

Tugas
Manajemen Jaringan



Disusun Oleh:

Nama : Novia Tri Lestari

NIM : 090112814190407

JURUSAN SISTEM KOMPUTER

FAKULTAS ILMU KOMPUTER

UNIVERSITAS SRIWIJAYA

2017

Feasibility of SNMP OID compression

Simple management protocol (SNMP) merupakan protocol lapisan aplikasi yang digunakan dalam mengelola jaringan data. Hampir semua vendor jaringan mendukung SNMP, sebuah jumlah yang signifikan dari aktivitas manajemen jaringan saat ini dilakukan dengan kerangka SNMP dengan persentase yang besar dari jaringan perusahaan serta jumlah yang cukup besar dari jaringan penyedia inti layanan. Kerangka SNMP termasuk selain fitur-fitur lainnya, model informasi dan protocol lapisan aplikasi. Model informasi ini merepresentasi virtual dan elemen jaringan untuk dikelola, model informasi dari SNMP grafis diwakili dengan Inverter tree- MIB (management information base) tree. Dalam pohon ini di setiap simpul daun berisi item data yang akan Get dan set, setiap simpul non-daun merupakan pengelompokan item data terkait. Setiap not di pohon ini merupakan identifikasi oleh jalurnya dari akar pohon.

Item data model informasi dicatat dan didaftarkan dalam SNMP permintaan dan respon pesan (daftar Varbind). Setiap item data yang terdapat dalam daftar disebut varbind dan mewakili unsur dalam model informasi yang mana setiap varbind termasuk identifikasi unik Objek Identifikasi (OIT) dan nilai. OID merupakan sub-identifikasi (integer) yang masing-masing sub-identifikasi merupakan nomor alami terkait dengan simpul di pohon MIB.

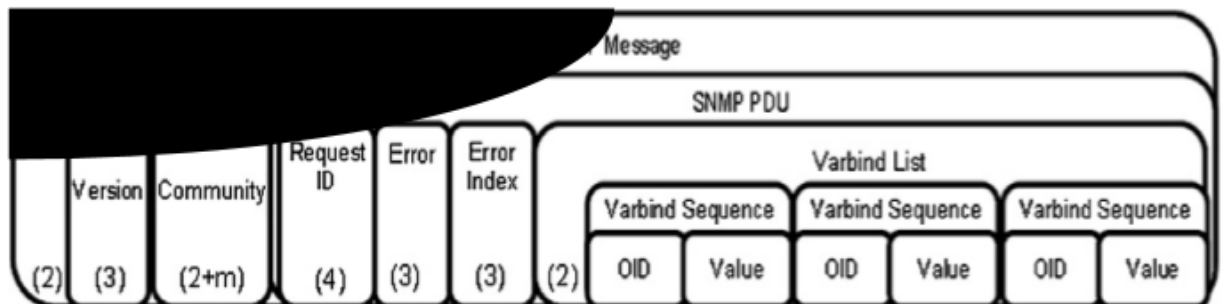
Representasi dari sebuah OID dalam daftar varbind dapat digunakan untuk menghemat bandwidth ketika beberapa item data muncul dalam daftar varbind, optimasi ini disebut sebagai kompresi SNMP OID yang memberikan penghematan lebih dalam siklus CPU disbanding dengan kebutuhan bandwidth untuk tugas-tugas manajemen.

Secara umum, prosedur optimasi kinerja jaringan baik disarankan untuk lapisan transformasi atau untuk lapisan aplikasi. Terdapat tiga algoritma untuk kompresi OID dalam literatur saat ini. Kode algoritma kompresi OID sehubungan dengan objek indentifikasi dalam daftar varbind sebelumnya; OID pertama dikodekan tanpa kompresi. Yang pertama yaitu OID Delta Compression (ODC) menggunakan tiga kombinasi pengkodean yang berbeda untuk representasi dikompresi untuk OID, yaitu single sub-identifier substitution, range of sub-identifier substitution, dan truncation. Ini menghilangkan redundansi lebih dalam OID terkompresi daripada kedua logika lainnya. Yang kedua kode hanya penggantian ekor OID sehubungan dengan OID sebelumnya. Skema ini gagal dalam menghilangkan redundansi di ekor namun

logika kompresi dan dekompresi sederhana untuk kode dan memelihara. Encode hanya sebagai sub-identifikasi substitusi dan pemotongan.

Pada pencarian US Patent terdapat dua aplikasi paten relevan, yang pertama menjelaskan metode kompresi untuk seluruh muatan pesan. Paten kedua menjelaskan metode kompresi untuk penyimpanan data tersrtuktur secara hierarkis.

Format pesan SNMP ditunjukkan pada gambar



Gambar 1 format pesan SNMP.

Seluruh pesan dikodekan menggunakan ASN.1 derivatif. Daftar varbind muncul diakhir pesan sebagai urutan ASN.1. setiap varbind adalah urutan mewakili item data dari underlying MIB. Semua item dalam daftar varbind termasuk daftar varbind dikodekan menggunakan type/Tag-Legth-Value(TLV) format menggunakan ASN.1.

Algoritma kompresi OID diimplementasikan dan terintegrasi dengan kerangka Net-SNMP dalam penelitian tersebut. Dalam ilustrasi yang dilakukan menggunakan istilah

- The `jangkar ID` (A) adalah objek identifikasi dari 1 data yang perlu ditransmisikan
- The `Obyek ID` (O) adalah pbjek identifikasi 2 data yang harus ditransmisikan.

Istilah mengimbangi mengacu jarak (di sub-identifikasi unit) dari terakhir sub-identifikasi dari ID jangkar. Jarak ke kiri yang terakhir sub-identifikasi dari Anchor dianggap positif dan jarak ke kanan dianggap negative. Karena jarak di sub-identifikasi unit antara yang terakhir sub-identifikasi dan itu adalah 0, offset untuk terakhir sub-identifikasi adalah 0 unit. Pencocokan prefix tidak dikirim untuk 2 objek identifikasi hanya bagian ketidakcocokan perlu dikirim. Panjang mismatch juga ditentukan dalam jumlah sub-identifikasi. Panjang urutan ketidakcocokan adalah salah satu output dari metode kompresi. Biasanya sub-identifikasi adalah bilangan dalam kisaran 1-128. Algoritma ini dapat menangani sub-identifikasi yang lebih dari satu octet. Offset informasi tambahan dilakukan bersama disemua encoding

dikompresi. Jika ada lima yang cocok sub-identifikasi maka kompresi menghemat lima unit ruang (karena lima pencocokan sub-identifikasi dianggap sebagai informasi yang berlebihan, maka itu tidak dikirim), dan menggunakan satu unit tambahan ruang untuk kode offset. Hal ini menghasilkan penghematan dari empat unit ruang. Tabungan juga dihitung dalam sub-identifikasi unit.

Input untuk metode pengkodean kompresi objek adalah ID jangkar dan ID objek. Output dari metode ini meliputi offset, mismatch urutan dan panjang urutan ketidakcocokan. Encoding yang dapat diperpanjang ketika ada lebih dari dua objek dalam pesan. Yang pertama objek identifikasi dikirim tanpa kompresi dan sebagai untuk objek kedua, hanya selisish tersebut yang dikirim. Objek kedua diidentifikasi dibangun pada akhir penerima dari jangkar objek identifikasi dan perbedaan dikodekan. Jika ada 3 benda, maka dapat dikodekan dan diterjemahkan sehubungan dengan 2 objek identifikasi dan sebagainya. Ketidaksesuaian antara jangkar dan objek identifikasi yang akan dikompresi.

Tabel 1 Kompresi pengkodean ketika panjang Obyek OID (O) adalah sama dengan jangkar OID (A). Jenis No.

	Masukan untuk metode kompresi: A dan O									deskripsi kasus	Tabungan di jumlah subids	Output dari metode pengkodean			
	1	2	3	4	5	6	7	8	9			Offset dari ujung	Su FFI x panjang	Su FFI x atau embed	
panjang yang sama															
1	SEBUAH	Sebuah 1	Sebuah 2	Sebuah 3	Sebuah 4	Sebuah 5	Sebuah 6	Sebuah 7	Sebuah 8	Sebuah 9	A = O				
	HAI	Sebuah 1	Sebuah 2	Sebuah 3	Sebuah 4	Sebuah 5	Sebuah 6	Sebuah 7	Sebuah 8	Sebuah 9		8	0	0	0
2	SEBUAH	Sebuah 1	Sebuah 2	Sebuah 3	Sebuah 4	Sebuah 5	Sebuah 6	Sebuah 7	Sebuah 8	Sebuah 9	SU FFI X mismatch				
	HAI	Sebuah 1	Sebuah 2	Sebuah 3	Sebuah 4	Sebuah 5	Sebuah 6	Hai 8	Hai 9			6	0	2	Hai 8 - Hai 9
3	SEBUAH	Sebuah 1	Sebuah 2	Sebuah 3	Sebuah 4	Sebuah 5	Sebuah 6	Sebuah 7	Sebuah 8	Sebuah 9	Sebuah 9 tertanam -mismatch				
	HAI	Sebuah 1	Sebuah 2	Sebuah 3	Sebuah 4	Sebuah 5	Hai 6	Hai 7	Sebuah 8	Sebuah 9		6	2	2	Hai 6 - Hai 7
4	SEBUAH	Sebuah 1	Sebuah 2	Sebuah 3	Sebuah 4	Sebuah 5	Sebuah 6	Sebuah 7	Sebuah 8	Sebuah 9	mismatch tersebar				
	SEBUAH	Sebuah 1	Sebuah 2	Sebuah 3	Sebuah 4	Hai 5	Hai 6	Sebuah 7	Sebuah 8	Hai 9		3	0	5	Hai 5 - Hai 9

Catatan: Tabel tidak menunjukkan hamburan dengan beberapa kelompok yang cocok. Jika ada beberapa kelompok yang cocok, kelompok yang paling kiri adalah salah satu yang dipertimbangkan

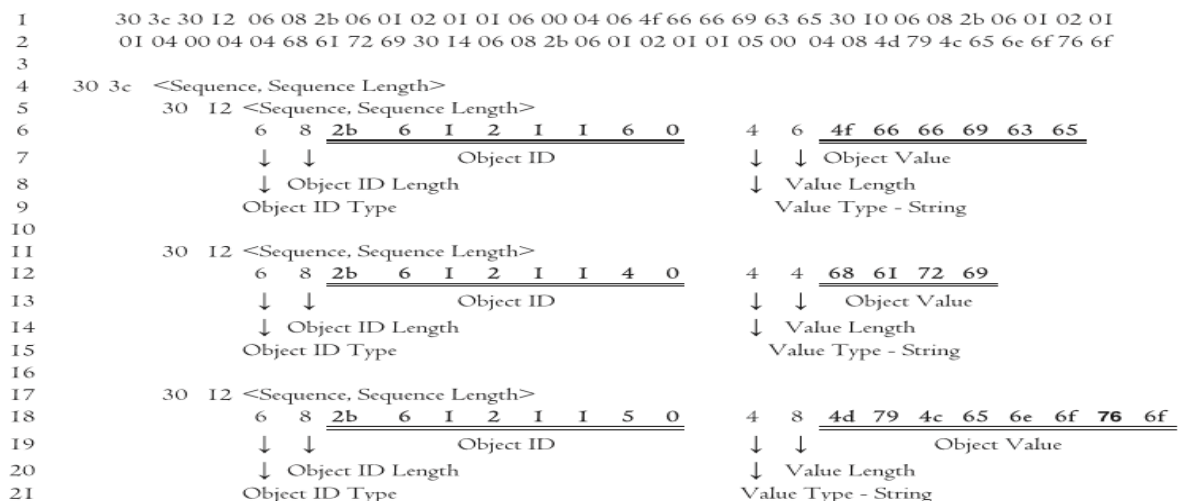
Tabel diatas menggambarkan empat jenis ketidaksesuaian ketika ID jangkar (panjang sub-identifikasi 9) dan ID objek (panjang sub-identifikasi juga 9) yang sama persis, yang berarti node yang sama dikirim dua kali. Semua 9 sub-identifikasi dari objek ID tidak perlu dikirimkan karena tidak ada ketidaksesuaian antara ID objek dan ID jangkar. Proses encoding untuk

memperoleh ID objek dan ID jangkar adalah offset urutan mismatch adalah 0, mismatch urutan panjang juga 0 karena tidak ada ketidakcocokan .

Pada jenis kedua terdapat suffix mismatch, itu berarti urutan kontinu sub-identifikasi hanya berbeda pada dua diakhir identifikasi. Proses encodingnya yaitu offset urutan mismatch 0 karena ketidakcocokan dimulai tepat dari terakhir sub-identifikasi dari ID jangkar. Panjang urutan mismatch 2 karena berakhir 2 sub-identifikasi berada antara ID jangkar dan ID objek.

Pada jenis ketiga ada mismatch tertanam yang menunjukkan bahwa beberapa urutan yang kontinu dari sub-identifikasi berbeda antara jangkar ID dan ID objek. Dapat dilihat pada sub-identifikasi 6 dan 7 berbeda antara ID jangkar dan ID objek oleh karena itu tidak ada kebutuhan untuk mengirimkan 5 sub-identifikasi yang terjadi sebelum 6 sub-identifikasi dan sub-identifikasi yang terjadi setelah 7 sub-identifikasi yang merupakan total tabungan tujuh sub-identifikasi. Proses encodingnya offset urutan mismatch adalah 2 karena ketidakcocokan dimulai dari 7 sub-identifikasi, panjang urutan mismatch adalah 2 karena termasuk 6 sub-identifikasi dan 7 sub-identifikasi.

Pada jenis keempat, terdapat ketidaksesuaian tersebar yang berarti bahwa ketidaksesuaian tersebar dan tidak terjadi dalam urutan yang kontinu, mereka menjadi terputus-putus dalam ID. Pada tabel diatas ke-5, 6, dan 9 sub-identifikasi berbeda antara ID jangkar dan ID objek. Oleh karena itu, tidak ada kebutuhan untuk mengirimkan pertama 4 sub-identifikasi yang terjadi sebelum tanggal 5 sub-identifikasi. Dari 5 sub-identifikasi seterusnya, ada ketidakcocokan jadi meskipun sub-identifikasi 7 dan 8 diantara 5 dan pertandingan ke 9, juga harus diganti sehingga penggantian terjadi dalam urutan yang kontinu. Proses encodingnya offset mismatch 0 karena ketidakcocokan terjadi tepat dari berakhir sub-identifikasi.



Gambar 2 Ilustrasi daftar varbind encoding.

No															
1	30	3c	<Sequence, Sequence Length>												
2			30	12	<Sequence, Sequence Length>										
3			6	8	<u>2b</u>	6	I	2	I	I	6	0	4	6	<u>4f 66 66 69 63 65</u>
4			↓	↓	Object ID							↓	↓	Object Value	
5			↓	Object ID Length							↓	Value Length			
6			Object ID Type										Value Type - String		
7															
8			30	12	<Sequence, Sequence Length>										
9			16	80	2	<u>4</u>	0	x	x	x	x	x	4	4	<u>68 61 72 69</u>
10			↓	↓	↓	Object ID							↓	↓	Object Value
11			↓	↓	Object ID Length							↓	↓		
12			↓	Offset							↓	Value Length			
13			Object ID Type										Value Type - String		
14															
15			30	12	<Sequence, Sequence Length>										
16			26	81	1	<u>5</u>	x	x	x	x	x	x	4	8	<u>4d 79 4c 65 6e 6f 76 6f</u>
17			↓	↓	↓	Object ID							↓	↓	Object Value
18			↓	↓	Object ID Length							↓	↓		
19			↓	Offset							↓	Value Length			
20			Object ID Type										Value Type - String		

Gambar menunjukkan daftar varbind terkompresi ,yang sesuai dengan yang ditunjukkan pada gambar 2. Meskipun objek ke 2 dan objek ke 3 dalam daftar dapat dikodekan sebagai tipe 3, demi ilustrasi varbind kedua dikodekan sebagai tipe 2 dan varbind ketiga dikodekan sebagai jenis 3. Sebelas sedikit octet digunakan untuk menyandikan tiga objek dalam daftar varbind yang terkompresi. Jumlah yang lebih besar dari ukuran terkait benda teap, pesan terkompresi tunggal dapat membawa isi dari tiga pesan normal. Jumlah siklus CPU diselamatkan oleh aplikasi dalam memproses dua pesan tambahan. Dua jenis OID yaitu : tipe 16 mengidentifikasi objek parsial identifikasi untuk semua jenis pengecualian tipe 2 dan tipe 26 mengidentifikasi objek parsial identifikasi untuk 2 jenis. Jenis nilai-nilai ini (16 dan 26) digunakan untuk ilustrasi. Offset dikodekan dengan menambahkan bias 128 (0x80) untuk mengkonversi nilai-nilai negative ke positif. Dengan demikian 0 offset dikodekan sebagai 0x89, 2 dikodekan sebagai 0x82, 2 dikodekan sebagai 0x7D. sebuah octet tunggal mencukupi untuk menyandikan berbagai nilai diimbangi karena jumlah sub-identifikasi dalam OID terbatas.

tabel 4 Membandingkan waktu respon dari kode asil dengan OID kompresi diaktifkan kode untuk SNMP Dapatkan Command. Tidak

	skalar	Datar	kode asil (T_{H0})	Comp tidak memilih (T_{H0})	Comp memilih (T_{0})	Peningkatan (1) T_{H0}/T_{0}	Peningkatan (2) T_n/T_{0}
1	5	0	1287	1117	681	1,89	1,84
2	25	0	335	341	249	1,35	1,37
3	30	0	1161	1152	764	1,52	1,51
4	60	0	1421	1363	879	1,62	1,54
5	90	0	2738	1393	1022	2,68	1,36
6	128	0	2282	2016	1694	1,43	1,26
7	0	30	6863	6669	6299	1,09	1,04
8	0	60	66.074	64.586	67.076	0,99	0,96
9	0	90	135.216	124.415	122.092	1,11	1,02
10	0	128	175.059	168.245	163.413	1,07	1,03
11	15	15	1166	1162	789	1,48	1,47
12	30	30	7696	7480	7387	1,04	1,01
13	45	45	36.227	37.637	38038	0,96	0,99
14	60	68	65.359	72.460	67700	0,97	1,07

tabel 5 Membandingkan waktu respon dengan pilihan kompresi dimatikan dan dihidupkan untuk Dapatkan Command Messal. Tidak

	skalar	Datar	Pengulangan	kode asil (T_{H0})	Comp tidak memilih (T_{H0})	Comp memilih (T_{0})	Peningkatan (1) T_{H0}/T_{0}	Peningkatan (2) T_n/T_{0}
1	0	5	3	1169	1232	616	1,93	2,00
2	0	4	6	1660	1418	1275	1,22	1,11
3	0	10	3	2626	2743	777	3,38	3,53
4	0	10	6	3694	3662	1966	1,89	1,82
5	0	10	9	1643	1632	445	3,69	3,67
6	0	10	10	54.403	55.334	52.953	1,03	1,04
7	2	1	98	28.355	21.084	21.132	1,34	1,00