

# ANALISIS TEKNIK PENGALAMATAN PADA WIRELESS SENSOR NETWORKS

*(Analyzing Addressing Techniques on Wireless Sensor Networks)*

Eddy Wijanto

Fakultas Teknik dan Ilmu Komputer Jurusan Teknik Elektro  
Universitas Kristen Krida Wacana – Jakarta  
eddy.wijanto@ukrida.ac.id

## Abstrak

*Wireless Sensor Networks* terdiri atas sejumlah besar titik (*nodes*) dengan kemampuan untuk melakukan *sensing*, komputasi, dan komunikasi. Dalam banyak aspek, *sensor networks* memiliki kesamaan dengan *mobile ad hoc networks* (MANET), namun memiliki beberapa perbedaan yang penting. Kemiripannya adalah dalam hal topologi, medium komunikasi yang digunakan bersama, dan permasalahan dalam konektivitas. Perbedaannya adalah *sensor networks* biasanya mencakup mobilitas yang lebih rendah dan sumber daya yang jauh lebih terbatas, oleh karenanya menimbulkan permasalahan skalabilitas. Dalam *paper* ini, dibahas teknik-teknik pengalamatan untuk *sensor networks*, *trade-off* dalam desain skema pengalamatan tertentu, dan perbandingan di antara berbagai teknik.

**Kata Kunci:** *wireless sensor networks*, *nodes*, pengalamatan, *wireless access media*, RETRI

## Abstract

*Wireless Sensor Networks* consist of *nodes* which are able to perform *sensing*, *computation*, and *communication*. In many aspects, *sensor networks* are similar but also different from the *mobile ad hoc networks* (MANET). Their similarities include *topology*, *shared communication medium*, and *connectivity problems*. Different from the MANET, the *sensor networks* typically contain lower *mobility*, much more limited resources, which therefore lead to *scalability problems*. This paper discusses the *addressing techniques* for *sensor networks*, the *trade-offs* in the design of certain *addressing scheme*, and the comparison among various techniques.

**Keywords:** *wireless sensor networks*, *nodes*, *addressing*, *wireless access media*, RETRI

## 1. PENDAHULUAN

*Wireless Sensor Networks* (WSN) adalah sekumpulan peralatan *sensing* dengan sumber daya baterai dan biaya yang relatif rendah, biasanya dikenal sebagai sensor. Sensor umumnya memiliki sumber daya baterai dan ketika disebar akan tetap berada di daerah tertentu untuk jangka waktu yang lama. Baterai akan sulit untuk diganti ataupun diisi kembali sehingga sensor mempunyai sifat keterbatasan energi. Untuk membatasi penggunaan energi, sensor tidak melakukan pengiriman ke jarak yang jauh untuk mencapai *sink*, melainkan melakukan komunikasi *multihop*, mengirimkan data melalui tetangganya [1].

Ukuran dari *sensor networks* mengalami pertumbuhan yang pesat. *ON World* memprediksikan bahwa pada tahun-tahun mendatang, lebih dari 465 juta modul *radio frequency* (RF) untuk *sensor* akan diproduksi, dibandingkan dengan pada tahun 2003 yang hanya berjumlah 3 juta modul. Tiga puluh lima persen dari modul ini akan digunakan untuk aplikasi industri dan 28 persen untuk kontrol dan otomatisasi di rumah-rumah.

Meskipun skalabilitas dari sebuah *networks* yang besar merupakan persoalan yang telah diketahui, namun ukuran *nodes* yang semakin diperkecil sering kali kurang diperhatikan. Interaksi manusia dengan benda kecil ini tidak mungkin dilakukan untuk setiap *node*, misalkan saja untuk melakukan konfigurasi ataupun penggantian baterai. Bila hal ini dilakukan secara satu per satu untuk setiap titik tentunya akan membuat biaya menjadi besar. Ukurannya yang kecil juga menandakan sumber dayanya yang terbatas, yaitu memori, CPU, dan baterai [1].

Walaupun mobilitas fisik tidak menjadi isu sentral pada sebagian besar *sensor*, namun *sensor* memiliki baterai yang kecil, hal ini berarti bahwa *sensor* harus beroperasi dengan pengurangan *duty cycle*. Dengan demikian, *sensor* hanya beroperasi untuk sebagian waktu, tetapi harus tetap menjaga konektivitas. Sifat *on-off* dari *nodes* merupakan faktor signifikan yang harus diperhitungkan untuk aspek komunikasi dan sinkronisasi dari *sensor networks*.

Aspek komunikasi merupakan isu sentral dari *sensor networks* karena energi yang digunakan jauh lebih besar dibandingkan dengan komputasi. Untuk komunikasi, diperlukan sebesar 10 mikro J untuk mengirimkan 100 bit dengan jarak 100 m, namun untuk mengeksekusi instruksi 32 bit hanya diperlukan 0,06 nJ (lebih rendah 100.000 kali lipat). Walaupun perbedaan ini tidak dapat dihilangkan, karena tugas utama dari *sensor node* di samping melakukan *sensing* juga harus *me-relay* informasi dari node lainnya, sehingga perlu dicoba untuk melakukan *trade-off* komunikasi untuk komputasi. Untuk melakukan hal ini, dapat digunakan kompresi dan agregasi (penggabungan) data karena *sensor networks* memerlukan aplikasi yang spesifik (*application-specific*).

Kompleksitas pengiriman data pada *sensor network* mulai dari komunikasi sederhana dari sumber ke tujuan hingga komunikasi dari banyak sumber ke banyak tujuan dengan tugas-tugas pengumpulan (agregasi), *data rate*, dan jaminan kualitas. Penyebaran dan pengumpulan data (*routing*, *querying*, dan *discovery*) perlu untuk melakukan adaptasi terhadap perbedaan kompleksitas tersebut [2].

*Routing* merupakan aspek dasar dari pengiriman data. *Traffic* pada *sensor networks* berbeda dengan *internet traffic*. Data yang dihasilkan *sensor* sensitif terhadap fenomena yang di-*sense* dan dapat dikompres, digabungkan, di-*delay*, atau dikirimkan dengan cepat, tergantung pada urgensinya. Karena perbedaan ini, protokol *routing* untuk *sensor network* perlu untuk didesain secara spesifik. *Querying* biasanya mirip dengan yang digunakan pada sistem *database*. *Stream* dari data yang dihasilkan oleh *sensor* dapat dilihat sebagai tabel. Namun tetap saja harus disesuaikan dengan kondisi dari *sensor network* [2].

## 2. PARADIGMA KOMUNIKASI PADA SENSOR NETWORKS

Dalam aspek penyebaran dan penggabungan (*routing*, *querying*, dan *discovery*), terdapat tiga paradigma komunikasi, yaitu terpusat pada *node* (*node-centric*), terpusat pada data (*data-centric*), dan terpusat pada posisi (*position-centric*).

### 2.1 Node-Centric

Cara pandang tradisional terhadap suatu jaringan adalah melalui pemberian label pada *nodes* dengan nama tertentu dan *routing* diimplementasikan berdasarkan nama tersebut. Internet yang ada saat ini menggunakan pendekatan *node-centric* ini, karena

pendekatan ini sangat intuitif. Pendekatan *node-centric* dapat diaplikasikan dengan mudah untuk pengalamatan bertingkat (*hierarchical*), tetapi bukan merupakan keuntungan pada *sensor networks* yang menerapkan *flat logical*.

Pada *sensor networks*, komunikasi *end point* seharusnya tidak diidentifikasi dengan menggunakan nama *node* terutama dikarenakan sifat *on-off* dari jaringan (pengurangan *duty cycle*). Algoritma proaktif sepanjang waktu memelihara rute ke seluruh tujuan. Algoritma reaktif melakukan pencarian rute hanya ketika diperlukan atau menjadi *invalid*. Kategori ketiga merupakan gabungan dari komponen proaktif dan reaktif. Pendekatan proaktif lebih baik digunakan pada jaringan *fixed* atau jaringan dengan mobilitas yang rendah, sedangkan pendekatan reaktif bekerja dengan baik untuk jalur-jalur yang sering berubah [3].

Pada *sensor networks*, mobilitas memiliki pengertian yang berbeda, yaitu *duty cycle* yang rendah dan periode tidur yang acak untuk setiap *nodes*. Hal ini mengakibatkan jalur akan sering mengalami invalidasi walaupun posisi dari *nodes* tidak pernah berubah. *Discovery* digunakan oleh algoritma *routing* reaktif untuk mencari jalur ke tujuan. Hal ini biasanya dicapai dengan *network-wide broadcasting*, yang diimplementasikan sebagai *flooding* terkontrol. Pada jaringan RF yang padat, hal ini membawa kepada fenomena yang dikenal sebagai *broadcast storm* dimana setiap *nodes* saling berbagi medium komunikasi untuk mengirimkan data yang sama. Efek ini memerlukan energi yang besar untuk *nodes* yang memiliki energi yang terbatas. Untuk mengurangi hal ini, para peneliti telah mencoba beberapa metode, diantaranya: *Fish Eye State*, *location servers*, dan sifat *small world*.

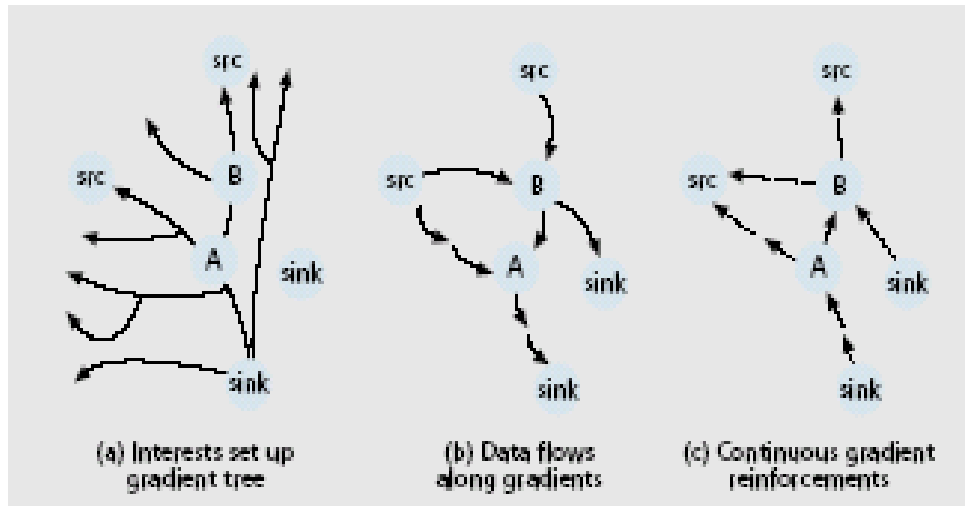
Metode *fish eye state* menggunakan prinsip bahwa hanya informasi yang berada dekat dengan *node* yang memiliki informasi yang detil dan aktual, sedangkan informasi yang berada jauh dari *node* memiliki resolusi yang lebih rendah. Sifat “*small world*” digunakan untuk mengurangi panjang dari jalur *discovery* (dua orang saling mengetahui satu dengan lainnya rata-rata dalam enam hubungan). Metode lainnya adalah dengan menggunakan *server* lokasi, misalnya dengan menggunakan skema *quad tree*.

Pengalamatan secara *node-centric* menangani masalah *multicast* dengan baik, namun untuk *sensor networks*, *multicast* dengan keanggotaan dan cabang-cabang akan menjadi sangat sulit untuk ditangani, khususnya karena usianya yang singkat. Persoalan dalam arah sebaliknya, dimana komunikasi dari banyak titik ke satu tujuan akan lebih banyak ditangani oleh paradigma yang selanjutnya.

## 2.2 Data-Centric

Pendekatan *data-centric* ditemukan untuk menjawab *query* “berikan saya data yang memenuhi kondisi tertentu (misal *range* suhu tertentu)”. Identitas dari *node* yang menghasilkan data yang sesuai dengan kondisi yang ditentukan tidaklah diketahui, sehingga *query* harus menggunakan *network-wide discoveries* untuk mencari identitas ini. Jika beberapa *sink* tertarik dengan data serupa, jalur-jalur yang terpisah harus ditemukan dan ditangani secara terpisah, sementara penggabungan data hanya dapat dilakukan di tempat tujuan [3].

Aspek yang paling penting dari paradigma *data-centric* adalah isi dari data yang dihasilkan oleh *sensor*. Paket dirutekan berdasarkan pada isinya, sementara identitas dari *nodes* tidak pernah disertakan pada proses *forwarding*. Difusi langsung merupakan skema yang mengkombinasikan *discovery*, *querying*, dan *routing* dalam satu prosedur. Analoginya adalah *sink* mendeklarasikan ketertarikannya terhadap data tertentu. Langkah pertama adalah melakukan *broadcasting* minat tersebut dari *sink* ke seluruh jaringan. Jalur masuk *routing* dinamakan gradien. Jika beberapa *sink* tertarik dengan data yang sama, difusi akan menggunakan gradien yang telah ada atau membentuk yang baru. Hasilnya adalah struktur dengan *tree* yang saling *overlapping* untuk membawa data dari banyak sumber ke banyak tujuan.

Gambar 1. Paradigma *data-centric*

Pada gambar a hanya mengindikasikan *broadcast* dari satu sumber. Sumber yang menghasilkan data yang sesuai akan mem-*forward* data sepanjang jalur (gambar b), dan *sink* memperkuat jalur terbaik, sementara yang lainnya akan *time out* dan dibuang dari *node*.

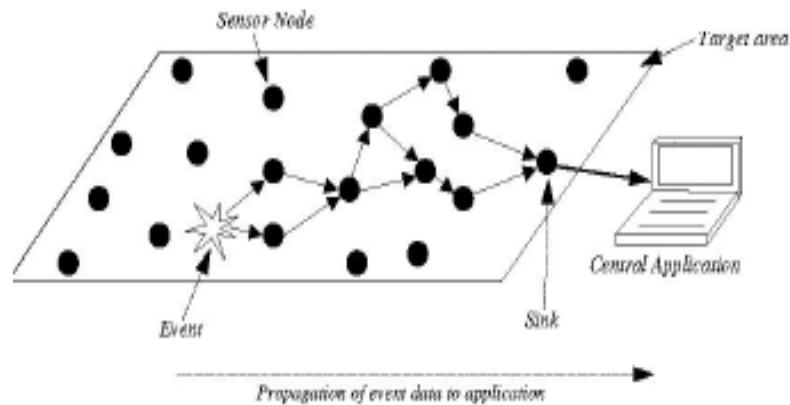
Difusi bekerja dengan baik untuk jaringan yang besar. Jumlah dari gradien yang disimpan dalam *node* tergantung dari jumlah *query* dan kepadatannya, tidak tergantung kepada jumlah dari sumber atau *sink*. Perbedaannya dengan pendekatan *node-centric* adalah tabel disini tidak sebanding dengan jumlah komunikasi *end point* tetapi sebanding dengan kompleksitas dari data yang di-*query*. Hal ini menjadi faktor pertimbangan yang penting ketika memilih paradigma komunikasi dalam jaringan yang besar.

### 2.3 *Position-Centric*

Jika kita melihat *sensor networks* sebagai benda fisik di dunia, maka data yang dilaporkan hampir selalu berkaitan dengan posisinya (misalkan temperatur atau deteksi pergerakan). Persoalan dalam posisi suatu titik telah dipecahkan oleh banyak peneliti, diantaranya *Vision*, *networking*, robotika, dan pemrosesan sinyal. Sebagian besar solusi ini tidak dapat langsung diadaptasi oleh *sensor* berkaitan dengan ukuran dan sumber daya yang terbatas. Karena tujuan utama *sensor networks* adalah untuk mengawasi kondisi fisik, maka operasinya secara intrinsik berkaitan dengan lokasi [3].

Dalam banyak kasus, akan lebih berguna melakukan pengalamatan *sensor* berdasarkan lokasinya daripada alamat IP nya. Pendekatan *Position-Centric* menggunakan posisi dari *nodes* untuk melakukan pengalamatan dan *routing* paket. Dalam bentuk sederhananya, dikenal sebagai *Cartesian Forwarding*, jika sumber mengetahui posisi dari tujuan, maka sumber tersebut akan melakukan *forward* paket ke tetangganya yang dekat dengan tujuan. Metode ini sebenarnya telah ditemukan pada tahun 1970, yang dikenal sebagai *packet radio networks*, awal mula dari jaringan *ad hoc* saat ini.

Pendekatan *Position-Centric* memiliki sejumlah keuntungan dan kerugian. Salah satu keuntungannya adalah tidak diperlukannya tabel *routing* pada jaringan, karena setiap *node* dapat memutuskan bagaimana paket di-*forward* hanya berdasarkan kepada posisi tujuannya dan berdasarkan beberapa informasi tambahan mengenai tetangganya. Aspek positif lainnya adalah tidak tergantung terhadap mobilitas, selama terdapat *intermediate node* yang diketahui posisinya antara sumber dan tujuan, *routing* dapat dilakukan tanpa perlu adanya *discoveries* dan *updating* tabel *routing*. Kerugiannya adalah sumber harus mengetahui posisi tujuannya.



Gambar 2. Paradigma *position-centric*

### 3. TANTANGAN DALAM DESAIN SENSOR NETWORKS

*Sensor networks* memiliki perbedaan yang signifikan bila dibandingkan dengan jaringan *wireless* lainnya dan memiliki beberapa tantangan dalam desainnya [4]:

- *Ad hoc deployment*: Sehubungan dengan ukuran *network* yang besar dan infrastruktur yang terbatas, maka tidak dapat dilakukan konfigurasi secara manual sehingga *network* harus bersifat dinamik dan melakukan konfigurasi sendiri.
- Keterbatasan Energi
- Model Pelaporan Data: *Sensing* data dan pelaporan data tergantung kepada aplikasi dan waktu kritis dari data.

### 4. PENGALAMATAN PADA WIRELESS SENSOR NETWORKS

Dua tipe pengalamatan yang biasanya digunakan untuk *nodes* pada *sensor networks*:

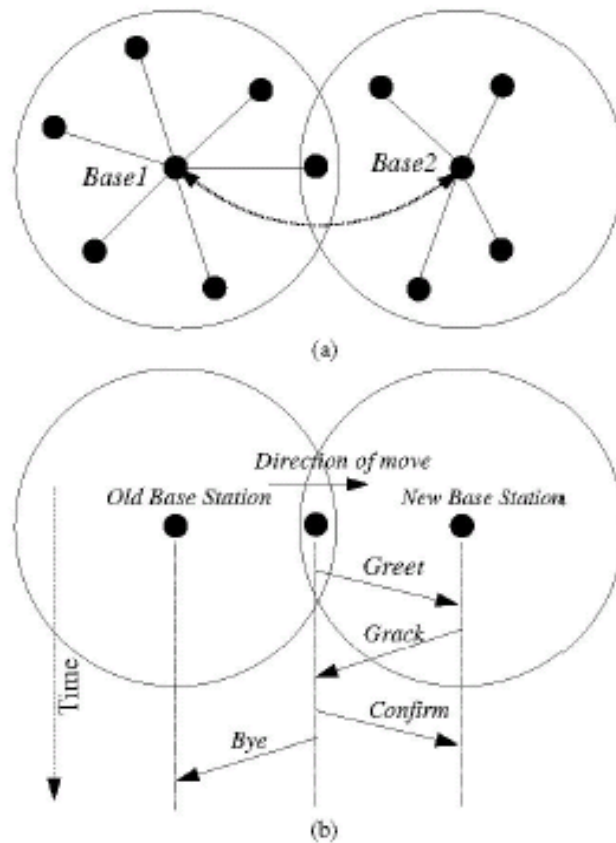
- Alamat *Network* (Alamat Jaringan): bertindak sebagai ID global dari *node* yang digunakan untuk menemukan rute ke tujuan tertentu. Alamat ini harus bersifat unik di seluruh *network*.
- Alamat MAC (*Medium Access Control*) atau alamat lokal: digunakan untuk mengidentifikasi *node* tetangga dalam jangkauan transmisi dari *node*. Alamat ini harus bersifat unik di antara tetangga, tetapi tidak untuk seluruh *network*.

Terdapat beberapa protokol pengalamatan, diantaranya sebagai berikut [4]:

#### a. Pengalamatan dinamik untuk *Wireless Media Access*

Ide dasarnya adalah bahwa pada *wireless network* setiap *node* seharusnya tidak memiliki ID yang unik untuk komunikasi data. Karena paket data yang dikirimkan oleh *node* hanya dilihat oleh tetangganya (*node* yang terdapat di sepanjang jangkauan transmisinya), untuk mengidentifikasi pengirim dan penerima dari paket, tetangga ini harus memiliki alamat yang berbeda. Jadi alamat tersebut dapat digunakan kembali untuk *nodes* yang berjarak cukup jauh sehingga memerlukan jumlah bit yang lebih sedikit dibandingkan dengan alamat global yang unik. Jaringan berisi beberapa *base stations* statik dan setiap *node* terdapat pada *base station* terdekat. Setiap *base station* diberikan alamat yang unik selama proses *setup* jaringan. *Node* dapat melakukan pengiriman paket ke *base station* atau menerima paket dari *base station*. Ketika *node*

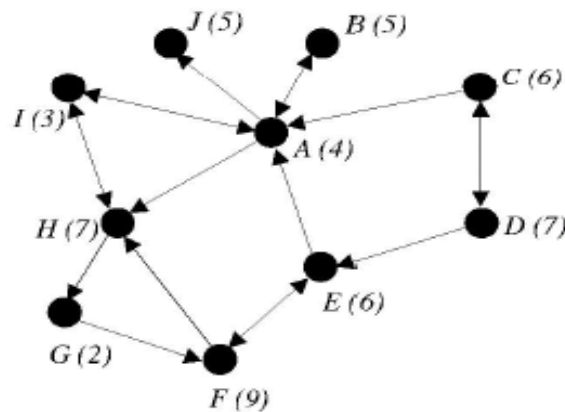
berusaha untuk mengirimkan data ke titik di dalam *base station* lainnya, transfer data akan ditangani oleh *base station* yang bersangkutan. Isu lainnya dalam pengalaman dinamis adalah masalah *handoff*. Ketika *node* bergerak meninggalkan sel dan memasuki sel lainnya, *handoff* terjadi. *Node* kemudian akan diberikan alamat baru dan alamat sebelumnya akan dikembalikan kepada *base station* yang lama untuk digunakan kembali.



Gambar 3. Handover

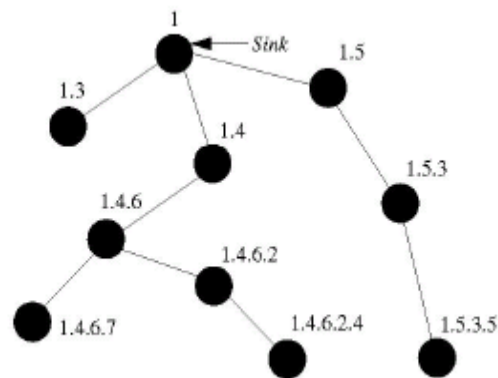
b. Alamat MAC Terdistribusi

Pada pendekatan ini, tetangga dari *node* dikelompokkan menjadi tiga tipe, yaitu *in-neighbors* – tetangga dimana *node* hanya dapat menerima masuknya paket, *out-neighbors* – tetangga dimana *node* hanya dapat mengirimkan paket, dan *bi-neighbors* – tetangga dimana *node* dapat menerima dan mengirimkan paket data (komunikasi dua arah). Skema pengalaman mengikuti prinsip sebagai berikut: Seluruh *bi-neighbors* memiliki alamat yang berbeda dan alamat *in-neighbors* berbeda dengan alamat *bi-neighbors*, tetapi dua *in-neighbors* dapat memiliki alamat yang sama. Pada kondisi awal (inisial), seluruh *node* tidak memiliki alamat tertentu. *Node* memilih alamat acak untuk dirinya dan melakukan *broadcast* alamat tersebut. Setiap *node* menyimpan alamat dari tetangganya. Jika *node* menerima *broadcast* dan mendeteksi adanya konflik dengan alamat *bi-neighbors*-nya, maka *node* tersebut akan mengirimkan pemberitahuan konflik. Jika konflik terjadi pada dua *in-neighbors*, tidak diperlukan tindakan apapun.



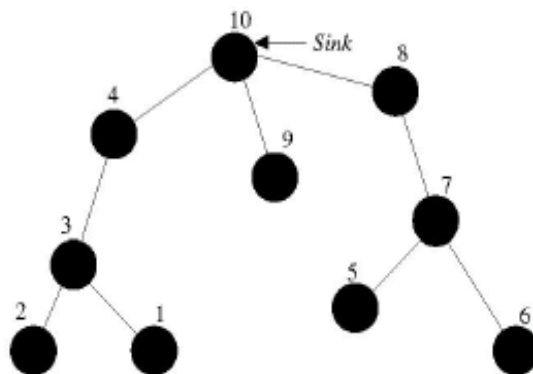
Gambar 4. Konsep alamat MAC terdistribusi

- c. *Pengalamatan Node Naming dengan Efisiensi Energi*  
 Pengalamatan *node naming* didasarkan pada *cluster*. *Node* dikelompokkan ke dalam beberapa *cluster* yang *disjoint* dan *node* diberikan alamat yang unik di dalam *cluster* tersebut. Penggunaan kembali alamat dapat dilakukan untuk *cluster* yang berbeda. Masing-masing *cluster* memiliki alamat yang unik. Ketika *node* melakukan komunikasi di dalam *cluster*, *node* akan menggunakan alamat lokalnya, tetapi ketika *node* berkomunikasi dengan *cluster* lainnya, *cluster ID* akan digabungkan dengan alamat lokalnya.
- d. *Identifikasi Global Terdistribusi*  
 Pada pendekatan ini, *node* diasumsikan tidak memiliki tanda pengenal yang unik ketika disebar. Pendekatan ini bekerja dalam tiga fase, yaitu Fase 1 – Pembentukan *tree* dan pemberian ID sementara, Fase 2 – Pengumpulan ukuran *sub-tree*, dan Fase 3 – pemberian ID permanen yang unik.
- e. *RETRI*  
*RETRI (Random, Ephemeral Transaction Identifiers)* memberikan tanda pengenal (*identifier*) paket ketika transmisi data yang baru terjadi. *Identifier* bersifat probabilistik unik di sepanjang tetangganya dan dapat digunakan kembali. Ketika *node* mengirimkan paket data, *node* akan memilih bit acak sebagai *identifier* dari paket. Ketika paket ini dipecah menjadi bagian-bagian yang kecil, masing-masing bagian memiliki *identifier* yang sama. Teknik ini dinamakan sebagai *address-free fragmentation (AFF)*. *AFF* menggunakan jumlah bit yang lebih sedikit bila dibandingkan dengan global ID.
- f. *TreeCast*  
*Treecast* merupakan pengalamatan global dan arsitektur *routing stateless* untuk *sensor networks*. Pada skema ini, *node* diatur dalam struktur *tree* dan diberikan alamat tergantung kepada posisinya dalam *tree*. *Sink* berfungsi sebagai akar dari *tree*. *Node Parent* menyimpan alamat dari *child* dan memastikan bahwa tidak ada dua *child* yang menggunakan alamat yang sama. *TreeCast* berbeda dengan pendekatan lainnya dalam hal skema alamat tidak hanya mengidentifikasi *node* secara unik pada jaringan, tetapi juga mendukung *routing* data berdasarkan pada alamatnya. Misalkan *sink* (1) ingin mengirimkan paket ke *Node* 1.4.6.2, maka data akan dirutekan mulai dari level 1.4 ke 1.4.6 hingga 1.4.6.2. Pendekatan ini juga bersifat *stateless* dimana *node* tidak menyimpan jalur *routing* yang pernah dilalui [5].

Gambar 5. *TreeCast*

## g. Penomoran Hirarki

Walaupun skema ini dirancang untuk jaringan *ad hoc*, namun skema ini dapat digunakan untuk *sensor network*. Teknik ini mirip dengan teknik *TreeCast* namun dengan penggunaan bit yang lebih efisien. Pada *TreeCast*, pengalamatan berdasarkan pada *level* kedalaman *tree* sehingga *node* yang semakin dalam akan memiliki alamat yang semakin panjang. Pada teknik ini, ketika *tree* terbentuk, maka *sink* akan menjelajahi *tree*. Ketika seluruh *child* telah dikunjungi, *node* kemudian diberikan nomor kunjungan dan nomor ini akan menjadi alamatnya. *Sink* akan mendapatkan nomor terakhir. Untuk jaringan dengan ukuran  $N$  (jumlah *node*  $N$ ), maka bit yang diperlukan untuk memberikan alamat yang unik paling sedikit  $\log_2(N)$ .



Gambar 6. Teknik penomoran hirarki

## 5. KESIMPULAN

Sensor umumnya memiliki sumber daya baterai yang sulit untuk diganti ataupun diisi kembali sehingga sensor mempunyai sifat keterbatasan energi. Untuk membatasi penggunaan energi, sensor tidak melakukan pengiriman ke jarak yang jauh untuk mencapai *sink*, melainkan melakukan komunikasi *multihop*, mengirimkan data melalui tetangganya. Untuk itu diperlukan teknik pengalamatan. Pada Pengalamatan dinamik untuk *Wireless Media Access* tidak semua *node* diberikan ID yang unik, ID yang unik hanya diberikan untuk *node* yang berdekatan sehingga ID tersebut dapat digunakan



kembali untuk *node* yang berjarak cukup jauh. Sistem *cluster* digunakan pada metode *node naming* sedangkan hirarki digunakan pada metode *treecast*.

## REFERENSI

- [1]. Louis, Winnie, Amitava Datta, Rachel Cardell, “*Network Management in Wireless Sensor Networks*”, The University of Western Australia, Australia, 2006.
- [2]. Buratti, Chiara, Andrea Conti, Davide Dardari, Roberto Verdone, “*An Overview on Wireless Sensor Networks Technology and Evolution*”, Journal of Sensors, 2009.
- [3]. Doss, R.C., Chandra L. Pan, W. Chou, M. U. Choudury, “*Dynamic Addressing in Wireless Sensor Networks Without Location Awareness*”, Deakin University, Australia, 2008.
- [4]. Ross, Chellappa, Chandra L. Pan, W. Chou, “*Address Reuse in Wireless Sensor Networks*”, Deakin University, Australia, 2005.
- [5]. Chaudhuri, Santashil, Shu Du, Amit, David Johnson, “*TreeCast: A Stateless Addressing and Routing Architecture for Sensor Networks*”, Rice University, Houston, 2004.