

Tugas Besar 02

IF3130 Jaringan Komputer

Distance Vector Simulation

Dipersiapkan oleh:
Asisten Laboratorium Sistem Terdistribusi

Didukung oleh:



Start Date: 16 November 2017

End Date: 30 November 2017

I. Background

Dora ingin pergi berpetualang ke Bandung. Saat Dora merogoh tasnya untuk mencari peta, Dora tidak menemukannya. Berikut percakapan Dora dengan Boots saat itu.

Dora : “Boots, apakah kamu tahu dimana petaku? Aku tidak bisa menemukannya di tas.”

Boots : “Maaf Dor, aku memakan petanya kemarin, aku khilaf karena lapar.”

Dora : “Yah”

Dora : “Hai kalian para pembaca, tolong bantu aku yaa”



Karena Dora bersikeras tidak mau memakai Google Maps, Dora harus membuat peta sendiri. Dora pun meminta bantuan kalian untuk membuat peta, caranya adalah dengan Distance Vector Routing Protocol.

II. Deskripsi Tugas

Pada Tugas ini kalian diminta untuk melakukan simulasi Distance Vector Protocol. Kalian akan membuat sebuah program yang akan menerima input berupa topologi network lalu program akan mengeluarkan output berupa routing table.

Pada distance vector, setiap node akan menyimpan sebuah vector yang berisi cost perpindahan dari node itu ke node lain yang berada pada jaringan. Secara berkala, vector tersebut akan dikirimkan kepada tetangga dari node tersebut. Ketika sebuah node menerima vector dari tetangganya, ia akan menambahkan informasi dari vector yang diterima tersebut terhadap vector miliknya.

Awalnya, nilai vector pada setiap node akan diset 1 untuk node yang menjadi tetangga, 0 untuk dirinya, serta tak terhingga untuk node-node lain. Ketika sebuah node mendapat informasi vector dari tetangganya, lalu menemukan bahwa salah satu nilai cost dalam vector yang dimiliki tetangga lebih kecil daripada cost yang ia ketahui atau cost bernilai sama tetapi nomor tetangga yang dimiliki lebih kecil, maka node akan merubah vektornya menjadi cost yang baru. Node juga akan mencatat bahwa packet yang datang akan dikirimkan melalui tetangga tersebut.

III. Spesifikasi Tugas

Program boleh diimplementasi menggunakan bahasa C, C++, Java 8, dan Python. Tidak boleh menggunakan library lain selain library bawaan dari bahasa yang anda pilih. Output routing table dari program akan dinilai dengan menggunakan autograder.

Pada topologi jaringan, terdapat N node yang diidentifikasi dari 1 sampai N . Terdapat juga S skenario penyaluran routing tables. Pada kondisi asli, setiap node akan mengirim informasi routing tables miliknya ke semua node yang terhubung secara langsung; namun agar output program lebih deterministik, pengiriman informasi akan dilakukan sesuai dengan input program. Node tidak dapat mengirim ke node yang tidak terhubung secara langsung dengannya.

Program akan memberikan output berupa routing table dari setiap node setelah program melakukan skenario simulasi pengiriman vector.

Format Input

Baris pertama berisi dua bilangan N dan M , jumlah node pada topologi dan jumlah edge pada topologi.

M baris berikutnya masing-masing berisi 2 bilangan. Baris ke- i berisi u_i dan v_i , yang menyatakan bahwa node u_i terhubung dengan node v_i .

Baris berikutnya berisi sebuah bilangan S , banyaknya skenario yang terjadi.

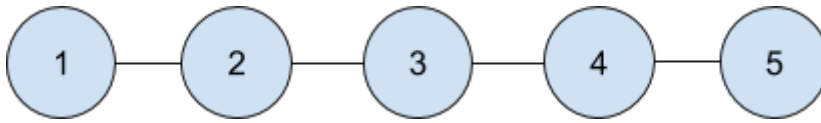
S baris berikutnya masing-masing terdiri dari 2 bilangan. Baris ke- i berisi x_i dan y_i , yang menyatakan bahwa node x_i akan mencoba mengirim informasi ke node y_i .

(yang didalam kurung tidak termasuk input)

Baris ke-	Input
1	5 4 (Jumlah node dan jumlah edge)
2	1 2 (Node 1 terhubung dengan Node 2)
3	2 3 (Node 2 terhubung dengan Node 3)
4	3 4 (Node 3 terhubung dengan Node 4)
5	4 5 (Node 4 terhubung dengan Node 5)
6	5 (terdapat 5 skenario)
7	1 2 (Node 1 mengirim informasi ke Node 2)
8	2 3 (Node 2 mengirim informasi ke Node 3)
9	2 1 (Node 2 mengirim informasi ke Node 1)
10	3 4 (Node 3 mengirim informasi ke Node 4)
11	4 5 (Node 4 mengirim informasi ke Node 5)

NB : Sebuah node dapat terhubung dengan lebih dari 2 node, tidak harus seperti contoh diatas.

Sebagai contoh, input diatas akan memiliki topologi sebagai berikut:



Simulasi program

Saat inialisasi, node hanya memiliki informasi ke node yang terhubung secara langsung. Node lain yang tidak terhubung diisi dengan -1.

Contoh routing table node 1 pada saat inialisasi

Node	Distance	NextHop
1	0	1
2	1	2
3	-1	-1
4	-1	-1
5	-1	-1

Kemudian program akan mengupdate routing table node sesuai urutan skenario program

1. Input baris ke-7 : 1 2

Node 1 mengirim informasi routing table ke node 2. Karena node 2 sudah mempunyai informasi node 1, node 2 tidak mengalami perubahan.

Routing table node 2

Node	Distance	NextHop
1	1	1
2	0	2
3	1	3
4	-1	-1
5	-1	-1

2. Input baris ke-8 : 2 3

Node 2 mengirim informasi ke Node 3. Node 3 mengupdate routing table nya, yaitu informasi mengenai node 1. Node 2 mempunyai distance 1 dari Node 1, dan Node 3 mempunyai distance 1 dengan Node 2, maka distance Node 3 dengan Node 1 adalah $1+1 = 2$

Routing table node 3 saat inisialisasi

Node	Distance	NextHop
1	-1	-1
2	1	2
3	0	3
4	1	4
5	-1	-1

Routing table node 3 setelah *update*

Node	Distance	NextHop
1	2	2
2	1	2
3	0	3
4	1	4
5	-1	-1

3. Input baris ke-9 : 2 1

Node 2 mengirim informasi ke node 1. Node 1 mengupdate informasi tentang jarak node 3.

Routing table di node 1 setelah *update*

Node	Distance	NextHop
1	0	1
2	1	2
3	2	2
4	-1	-1
5	-1	-1

4. Input baris ke-10 : 3 4

Node 3 mengirim informasi ke Node 4. Node 4 mengupdate informasi tentang keberadaan Node 1 dan 2.

Routing table node 4 saat inisialisasi

Node	Distance	NextHop
1	-1	-1
2	-1	-1
3	1	3
4	0	4
5	1	5

Routing table node 4 setelah *update*

Node	Distance	NextHop
1	3	3
2	2	3
3	1	3
4	0	4
5	1	5

5. Input baris ke-11 : 4 5

Node 4 mengirim informasi ke Node 5. Node 5 kemudian mengupdate informasi tentang keberadaan Node 1, 2, dan 3 dari routing table milik Node 4.

Routing table node 5 saat inisialisasi

Node	Distance	NextHop
1	-1	-1
2	-1	-1

3	-1	-1
4	1	4
5	0	5

Routing table node 5 setelah *update*

Node	Distance	NextHop
1	4	4
2	3	4
3	2	4
4	1	4
5	0	5

Format Output

Keluarkan N^2 baris. N baris pertama berisi informasi *routing table* pada node 1, N baris berikutnya berisi informasi pada node 2, ..., N baris terakhir berisi informasi pada node N.

Untuk setiap N baris pada setiap node, baris ke-i berisi 2 buah bilangan d_i dan h_i , yang menyatakan *distance* dan *nextHop* node ke-i pada *routing table* yang bersangkutan. Jika ada beberapa kemungkinan h_i , ambillah h_i yang terkecil. Kedua bilangan tersebut dipisahkan oleh spasi.

Penulisan *routing table* sesuai dengan urutan node, mulai dari *routing table* pada node 1, node 2, dan seterusnya.

Sebagai contoh, berikut adalah keluaran dari input diatas.

(yang didalam kurung tidak termasuk output)

0 1 (Routing table Node 1)
1 2
2 2
-1 -1
-1 -1
1 1 (Routing table Node 2)
0 2
1 3
-1 -1
-1 -1

2 2 (Routing table Node 3)

1 2

0 3

1 4

-1 -1

3 3 (Routing table Node 4)

2 3

1 3

0 4

1 5

4 4 (Routing table Node 5)

3 4

2 4

1 4

0 5

Batasan

- $1 \leq N \leq 1.000$
- $0 \leq M \leq N * (N - 1) / 2$
- $0 \leq S \leq 100.000$
- $1 \leq u_i, v_i \leq N, u_i \neq v_i$
- $1 \leq x_i, y_i \leq N, x_i \neq y_i$
- Input skenario dijamin valid

IV. Pengumpulan dan Deliverables

Tugas dikerjakan dalam kelompok sebanyak maksimal 3 orang (anggota kelompok tidak boleh dari kelas yang berbeda). Kerjakan persoalan yang diberikan pada deskripsi di atas maksimal 30 November 2017 pukul 23:59 WIB. Perhatikan bahwa nilai yang diambil merupakan hasil dari penilaian *grader* sehingga keterlambatan pengumpulan dapat mengakibatkan nilai 0 (nol). Platform grading yang digunakan akan diberikan maksimal *H-7 deadline* melalui milis.

Kumpulkan tugas besar ini ke *autograder* yang telah diberikan dengan dalam format TB2_KXX_YY_ABC.zip dengan XX menandakan nomor kelas, YY menandakan nomor kelompok, dan ABC adalah nama kelompok anda. File zip tersebut terdiri dari:

- Direktori src yang berisi source code yang anda buat.
- Direktori data yang berisi data yang digunakan untuk testing program.
- Makefile. Buatlah sehingga kompilasi program dapat dilakukan hanya dengan pemanggilan command 'make' saja.
- File README.md yang berisi:
 - Petunjuk penggunaan program.
 - Bagaimana anda mensimulasikan sebuah node dalam program anda? Bagaimana proses pengiriman pesan antar node terjadi? Jelaskan dengan menggunakan nama fungsi dan/atau struktur data yang anda gunakan.
 - Pembagian tugas. Sampaikan dalam list pengerjaan untuk setiap mahasiswa. Sebagai contoh: XXXX mengerjakan fungsi YYYY, ZZZZ, dan YYZZ.
 - Jawaban dari pertanyaan dibawah ini:

- Apakah perbedaan dari routing protocol distance-vector dan link state? Manakah yang lebih baik untuk digunakan?
- Pada implementasinya saat ini manakah yang lebih banyak digunakan, distance-vector atau link state? Kenapa?

V. Referensi

http://www.cs.bu.edu/fac/byers/courses/791/F99/scribe_notes/cs791-notes-990923.html