# Routing Algorithms

*CS60002: Distributed Systems*

**Pallab Dasgupta**
**Professor,**
**Dept. of Computer Sc. & Engg.,**
**Indian Institute of Technology Kharagpur**

# Main Features

- ***Table Computation***
  - The routing tables must be computed when the network is initialized and must be brought up-to-date if the topology of the network changes

- ***Packet Forwarding***
  - When a packet is to be sent through the network, it must be forwarded using the routing tables

# Performance Issues

***Correctness*** : The algorithm must deliver every packet to its ultimate destination

***Complexity*** : The algorithm for the computation of the tables must use as few messages, time, and storage as possible

***Efficiency*** : The algorithm must send packets through *good* paths

***Robustness*** : In the case of a topological change, the algorithm updates the routing tables appropriately

***Fairness*** : The algorithm must provide service to every user in the same degree

# Good paths …

***Minimum hop***: The cost of a path is the number of hops

***Shortest path***: Each channel has a non-negative cost – the path cost is the sum of the cost of the edges. Packets are routed along shortest paths.

***Minimum delay/congestion***: The bandwidth of a path is the minimum among the bandwidths of the channels on that path.

***Most robust path:*** Given the probability of packet drops in each channel, packets are to be routed along the most reliable paths.

# Destination-based Forwarding

*// A packet with destination d was received or generated at node u*

**if** $d = u$

**then** deliver the packet locally

**else** send the packet to $table\_lookup_u(d)$

# Floyd-Warshall Algorithm

```
begin
        S = Φ;
        forall u, v do
                if u = v then D[u, v] = 0
                else if uv ∈ E then D[u, v] = w_{u,v}
                else D[u, v] = ∞ ;
        while S ≠ V do  // Loop invariant: ∀u, v: D[u, v] = d^S(u, v)
                begin pick w from V \ S;
                    forall u ∈ V do
                        forall v ∈ V do
                                D[u, v] = min{ D[u, v], D[u, w] + D[w, v] }
                    S = S U { w }
                end
end
```

# The simple distributed algorithm

*// For node u …*

var $S_u$ : set of nodes;

$D_u$ : array of weights;

$Nb_u$ : array of nodes;

begin

$S_u = \Phi$;

forall $v \in V$ do

if $v = u$ then

begin $D_u[v] = 0$; $Nb_u[v] = udef$ end

else if $v \in Neigh_u$ then

begin $D_u[v] = w_{u,v}$; $Nb_u[v] = v$ end

else begin $D_u[v] = \infty$ ; $Nb_u[v] = udef$ end;

# The simple distributed algorithm contd…

```
while Su ≠ V do
          begin pick w from V \ Su;      // All nodes must pick the same w
                    if u = w
                       then broadcast the table Dw
                       else receive the table Dw
                    forall v ∈ V do
                       if Du[w] + Dw[v] < Du[v] then
                       begin
                              Du[v] = Du[w] + Dw[v] ;
                              Nbu[v] = Nbu[w]
                       end ;
                 Su = Su U { w }
          end
end
```

# Important property of the simple algorithm

Let *S* and *w* be given and suppose that

(1) for all $u$, $D_u[w] = d^S(u, w)$ and

(2) if $d^S(u, w) < \infty$ and $u \neq w$, then $Nb_u[w]$ is the first channel of a shortest S-path to $w$

Then the directed graph $T_w = (V_w, E_w)$, where

$(u \in V_w \Leftrightarrow D_u[w] < \infty)$ and

$(ux \in E_w \Leftrightarrow (u \neq w \wedge Nb_u[w] = x))$

is a tree rooted towards *w.*

# Toueg's improvement

- **Toueg's observation:**
  - A node $u$ for which $D_u[w] = \infty$ at the start of the $w$-pivot round does not change its tables during the $w$-pivot round.
  - If $D_u[w] = \infty$ then $D_u[w] + D_w[v] < D_u[v]$ is false for every $v$.
  - Consequently, only the nodes that belong to $T_w$ need to receive $w$'s table, and the broadcast operation can be done efficiently by sending the table $D_w$ only via the channels that belong to the tree $T_w$

# The Chandy-Misra Algorithm

var $D_u[v_0]$ : weight init $\infty$ ;
$Nb_u[v_0]$ : node init *udef* ;

**For node $v_0$ only:**

begin $D_{v0}[v_0] = 0$ ;
forall $w \in Neigh_{v0}$ do send $\langle$mydist, $v_0$, 0$\rangle$ to $w$
end

**Processing a $\langle$mydist, $v_0$, $d\rangle$ message from neighbor $w$ by $u$:**

{ $\langle$mydist, $v_0$, $d\rangle \in M_{wu}$ }
begin receive $\langle$mydist, $v_0$, $d\rangle$ from $w$ ;
if $d + \omega_{uw} < D_u[v_0]$ then
begin $D_u[v_0] = d + \omega_{uw}$ ; $Nb_u[v_0] = w$ ;
forall $x \in Neigh_u$ do send $\langle$mydist, $v_0$, $D_u[v_0]$ $\rangle$ to $x$
end
end

# The Netchange Algorithm

- Computes routing tables according to *minimum-hop* measure
- <u>Assumptions:</u>
  - **N1:** The nodes know the size of the network ($N$)
  - **N2:** The channels satisfy the FIFO assumption
  - **N3:** Nodes are notified of failures and repairs of their adjacent channels
  - **N4:** The cost of a path equals the number of channels in the path
- <u>Requirements:</u>

**R1.** If the topology of the network remains constant after a finite number of topological changes, then the algorithm terminates after a finite number of steps.

**R2.** When the algorithm terminates, the tables $Nb_u[v]$ satisfy

(a) if $v = u$ then $Nb_u[v] = local$ ;

(b) if a path from $u$ to $v \neq u$ exists then $Nb_u[v] = w$, where $w$ is the first neighbor of $u$ on a shortest path from $u$ to $v$ ;

(c) if no path from $u$ to $v$ exists then $Nb_u[v] = udef.$

# The Netchange Algorithm

var $Neigh_u$ : set of nodes ; // The neighbors of $u$

$D_u$ : array of 0 .. N ; // $D_u[v]$ estimates $d(u,v)$

$Nb_u$ : array of nodes ; // $Nb_u[v]$ is preferred neighbor for $v$

$ndis_u$ : array of 0 .. N ; // $ndis_u[w, v]$ estimates $d(w,v)$

**Initialization:**

begin forall $w \in Neigh_u$, $v \in V$ do $ndis_u[w, v] = N$ ;

forall $v \in V$ do

begin $D_u[v] = N$ ; $Nb_u[v] = udef$ end ;

$D_u[u] = 0$ ; $Nb_u[u] = local$ ;

forall $w \in Neigh_u$ do send $\langle$mydist, $u$, 0$\rangle$ to $w$

end

# The Netchange Algorithm contd.

**Procedure *Recompute*( $v$ ):**

```
begin if v = u
        then begin D_u[v] = 0 ; Nb_u[v] = local end
        else begin              // estimate distance to v
                d = 1 + min{ ndis_u[w,v] : w ∈ Neigh_u } ;
                if d < N then
                  begin    D_u[v] = d ;
                           Nb_u[v] = w with 1 + ndis_u[w,v] = d
                  end
                else begin D_u[v] = N ; Nb_u[v] = udef end
            end ;
        if D_u[v] has changed then
            forall x ∈ Neigh_u do send ⟨mydist, v, D_u[v] ⟩ to x
end
```

# The Netchange Algorithm contd.

**Processing a ⟨mydist, *v*, *d*⟩ message from neighbor *w*:**

{ A ⟨mydist, *v*, *d*⟩ is at the head of $Q_{wv}$ }
begin receive ⟨mydist, *v*, *d*⟩ from *w* ;
    $ndis_u[w,v] = d$ ; *Recompute*( *v* )
end

**Upon failure of channel *uw*:**

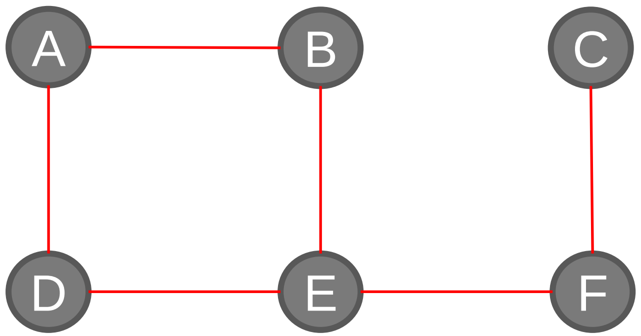
begin receive ⟨fail, *w*⟩ ; $Neigh_u = Neigh_u \setminus \{w\}$ ;
    forall $v \in V$ do *Recompute*( *v* )
end

**Upon repair of channel *uw*:**

begin receive ⟨repair, *w*⟩ ; $Neigh_u = Neigh_u \cup \{w\}$ ;
    forall $v \in V$ do
        begin $ndis_u[w,v] = N$ ;
            send ⟨mydist, *v*, $D_u[v]$ ⟩ to *w*
        end
end

# Example

**Let us observe how this network comes up.**



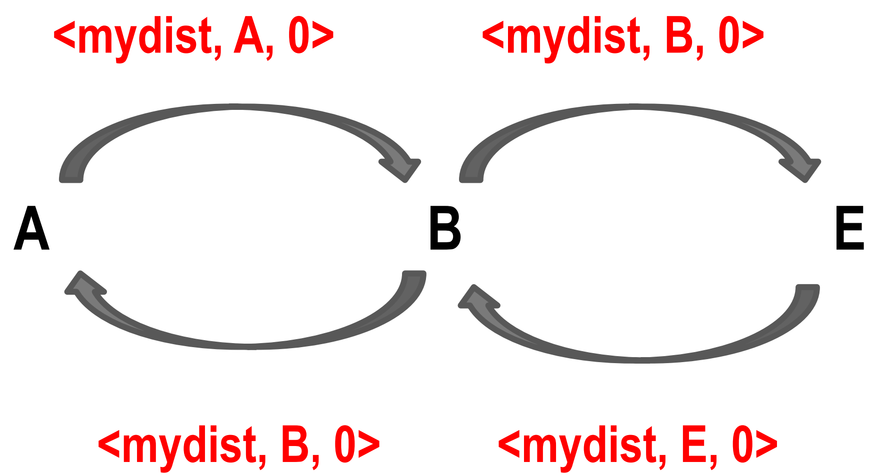

**Node Initialization:**

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

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

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

- $ndis_u$ [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.



# 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 | - |
| $N_A$ | l | B | u | D | B | u |
| $D_B$ | 1 | 0 | - | 2 | 1 | 2 |
| $N_B$ | A | l | u | A | E | E |
| $D_C$ | - | - | 0 | - | 2 | 1 |
| $N_C$ | u | u | l | u | F | F |
| $D_D$ | 1 | 2 | - | 0 | 1 | 2 |
| $N_D$ | A | A | u | l | E | E |
| $D_E$ | 2 | 1 | 2 | 1 | 0 | 1 |
| $N_E$ | B | B | F | D | l | F |
| $D_F$ | - | 2 | 1 | 2 | 1 | 0 |
| $N_F$ | u | E | C | E | E | l |

**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:**

**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 |

*Recomputing for each message*

| | | | | | | |
|---|---|---|---|---|---|---|
| $D_A$ | 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 |


**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 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

$Z_N = \{ 0, 1, \dots, N-1 \}$

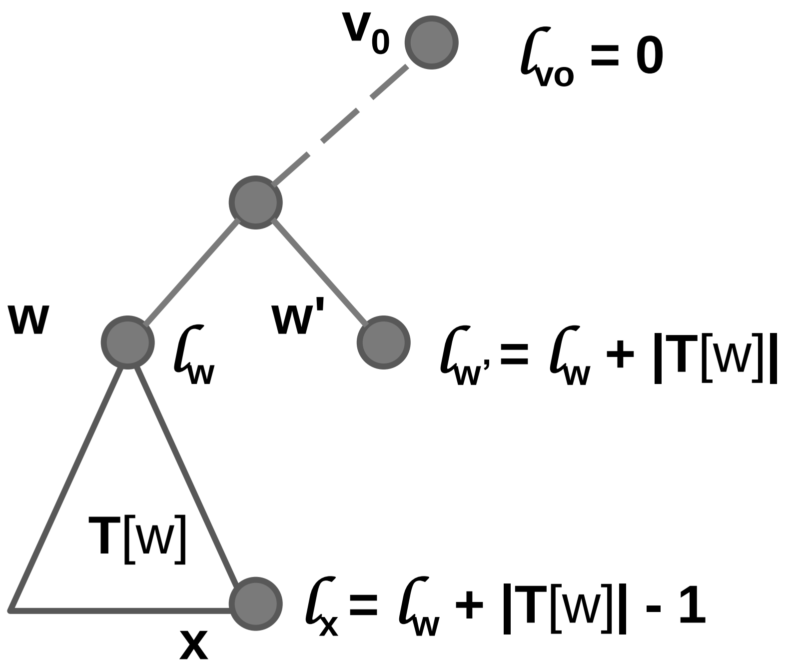
<u>Cyclic Intervals</u>

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

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

- $[a,a) = Z_N$
- 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 ∈ [α_i , α_{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:

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