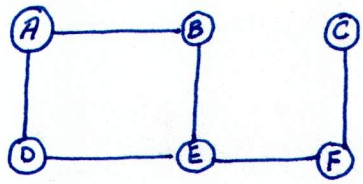
NETCHANGE ALGORITHM



1. Every node initializes.

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

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

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

$\Rightarrow ndis_u[w, v] = N \quad \forall w \in Neigh_u$

$\Rightarrow V = \{A, B, C, D, E, F\}$

$\Rightarrow D_u[v] = N$, $Nb_u[v] = udef$, $\forall u \in V$

$\Rightarrow D_A[A] = D_B[B] = D_C[C] = D_D[D] = D_E[E] = D_F[F] = 0$

$Nb_A[A] = Nb_B[B] \ldots = local$

$\Rightarrow$ Every node u sends $\langle mydist, u, 0\rangle$ to all its Neighbours.

2. After the first exchange of messages.

$ndis_A$

	A	B	C	D	E	F
A	N	N	N	N	N	N
B	N	0	N	N	N	N
C	N	N	N	N	N	N
D	N	N	N	0	N	N
E	N	N	N	N	N	N
F	N	N	N	N	N	N

$ndis_B$

	A	C	D	E	F
A	0	N	N	N	N
C	N	N	N	N	N
D	N	N	N	N	N
E	N	N	N	0	N
F	N	N	N	N	N

$ndis_C$

	A	B	D	E	F
A	N	N	N	N	N
B	N	N	N	N	N
D	N	N	N	N	N
E	N	N	N	N	N
F	N	N	N	N	0

$ndis_D$

	A	B	C	E	F
A	0	.	.	.	.
B	.	.	.	.	.
C	.	.	.	.	.
E	.	.	.	.	.
F	.	.	.	.	0

$ndis_E$

	A	B	C	D	F
A	.	.	.	.	.
B	.	0	.	.	.
C	.	.	.	.	.
D	.	.	.	0	.
F	.	.	.	.	0

$ndis_F$

	A	B	C	D	E
A	.	.	.	.	.
B	.	.	.	.	.
C	.	.	0	.	.
D	.	.	.	.	.
E	.	.	.	.	0

Recomputing for each message.

D_A [0, 1, N, 1, N, N] $\quad D_B$ [1, 0, N, N, 1, N] $\quad D_C$: [N, N, 0, N, N, 1]

Nb_A [l, B, u, D, u, u] $\quad Nb_B$ [A, l, u, u, E, u] $\quad Nb_C$: [u, u, l, u, u, F]

D_D · [1, N, N, 0, 1, N] $\quad D_E$ [N, 1, N, 1, 0, 1] $\quad D_F$ [N, N, 1, N, 1, 0]

Nb_D [A, u, u, l, E, u] $\quad Nb_E$ [u, B, u, D, l, F] $\quad Nb_F$ [u, u, C, u, E, l]

3. Each node sends a mydist msg to its neighbours for distances that change.

$ndis_A$

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

[0, 1, N, 1, 2, N] [l, B, u, D, B, u]

B

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

[1, 0, N, 2, 1, 2] [A, l, u, A, E, E]

C

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

[N, N, 0, N, 2, 1] [u, u, l, u, F, F]

D

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

[1, 2, N, 0, 1, 2] [A, A, u, l, E, E]

E

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

[2, 1, 2, 1, 0, 1] [B, B, F, D, l, F]

F

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

[N, 2, 1, 2, 1, 0] | D

[u, E, C, E, E, l] | Nb

ndis

ROUND 3

	A	B	C	D	E	F
ndis	A B C D E F B 1 0 N 2 1 2 D 1 2 N 0 1 2	A B C D E F A 0 1 N 1 2 N E 2 1 2 1 0 1	A B C D E F F N 2 1 2 1 0	A B C D E F A 0 1 N 1 2 N E 2 1 2 1 0 1	A B C D E F B 1 0 N 2 1 2 D 1 2 N 0 1 2 F N 2 1 2 1 0	A B C D E F C N N 0 N 2 1 E 2 1 2 1 0 1
D	[0, 1, N, 1, 2, 3]	[1, 0, 3, 2, 1, 2]	[N, 3, 0, 3, 2, 1]	[1, 2, 3, 0, 1, 2]	[2, 1, 2, 1, 0, 1]	[3, 2, 1, 2, 1, 0]
Nb	[l, B, u, D, B, B]	[A, l, E, A, E, E]	[u, F, l, F, F, F]	[A, A, E, l, E, E]	[B, B, F, D, l, F]	[E, E, C, E, E, l]

ROUND 4

	A	B	C	D	E	F
ndis	B 1 0 3 2 1 2 D 1 2 3 0 1 2	A 0 1 N 1 2 3 E 2 1 2 1 0 1	F 3 2 1 2 1 0	A 0 1 N 1 2 3 E 2 1 2 1 0 1	B 1 0 3 2 1 2 D 1 2 3 0 1 2 F 3 2 1 2 1 0	C N 3 0 3 2 1 E 2 1 2 1 0 1
D	[0, 1, 4, 1, 2, 3]	[1, 0, 3, 2, 1, 2]	[4, 3, 0, 3, 2, 1]	[1, 2, 3, 0, 1, 2]	[2, 1, 2, 1, 0, 1]	[3, 2, 1, 2, 1, 0]
Nb.	[l, B, B, D, B, B]	[A, l, E, A, E, E]	[E, F, l, F, F, F]	[A, A, E, l, E, E]	[B, B, F, D, l, F]	[E, E, C, E, E, l]

ROUND 5

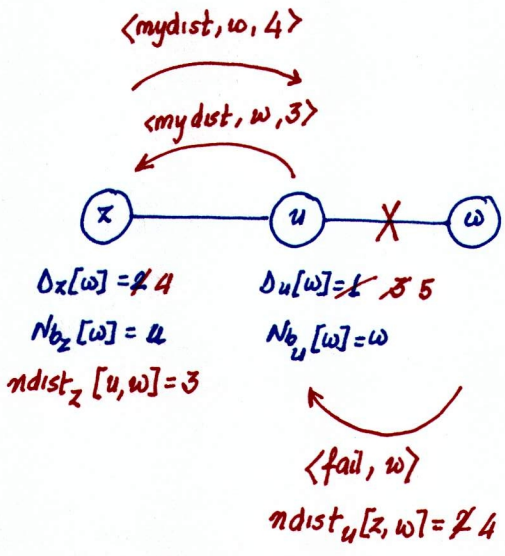
	A	B	C	D	E	F
ndis	B 1 0 3 2 1 2 D 1 2 3 0 1 2	A 0 1 4 1 2 3 E 2 1 2 1 0 1	F 3 2 1 2 1 0	A 0 1 4 1 2 3 E 2 1 2 1 0 1	B 1 0 3 2 1 2 D 1 2 3 0 1 2 F 3 2 1 2 1 0	C 4 3 0 3 2 1 E 2 1 2 1 0 1

Count to Infinity

Stable State :

$D_z[\omega] = 2$ $D_u[\omega] = 1$

$Nb_z[\omega] = u$ $Nb_u[\omega] = \omega$

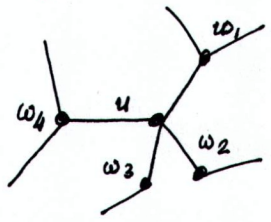


Routing with Compact Routing Tables

⇒ Netchange, etc. require each node to maintain a RT with an entry for each possible destination.

⇒ No. of destinations can be large.

★ Reorganize the RT such that the table contains for each channel of the node an entry telling which destination must be routed via the channel.



dest	chan.
v_1	ω_2
u	-
v_3	ω_3
⋮	⋮
v_N	ω_1

⇒

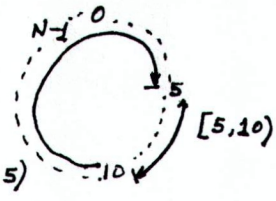
chan.	destinations
ω_1	$\ldots, v_N$
ω_2	$v_1, \ldots$
ω_3	$\ldots, v_j, \ldots$
ω_4	$\ldots$

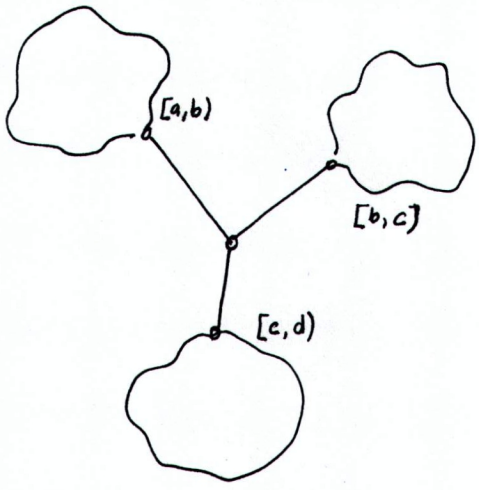
TREE-LABELING SCHEME

Cyclic Interval

$[a,b) = \{a, a+1, \ldots, b-1\}$ if $a < b$

$= \{0, \ldots, b-1, a, \ldots, N-1\}$ if $a \geq b$





Addresses on the internet

⇒ Using a network mask