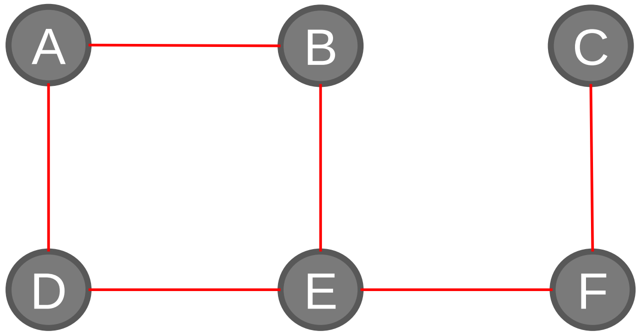
Routing Algorithms – Continued…

CS60002: Distributed Systems

Antonio Bruto da Costa
Research Scholar,
Dept. of Computer Sc. & Engg.,
Indian Institute of Technology Kharagpur

Example: Netchange – Working Example

Let us observe how this network comes up.



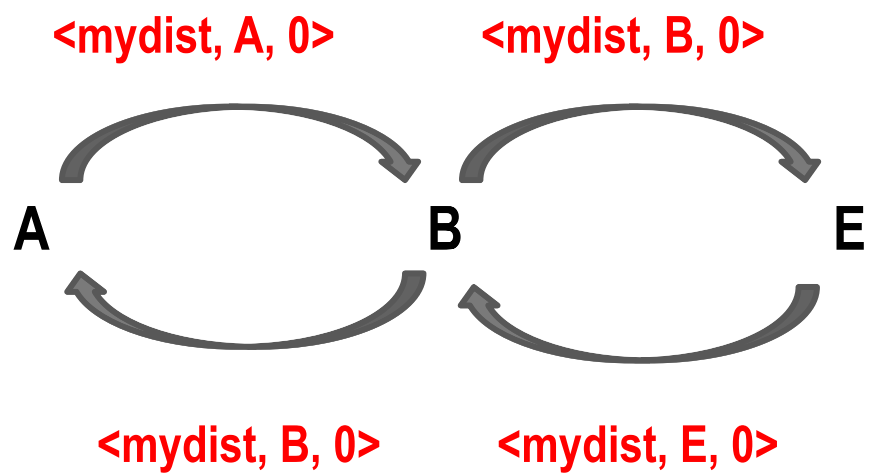
- $ndis_u\ [w,v] = N$ for all w in $Neigh_u$
- $V = \{A, B, C, D, E, F\}$
- $D_u\ [v] = N$ (shown as -), $Nb_u\ [v] = udef$, for all v in V
- $D_A\ [A] = D_B\ [B] = D_C\ [C] = D_D\ [D] = D_E\ [E] = D_F\ [F] = 0$
- $Nb_A\ [A] = Nb_B\ [B] = Nb_C\ [C] = Nb_D\ [D] = Nb_E\ [E] = Nb_F\ [F] = local$
- Every node u sends <mydist, u, 0> to all its neighbors.

Node Initialization:

$Neigh_A = \{B,D\}$ $Neigh_D = \{A,E\}$

$Neigh_B = \{A,E\}$ $Neigh_E = \{B,D,F\}$

$Neigh_C = \{F\}$ $Neigh_F = \{C,E\}$



Example

After first exchange of messages- ROUND-1

A	A	B	C	D	E	F
B	-	0	-	-	-	-
D	-	-	-	0	-	-

B	A	B	C	D	E	F
A	0	-	-	-	-	-
E	-	-	-	-	0	-

C	A	B	C	D	E	F
F	-	-	-	-	-	0

D	A	B	C	D	E	F
A	0	-	-	-	-	-
E	-	-	-	-	0	-

E	A	B	C	D	E	F
B	-	0	-	-	-	-
D	-	-	-	0	-	-
F	-	-	-	-	-	0

F	A	B	C	D	E	F
C	-	-	0	-	-	-
E	-	-	-	-	0	-

Recomputing for each message

D_A	0	1	-	1	-	-
N_A	I	B	u	D	u	u
D_B	1	0	-	-	1	-
N_B	A	I	u	u	E	u
D_C	-	-	0	-	-	1
N_C	u	u	I	u	u	F
D_D	1	-	-	0	1	-
N_D	A	u	u	I	E	u
D_E	-	1	-	1	0	1
N_E	u	B	u	D	I	F
D_F	-	-	1	-	1	0
N_F	u	u	C	u	E	I

A	A	B	C	D	E	F
B	1	0	-	-	1	-
D	1	-	-	0	1	-

B	A	B	C	D	E	F
A	0	1	-	1	-	-
E	-	1	-	1	0	1

C	A	B	C	D	E	F
F	-	-	1	-	1	0

D	A	B	C	D	E	F
A	0	1	-	1	-	-
E	-	1	-	1	0	1

E	A	B	C	D	E	F
B	1	0	-	-	1	-
D	1	-	-	0	1	-
F	-	-	1	-	1	0

F	A	B	C	D	E	F
C	-	-	0	-	-	1
E	-	1	-	1	0	1

Example

ROUND 2:

A	A	B	C	D	E	F
B	1	0	-	-	1	-
D	1	-	-	0	1	-

B	A	B	C	D	E	F
A	0	1	-	1	-	-
E	-	1	-	1	0	1

C	A	B	C	D	E	F
F	-	-	1	-	1	0

D	A	B	C	D	E	F
A	0	1	-	1	-	-
E	-	1	-	1	0	1

E	A	B	C	D	E	F
B	1	0	-	-	1	-
D	1	-	-	0	1	-
F	-	-	1	-	1	0

F	A	B	C	D	E	F
C	-	-	0	-	-	1
E	-	1	-	1	0	1

Recomputing for each message

	A	B	C	D	E	F
D_A	0	1	-	1	2	-
N_A	I	B	u	D	B	u
D_B	1	0	-	2	1	2
N_B	A	I	u	A	E	E
D_C	-	-	0	-	2	1
N_C	u	u	I	u	F	F
D_D	1	2	-	0	1	2
N_D	A	A	u	I	E	E
D_E	2	1	2	1	0	1
N_E	B	B	F	D	I	F
D_F	-	2	1	2	1	0
N_F	u	E	C	E	E	I

A	A	B	C	D	E	F
B	1	0	-	2	1	2
D	1	2	-	0	1	2

B	A	B	C	D	E	F
A	0	1	-	1	2	-
E	2	1	2	1	0	1

C	A	B	C	D	E	F
F	-	2	1	2	1	0

D	A	B	C	D	E	F
A	0	1	-	1	2	-
E	2	1	2	1	0	1

E	A	B	C	D	E	F
B	1	0	-	2	1	2
D	1	2	-	0	1	2
F	-	2	1	2	1	0

F	A	B	C	D	E	F
C	-	-	0	-	2	1
E	2	1	2	1	0	1

Example

ROUND 3:

A

	A	B	C	D	E	F
B	1	0	-	2	1	2
D	1	2	-	0	1	2

B

	A	B	C	D	E	F
A	0	1	-	1	2	-
E	2	1	2	1	0	1

C

	A	B	C	D	E	F
F	-	2	1	2	1	0

D

	A	B	C	D	E	F
A	0	1	-	1	2	-
E	2	1	2	1	0	1

E

	A	B	C	D	E	F
B	1	0	-	2	1	2
D	1	2	-	0	1	2
F	-	2	1	2	1	0

F

	A	B	C	D	E	F
C	-	-	0	-	2	1
E	2	1	2	1	0	1

Recomputing for each message

	A	B	C	D	E	F
D_A	0	1	-	1	2	3
N_A	I	B	u	D	B	B
D_B	1	0	3	2	1	2
N_B	A	I	E	A	E	E
D_C	-	3	0	3	2	1
N_C	u	F	I	F	F	F
D_D	1	2	3	0	1	2
N_D	A	A	E	I	E	E
D_E	2	1	2	1	0	1
N_E	B	B	F	D	I	F
D_F	3	2	1	2	1	0
N_F	E	E	C	E	E	I

A

	A	B	C	D	E	F
B	1	0	3	2	1	2
D	1	2	3	0	1	2

B

	A	B	C	D	E	F
A	0	1	-	1	2	3
E	2	1	2	1	0	1

C

	A	B	C	D	E	F
F	3	2	1	2	1	0

D

	A	B	C	D	E	F
A	0	1	-	1	2	3
E	2	1	2	1	0	1

E

	A	B	C	D	E	F
B	1	0	3	2	1	2
D	1	2	3	0	1	2
F	3	2	1	2	1	0

F

	A	B	C	D	E	F
C	-	3	0	3	2	1
E	2	1	2	1	0	1

Example

ROUND 4:

Node	Neighbor	A	B	C	D	E	F
A	B	1	0	3	2	1	2
A	D	1	2	3	0	1	2
B	A	0	1	-	1	2	3
B	E	2	1	2	1	0	1
C	F	3	2	1	2	1	0
D	A	0	1	-	1	2	3
D	E	2	1	2	1	0	1
E	B	1	0	3	2	1	2
E	D	1	2	3	0	1	2
E	F	3	2	1	2	1	0
F	C	-	3	0	3	2	1
F	E	2	1	2	1	0	1

Recomputing for each message

	A	B	C	D	E	F
D_A	0	1	4	1	2	3
N_A	I	B	B	D	B	B
D_B	1	0	3	2	1	2
N_B	A	I	E	A	E	E
D_C	4	3	0	3	2	1
N_C	F	F	I	F	F	F
D_D	1	2	3	0	1	2
N_D	A	A	E	I	E	E
D_E	2	1	2	1	0	1
N_E	B	B	F	D	I	F
D_F	3	2	1	2	1	0
N_F	E	E	C	E	E	I

Node	Neighbor	A	B	C	D	E	F
A	B	1	0	3	2	1	2
A	D	1	2	3	0	1	2
B	A	0	1	-	1	2	3
B	E	2	1	2	1	0	1
C	F	3	2	1	2	1	0
D	A	0	1	-	1	2	3
D	E	2	1	2	1	0	1
E	B	1	0	3	2	1	2
E	D	1	2	3	0	1	2
E	F	3	2	1	2	1	0
F	C	-	3	0	3	2	1
F	E	2	1	2	1	0	1

Example

ROUND 5:

A						
	A	B	C	D	E	F
B	1	0	3	2	1	2
D	1	2	3	0	1	2

B						
	A	B	C	D	E	F
A	0	1	4	1	2	3
E	2	1	2	1	0	1

C						
	A	B	C	D	E	F
F	3	2	1	2	1	0

D						
	A	B	C	D	E	F
A	0	1	4	1	2	3
E	2	1	2	1	0	1

E						
	A	B	C	D	E	F
B	1	0	3	2	1	2
D	1	2	3	0	1	2
F	3	2	1	2	1	0

F						
	A	B	C	D	E	F
C	4	3	0	3	2	1
E	2	1	2	1	0	1

Routing with Compact Routing Tables

- The algorithms studied require each node to maintain a routing table with an entry for each possible destination.
  - How many destinations can there be?

- Is there a way to reorganize the routing tables and still remember which channel caters to which destinations?
  - Indexing the routing table by channel.
  - Each channel of the node has an entry informing which destinations must be routed via that channel.

Tree-Labeling Scheme

$Z_N = \{ 0, 1, \ldots, N\text{-}1 \}$

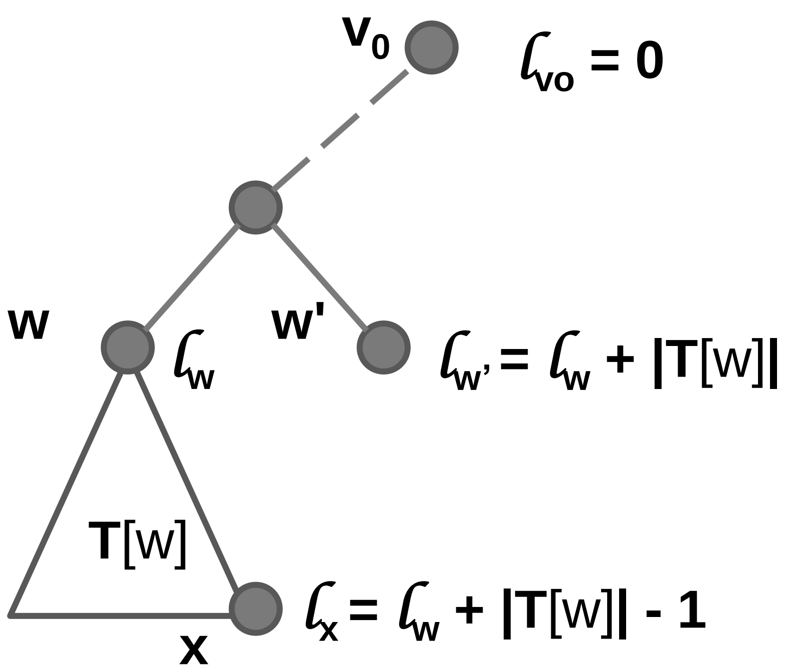
Cyclic Intervals

The cyclic interval [a,b) in Z_N is the set of integers defined by:

$$[a, b) = \begin{cases} \{a, a+1, \ldots, b-1\} & \text{if } a<b \\ \{0, \ldots, b-1, a, \ldots, N-1\} & \text{if } a \geq b \end{cases}$$

- $[a,a) = Z_N$
- The *compliment* of [a,b) is [b,a) when $a \neq b$
- The cyclic interval [a,b) is called *linear* if $a < b$.

Tree-Labeling Scheme



- The nodes of a tree *T* can be numbered such that
  - For each outgoing channel of each node, the set of destinations that must be routed via that channel is a *cyclic interval.*
- Let v_0 be the root of the tree.
- For each node *w* let T[*w*] denote the subtree of T rooted at w
- Number the nodes in the tree such that for each *w* the numbers assigned to the nodes in T[w] form a *linear interval*.

- Let $[a_w, b_w)$ denote the interval assigned to nodes in T[w].
- Node *w* forwards to a son *u* packets with destinations in T[u], with labels in $[a_u, b_u)$
- Node *w* forwards to a father packets with destinations not in T[u], with labels in $Z_N \setminus [a_w, b_w) = [b_w, a_w)$

Tree-Labeling Scheme

Procedure for Interval Forwarding (for node u) :

if $d = l_u$

then deliver the packet locally

else begin

select α_i s.t $d \in [\alpha_i, \alpha_{i+1}]$;

send packet via the channel labeled with α_i

end ;

- Representing a single interval requires 2 log N bits
- A node u has many intervals at the node.
- Only the start point of the interval for a channel is stored; the end point being the next begin point of an interval at the same node.
- At node u, the begin point of the interval for channel *uw* is given by:

$$\alpha_{uw} = \begin{cases} l_w & \text{if } w \text{ is a son of } u \\ l_u + |T[u]| & \text{if } w \text{ is the father of } u \end{cases}$$