersebut mestilah kosong". Kedua-dua syarat ini diwakili oleh pernyataan *msg* = *emptyPacket* dan *medium* = *emptyPacket*. Operasi ini tidak akan menyebabkan perubahan kepada *Node*, tetapi akan menyebabkan perubahan kepada *Controller*. Jika syarat tersebut dipatuhi, mesej pada penimbal nod akan diberikan kepada medium. Perubahan tersebut diwakili oleh pernyataan *medium'* = *msg*. Skema Transmit dibangunkan seperti yang ditunjukkan berikut:

Transmit

$\Xi Node$
$\Delta Controller$

$msg \neq emptyPacket$
$medium = emptyPacket$
$signal' = signal \oplus \{(Id \text{ a } transceive)\}$
$medium' = msg$

Operasi: OK

Operasi OK ialah operasi yang menunjukkan nod selesai menghantar mesej. Syarat bagi operasi ini berlaku ialah: "nod telah menjana isyarat kosong bagi medium" dan "penimbal nod mempunyai mesej di dalamnya". Syarat ini diwakili oleh pernyataan *signal(Id)* = *clear* dan *msg* ≠ *emptyPacket*. Ini bermakna nod tersebut telah selesai melakukan operasi penghantaran mesej dan penimbal nod perlu dikosongkan. Operasi ini akan menyebabkan perubahan kepada *Node*, tetapi tidak akan menyebabkan perubahan kepada *Controller*. Jika memenuhi syarat, penimbal nod akan dikosongkan. Perubahan tersebut diwakili oleh pernyataan *msg'* = *emptyPacket*. Skema operasi *OK* dibangunkan seperti yang ditunjukkan berikut:

OK

$\Xi Node$
$\Delta Controller$

$signalID - clear$
$msg \neq emptyPacket$
$msg' = emptyPacket$
$Inbuf' = Inbuf$
$Id' = Id$

Operasi: *coll_D*

Operasi *coll_D* ialah operasi yang menunjukkan nod mengesan pelanggaran. Syarat bagi operasi ini berlaku ialah: "nod mengesan pelanggaran

berlaku dalam medium". Syarat ini diwakili oleh pernyataan *medium = undefined*. Operasi ini tidak akan menyebabkan perubahan kepada *Node*, tetapi akan menyebabkan perubahan kepada *Controller*. Jika syarat tersebut dipatuhi, nod tersebut akan menjana isyarat untuk menyatakan pelanggaran sedang berlaku. Perubahan tersebut diwakili oleh pernyataan *signal' = signal ± (Id Hϕ collision)*. Skema operasi *coll_D* dibangunkan seperti yang ditunjukkan berikut:

coll_D

$\Xi Node$
$\Delta Controller$

$medium = undefined$
$signal' = signal \oplus \{(Id \text{ a } collision)\}$
$medium' = medium$

Operasi: *ready*

Operasi *ready* ialah operasi yang menunjukkan nod bersedia untuk menghantar mesej. Syarat bagi operasi ini berlaku ialah: "nod mengesan medium adalah kosong". Syarat ini diwakili oleh pernyataan *signal(Id) = clear*. Operasi ini tidak akan menyebabkan sebarang perubahan kepada *Node* ataupun *Controller*. Skema operasi *ready* dibangunkan seperti yang ditunjukkan berikut:

ready

$\Xi Node$
$\Xi Controller$

$signal\ Id = clear$

Operasi: *receive*

Operasi *receive* ialah operasi yang menunjukkan nod sedang menerima mesej. Syarat bagi operasi ini berlaku ialah: "nod mengesan mesej telah dihantar kepadanya". Syarat ini diwakili oleh pernyataan *medium.1 = Id*. Operasi ini akan menyebabkan perubahan kepada *Node* dan *Controller*. Jika syarat tersebut dipatuhi, mesej tersebut akan dimasukkan ke dalam penimbal nod dan nod akan menjana isyarat yang menyatakan penghantaran sedang berlaku. Kedua-dua perubahan tersebut diwakili oleh pernyataan *inbuf' = medium* dan Id Hϕ transceive. Skema operasi receive dibangunkan seperti yang ditunjukkan berikut:

receive

$\Delta Node$
$\Delta Controller$

$medium.1 = Id$
$inbuf' = medium$
$signal' = signal \oplus \{(Id \text{ a } transceive)\}$
$medium' = medium$
$msg' = msg$
$Id' = Id$

Operasi: *message*

Operasi *message* ialah operasi yang menunjukkan penghantaran mesej sedang berlaku dalam medium. Syarat bagi operasi ini berlaku ialah: "medium mestilah tidak kosong" dan "tiada pelanggaran berlaku di dalamnya". Kedua-dua syarat ini diwakili oleh pernyataan $medium \neq emptyPacket$ dan $medium \neq undefined$. Operasi ini tidak akan menyebabkan sebarang perubahan kepada *Node* ataupun *Controller*. Skema operasi message dibangunkan seperti yang ditunjukkan berikut:

message

$\Xi Controller$

$medium \neq undefined$
$medium \neq emptyPacket$

Operasi: *resetM*

Operasi *resetM* ialah operasi yang menunjukkan penghantaran mesej dalam medium telah selesai. Syarat bagi operasi ini berlaku ialah: "medium mengesan isyarat penghantaran sedang berlaku" dan "terdapat paket di dalam medium". Syarat ini diwakili oleh pernyataan $\exists\ i: NODE\_ID \mid i = medium.1 \bullet signal\ i = transceive$ dan $medium \neq emptyPacket$. Operasi ini tidak melibatkan *Node*, tetapi akan menyebabkan perubahan kepada *Controller*. Jika syarat tersebut dipatuhi, iaitu penghantaran selesai dilakukan, medium akan dikosongkan dan isyarat medium telah kosong akan disebarkan. Perubahan tersebut diwakili oleh pernyataan $medium' = emptyPacket$ dan $\forall\ i: NODE\_ID \mid i = medium.1 \bullet signal'\ i = clear$. Skema operasi *resetM* dibangunkan seperti yang ditunjukkan berikut:

resetM

$\Delta Controller$

$\exists i: NODE\_ID \mid i = medium.1 \bullet signal\ i = transceive$
$medium \neq emptyPacket$
$medium' = emptyPacket$
$\forall i: NODE\_ID \mid i = medium.1 \bullet signal'\ i = clear$

Operasi: *collisionD*

Operasi *collisionD* ialah operasi yang menunjukkan pelanggaran sedang berlaku dalam medium. Syarat bagi operasi ini berlaku ialah: "medium mengesan pelanggaran berlaku di dalamnya". Syarat ini diwakili oleh pernyataan *medium = undefined*. Operasi ini tidak melibatkan *Node* dan tidak akan menyebabkan perubahan kepada *Controller*. Skema operasi *collisionD* dibangunkan seperti yang ditunjukkan berikut:

collisionD

$\Xi Controller$

$medium = undefined$

Operasi: *resetC*

Operasi *resetC* ialah operasi yang menunjukkan pelanggaran dalam medium telah selesai. Syarat bagi operasi ini berlaku ialah: "terdapat satu nod menjana isyarat pelanggaran sedang berlaku" dan "terdapat paket di dalam medium". Syarat ini diwakili oleh pernyataan $\exists i: NODE\_ID \bullet signal\ i = collision$ dan $medium \neq emptyPacket$. Operasi ini tidak melibatkan *Node*, tetapi akan menyebabkan perubahan kepada *Controller*. Jika syarat tersebut dipatuhi, medium akan dikosongkan dan isyarat medium telah kosong akan disebarkan. Perubahan tersebut diwakili oleh pernyataan $medium' = emptyPacket$ dan $\forall i: NODE\_ID \bullet signal'\ i = clear$. Skema operasi tersebut dibangunkan seperti yang ditunjukkan berikut:

resetC

$\Delta Controller$

$\exists i: NODE\_ID \mid \bullet signal\ i = collision$
$medium \neq emptyPacket$
$medium' = emptyPacket$
$\forall i: NODE\_ID \mid i = signal'\ i = clear$

Penyemakan jenis menggunakan alatan Z/EVES

Seperti yang ditunjukkan dalam Rajah 2, setelah spesifikasi Z CSMA/CD siap dibangunkan, penyemakan jenis dilakukan ke atas spesifikasi tersebut menggunakan Z/EVES. Ia adalah untuk memastikan spesifikasi bebas daripada kesilapan jenis dan sebagai syarat sebelum pembuktian teorem boleh dilakukan menggunakan alatan tersebut. Penyemakan jenis yang dilakukan menggunakan Z/EVES dapat mengesan kesilapan jenis dalam spesifikasi yang terlepas pandang oleh penspesifikasi semasa membangunkan spesifikasi tersebut.

Pembangunan teorem prakeadaan

Pembuktian prakeadaan bertujuan untuk menunjukkan setiap operasi tidak diaplikasikan di luar domain iaitu situasi yang mana hasil operasi tidak dapat dikenalpasti (Woodcock & Davies 1996). Pembuktian teorem prakeadaan dimulakan dengan membangunkan teorem prakeadaan. Teorem-teorem prakeadaan dibangunkan berdasarkan kepada skema operasi dalam bentuk seperti yang telah ditunjukkan dalam Saaltink (1997).

theorem NamaTeorem
$\tilde{O}S\ ; in? : IN \mid P \times pre\ Op$

yang mana

NamaTeorem adalah nama bagi teorem tersebut dan boleh sebarangan. Namun, adalah lebih bermakna jika nama tersebut mengambil nama skema operasi yang berkaitan, ditambah dengan perkataan Pre di hujungnya.
S ialah skema-skema lain yang terlibat dalam skema operasi Op.
in? ialah pemboleh ubah input dalam bahagian pengisytiharan skema operasi Op.
IN ialah jenis bagi pemboleh ubah input dalam bahagian pengisytiharan skema operasi Op.
P ialah prakeadaan yang sepatutnya bagi operasi Op.
Op ialah skema operasi yang ingin dibuktikan sama ada ia diaplikasikan dalam domain yang betul atau tidak.

Setelah teorem siap dibangunkan, penyemakan jenis dilakukan ke atas teorem tersebut menggunakan fungsi yang disediakan oleh Z/Eves. Ia adalah untuk memastikan teorem

bebas daripada kesilapan jenis dan sebagai syarat sebelum pembuktian teorem boleh dilakukan menggunakan alatan tersebut. Jika terdapat kesilapan jenis, Z/Eves akan menyenaraikan kesilapan tersebut. Jadual 1 menunjukkan pembangunan teorem prakeadaan berdasarkan kepada setiap operasi yang telah disemak.

Pembuktian menggunakan pembukti Z/EVES

Setelah teorem dibangunkan, penyemakan jenis dilakukan ke atas semua teorem tersebut dan ianya bebas daripada kesilapan jenis. Seterusnya, Z/EVES digunakan untuk

Jadual 1. Pembangunan teorem prakeadaan

Operasi disemak	Prasyarat bagi operasi	Ungkapan matematik	Teorem prakeadaan
Transmit	penimbal nod mestilah mempunyai mesej untuk dihantar, medium mestilah kosong	*msg = emptyPacket* *medium = emptyPacket*	**theorem** *TransmitPre* ∀*Node; Controller* | *msg ≠ emptyPacket* ∧ *medium = emptyPacket • pre Transmit*
OK	nod telah menjana isyarat bagi medium, penimbal nod mempunyai mesej di dalamnya	*signal (Id) = clear* *msg ≠ emptyPacket*	**theorem** *OKPre* ∀*Node; Controller* | *signal Id = clear* ∧ *msg ≠ emptyPacket • pre OK*
coll_D	nod mengesan perlanggaran berlaku dalam medium	*medium = undefined*	**theorem** *coll_D Pre* ∀*Node; Controller* | *medium = undefined • pre coll_D*
ready	nod mengesan medium adalah kosong	*signal (Id) = clear*	**theorem** *ready Pre* ∀*Node; Controller* | *signal Id = clear • pre ready*
receive	nod mengesan mesej telah dihantar kepadanya	*medium.*1 = *Id*	**theorem** *receice Pre* ∀*Node; Controller* | *medium.*1 = *Id • pre receive*
message	medium mestilah tidak kosong, tiada perlanggaran berlaku di dalamnya	*medium ≠ emptyPacket* *medium ≠ undefined*	**theorem** *message Pre* ∀ *Controller* | *medium ≠ undefined* ∧ *medium ≠ emptyPacket • pre message*
resetM	medium mengesan isyarat yang mengatakan penghantaran sedang berlaku, terdapat paket di dalam medium	∃ *i: NODE_ID* | *i = medium.1 • signal i = tranceive* *medium ≠ emptyPacket*	**theorem** *resetMPre* ∀ *Controller* | *(∃ i: NODE_ID* | *i = medium.1 • signal i = tranceive)* ∧ *medium ≠ emptyPacket • pre resetM*
collisionD	medium mengesan perlanggaran berlaku di dalamnya	*medium = undefined*	**theorem** *collisionDPre* ∀ *Controller* | *medium ≠ undefined • pre collisionD*
resetC	terdapat satu nod menjana isyarat perlanggaran sedang berlaku terdapat paket di dalam medium	∃ *i: NODE_ID •* *signal i = collision* *medium ≠ emptyPacket*	**theorem** *resetCPre* ∀ *Controller* | *(∃ i: NODE_ID • signal Id = collision* ∧ *medium ≠ emptyPacket • pre resetC*

melaksanakan pembuktian ke atas setiap teorem bagi membuktikan skema operasi yang telah dibangunkan dalam spesifikasi adalah konsisten dan diaplikasikan dalam domain yang betul. Terdapat dua versi Z/EVES yang boleh digunakan iaitu versi 2.1 atau 2.3. Z/EVES versi 2.1 mempunyai antara muka menggunakan baris arahan. Manakala Z/EVES versi 2.3 mempunyai antara muka bergrafik yang dikenali sebagai Z/EVES GUI.

Berikut adalah contoh langkah pembuktian bagi teorem prakeadaan *TransmitPre* dalam Z/EVES 2.1. Teorem untuk dibuktikan perlu ditentukan menggunakan arahan *try*. Setelah teorem ditentukan, arahan pembuktian boleh ditaip untuk membuktikan teorem tersebut. Arahan *prove by reduce* dalam Z/EVES digunakan untuk membuktikan teorem *TransmitPre*. Arahan tersebut menghasilkan keputusan *true* yang membuktikan matlamat pembuktian yang diwakilkan dalam teorem adalah benar.

Bukti
try lemma TransmitPre;
prove by reduce;

Langkah pembuktian dalam Z/EVES 2.3 adalah sama, cuma Z/EVES 2.3 lebih mesra pengguna berbanding Z/EVES 2.1. Teorem *TransmitPre, OKPre, coll_DPre, readyPre, receivePre, messagePre* dan *collisionDPre* telah dapat dibuktikan benar menggunakan arahan prove by reduce dalam Z/EVES berdasarkan kepada langkah pembuktian yang telah ditunjukkan di atas. Keputusan benar dalam hasil pembuktian di atas menunjukan operasi Transmit, OK, coll_D, ready, receive, message dan collisionD yang telah dispesifikasi diaplikasikan dalam domain yang betul.

Manakala bagi teorem *resetM* dan *resetC*, pembuktian pada mulanya dilakukan dalam alatan Z/EVES menggunakan arahan *prove by reduce*. Namun, kedua-dua teorem tersebut menghasilkan hasil pembuktian yang tidak dapat dikembangkan lagi oleh Z/EVES. Ini bermakna interaksi pengguna diperlukan untuk meneruskan pembuktian bagi kedua-dua teorem tersebut. Namun, menurut Ledru (1998) dan Simpson (2004), pengalaman dan lengkungan pembelajaran yang sedikit dalam alatan Z/EVES dan teknik pembuktian teorem akan membataskan demonstrasi pembuktian teorem menggunakan alatan tersebut. Sehubungan itu, pembuktian bagi teorem *resetM* dan *resetC*, kemudiannya diteruskan secara manual dengan menganalisis hasil pembuktian selepas arahan *prove by reduce* yang terakhir dilaksanakan. Berikut adalah contoh pembuktian yang diteruskan secara manual bagi teorem prakeadaan *resetM*.

Rajah 4 menunjukkan hasil pembuktian selepas arahan *prove by reduce* yang terakhir dilaksanakan. Penomboran di sebelah kiri hasil pembuktian hanya untuk rujukan dalam analisis hasil tersebut.

Merujuk hasil dalam rajah 4, ungkapan pada baris 2 dan 5 adalah benar. Maka, $\Rightarrow \exists$ signal':P (NODE_ID x signalValue) • signal' $\in$ NODE_ID ϕ signalValue $\wedge$ signal' medium.1 = clear, juga benar. Oleh itu teorem prakeadaan resetMPre adalah benar. Keputusan benar dalam hasil pembuktian di atas menunjukan operasi resetM yang telah dispesifikasi diaplikasikan dalam domain yang betul.

KESIMPULAN

Kajian ini telah berjaya mendemonstrasikan penspesifikasian dan pengesahsahihan secara formal protokol CSMA/CD. Penspesifikasian bagi protokol CSMA/CD telah dibangunkan menggunakan bahasa formal Z. Kemudian,

```
1 medium ∈ NODE_ID × NODE_ID × DATA
2 ∧ signal × NODE_ID → signalValue
3 ∧ i = medium.1
4 ∧ medium.1 ∈ NODE_ID
5 ∧ signal medium.1 = transceive
6 ∧ emptyPacket ∈ Node_ID × NODE_ID × DATA
7 ∧ ¬ medium = emptyPacket
8 ⇒ (∃ signal': P (NODE_ID × signalValue)
9     • signal' ∈ NODE_ID  signalValue ∧ signal ' medium.1
```

Rajah 4. Hasil dalam pembuktian selepas arahan prove by reduce yang terakhir dilaksanakan ke atas teorem prakeadaan resetMPre

pengesahsahihan formal dilakukan ke atas spesifikasi Z CSMA/CD tersebut menggunakan teknik pembuktian teorem. Ia adalah untuk memastikan spesifikasi Z yang telah dibangunkan konsisten dan diaplikasikan dalam domain yang betul. Sebelum pembuktian dilaksanakan, penyemakan jenis telah dilakukan ke atas spesifikasi Z yang telah dibangunkan untuk memastikan spesifikasi bebas daripada kesilapan jenis yang terlepas pandang oleh penspesifikasi. Kemudian pembuktian dilakukan ke atas teorem yang telah dibangunkan menggunakan arahan prove by reduce dalam Z/EVES.

Pembuktian teorem adalah proses yang panjang, berulang dan adalah tugasan yang ralat cenderung berlaku jika dilakukan secara manual. Namun dengan menggunakan alatan Z/EVES sebagai alatan sokongan dalam proses pembuktian dapat menjimatkan masa dan tenaga kerana proses pembuktian yang berulang-ulang dan panjang dilakukan oleh alatan tersebut. Pengaplikasian teorem dan hukum diaplikasikan berulang kali secara automatik oleh Z/EVES dengan menggunakan arahan tertentu. Penumpuan dapat diberikan kepada penganalisaan hasil output yang dihasilkan oleh Z/EVES tentang spesifikasi yang telah dibangunkan tanpa perlu penumpuan kepada langkah pembuktian yang terperinci seperti dalam pembuktian manual. Ini kerana kesilapan dalam hasil output pembuktian berkemungkinan menunjukkan kesilapan dalam spesifikasi tersebut. Dari segi kekemasan pula, langkah-langkah pembuktian juga dipaparkan dalam Z/EVES mengikut urutan untuk memudahkan penyemakan. Selain daripada itu, kecenderungan ralat berlaku adalah kecil.

PENGHARGAAN

Penulis merakamkan setinggi-tinggi penghargaan kepada Universiti Kebangsaan Malaysia dan Kerajaan Malaysia atas sokongan kewangan bagi kajian ini melalui geran IRPA 04-02-02 0055 EA222.

RUJUKAN

Ainita Ban dan Zarina Shukur. 2009. Formal Methods in analysing Electronic Voting Protocol. Laporan Teknikal FTSM/Mei 2009/LT01. Universiti Kebangsaan Malaysia.

Barden, R., Stepney, S. & Cooper, D. 1994. *Z In Practice.* London: Prentice Hall International (UK) Limited.

Bochmann, G.V. & Sunshine, C.A. 1983. *A Survey of Formal Methods in Computer Network And Architectures and Protocols.* IBM Corporation Yorktown Heights, New York: Plenum Press.

Bowen, J.P. 15 Jun 2005.comp.specification.z Frequently Asked Questions (FAQ). http://www.faqs.org/faqs/z-faq/ [20 Disember 2004]

Clarke, E.M. & Wing J.M. 1996. Formal Methods: State of the Art and Future Directions. *ACM Computing Surveys* 28 (4): 626 – 643.

Giunchiglia, F. & Traverso, P. 2000. Theorem proving in technology transfer: the user's point of view. *International Journal on Software Tools for Technology Transfer* 3(1): 1–12.

Haggarty, R. 2002. *Discrete Mathematics For Computing.* Dorset: Pearson Education Limited.

IEEE/ANSI. 1985. Local Area Networks, Carrier Sense Multiple Access / Collision Detection. ANSI/IEEE Std 802.3. New York. IEEE, Inc.

ISO/IEC. 2002. *Information technology — Z formal specification notation — Syntax, type system and semantics.* ISO/IEC 13568:2002 (Z standard). Switzerland

ITSEC (June 1991). *Information Technology Security Evaluation Criteria (ITSEC): Preliminary Harmonised Criteria.* Document COM(90) 314, Version 1.2. Commission of the European Communities. (atas talian) http://www.ssi.gouv.fr/ site_documents/ITSEC/ITSEC-uk.pdf. [30 April 2009]

Ledru, Y. 1998. Identifying Pre-Conditions with the Z/EVES Theorem Prover. *Proceedings of the 13th IEEE International Conference on Automated Software Engineering*: 32.

Lundy, G.M. & Miller, R. 1993. Analyzing a CSMA/CD Protocol Through a System of Communicating Machines Specification. *Transaction On Communication.* 41 (3).

NASA ARC. 25 Januari 2002. V&V OF ADVANCED SYSTEMS AT NASA. TASK. NO 10 TA-5.3.3 (WBS 1.4.4.5.3). Northrop Grumman Corp.

Quemad, J. 2004. Formal Description Techniques and Software Engineering: Some Reflections after 2 Decades of Research. *Proceedings of FORTE 2004*. LNCS 3235: 33-42.

Saaltink, M. 1997. *The Z/EVES User Guide*. Ottawa. ORA Canada.

Simpson, C. 2004. Computer Theorem Proving in Mathematics. *Letters in Mathematical Physics* 62: 287–315.

Sommerville, I. 1995. Chapter 11 Model-based Specification. http://www.comp.lancs.ac.uk/computing/resources/IanS/SE7/ElectronicSupplements/Model Spec.pdf [2 Mei 2004]

Spivey, J. M. 1998. The Z Notation: A Reference Manual Second Edition. Oxford. Oriel College, Oxford.

Sremack, J. (tanpa tarikh). Automated Reasoning and Theorem Provers, Chapter 5. ages.wooster.edu/jsremack/is/theorem.pdf. [17 Januari 2005]

WetStone Technologies, Inc. 1999. Formal Methods Framework. http://www.cs.utexas.edu/users/csed/FM/work/collected-v3.pdf. [31 Januari 2004]

Wing, J.M. 1990. A Specifier's Introduction to Formal Methods. *Computer* 23(9):8-24.

Wing, J.M. 1998. A Symbiotic Relationship Between Formal Methods and Security. Proceeding of the Conference on Computer Security, Dependablity and Assurance: *From Needs to Solution:* 447-449.

Woodcock, J. & Davies, J. 1996. Using Z: *Specification, Refinement and Proof.* London. Prentice Hall.

Zarina Shukur dan Nursyahidah Alias. 2007a. Tinjauan Penggunaan Kaedah Formal Ke Atas Protokol Authentication. *Persidangan Kebangsaan Sains Pengaturcaraan* 2007: 204-215

Zarina Shukur dan Nursyahidah Alias. 2007b. Kaedah Formal Dalam Protokol Security Electronic Transaction. Laporan Teknikal FTSM/Disember 2007/LT06. Universiti Kebangsaan Malaysia.

Zarina Shukur, Nursyahidah Alias, Mohd. Hazali Mohd. Halip and Bahari Idrus, 2007c, Formal Validation of The Safety Property of SACK Protocol Using Theorem Proving Technique, *Journal of Computer Science,* 3(6): 449-453.