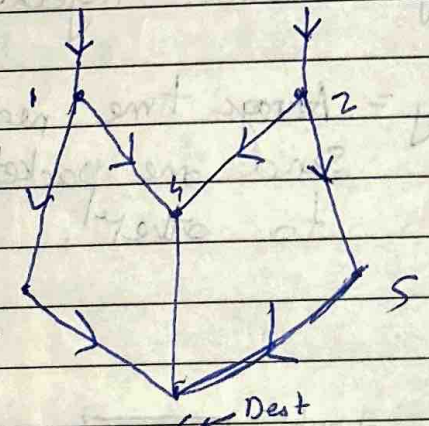
PAGE NO.:
DATE: / /

Routing

- Different packets between same origin and destination may follow different paths in standard ~~compit~~ computer network connections to govern this property, planner routing is needed.
- Generally routing is done by a complex set of algorithms co-operating with each other.
- Routing demands communication (if needed) b/w any two pairs of nodes. (Broadcasting is involved)
- It needs to incorporate for ~~com~~ the ~~saturations~~ situations of node and link failure
- Routing must allow modification of routes in case of congestion of network



All links have capacity 10.

Both 1 and 2 get 5 unit input

1-3-6 and 2-5-6

No problem

1-4-6 and 2-4-6

Some problem

Teacher's Signature.....

1 gets 5 unit and 2 gets 15 units

Potential Routing:

1-3-6 5 unit

2-4-6 7.5 units

2-5-6 7.5 unit

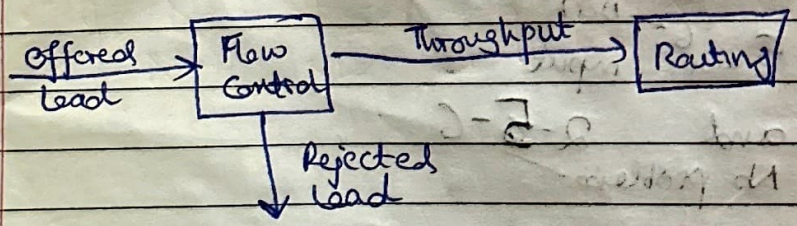
Routing
- Selection of route
- Delivery of message through selected route

(keeping record of selected routes in Routing table)

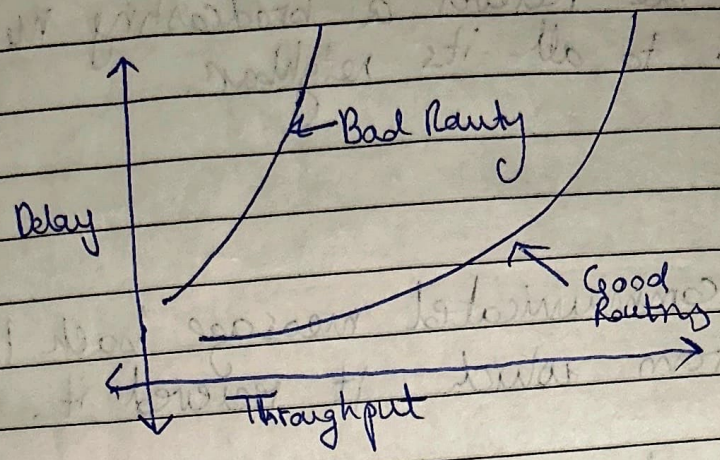
Two main performance measures
- Throughput
- Average Packet Delay

Throughput = ~~Offered~~ load − Rejected load

Average packet Delay = Average time needed to send one packet from one node to other.



Teacher's Signature.....



Classification

1st

Centralised ← Calculation of path happens at one node

vs

Distributed ← Calculation of path happens at various nodes.

2nd

Static ← Path remain same for same origin and destination

vs

Adaptive ← Paths change for different packets.

Broadcasting and Flooding

Nodes may need to communicate with all other nodes (or some other multiple nodes) time to time (to inform ~~fail~~ failure, change in protocol, etc) and hence they used to broadcast time to time.

One such method is called Flooding

- Origin communicates with all its neighbours.

Teacher's Signature.....

- Each node once recieves a broadcasting message communicates to all its neighbours.

Rules

1. No node communicated message back to the node from which it recieved it.

2. If a node has communicated j^{th} message of i^{th} node to k^{th} node to its neighbours already then it wont communicate i^{th} message of i^{th} node to k^{th} node to its neighbours for $L \leq j$.

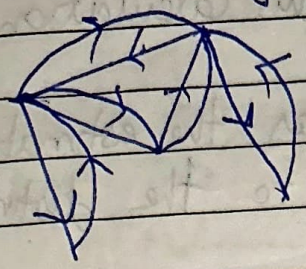
Spanning Tree Broadcasting

- Construct a spanning tree and call the origin of the message its root. All nodes preserves the info about spanning tree
- Root communicates message to all its members
- Every node, if recieves broadcasting message, communicates to all its neighbours away from root.
- If N is the no. of nodes, Complete communication in N-1 steps.

Teacher's Signature.....

PAGE NO. :
DATE : / /

Shortest Path Algorithm



Views networks as an edge weighted directed graph.

Interpret edge weights as lengths

- A link can have different lengths in different directions
- Each path b/w two nodes has a length equal to the sum of the lengths of the links. lengths may change as time progresses.
- A shortest path routing algorithm routes each packet along a minimal length path b/w origin and destination.

Example

Bellman Ford Method

D_i = Estimated ~~distance shortest~~ shortest distance of node i to the destination and d_{ij} is the length of the link (i,j)

Then,

$$D_i := \min_j \{d_{ij} + D_j\}$$

Teacher's Signature.....

- Each node periodically does this calculation
- $d_{ij} + D_j$ may be veiwed as the estimate of shortest distance from node i to the destination going through j

D_i^h : Distance of shortest path from i to 1 that contain $\leq h$ edges

$D_1^h = 0 \quad \forall h$

$D_i^0 = \infty \quad \forall i \neq 1$

We shall prove

$$D_i^{h+1} = \min_j [d_{ij} + D_j^h]$$

The algorithm terminates if

$$D_i^h = D_i^{h-1} \quad \forall i \quad \text{for some } h. \qquad (*)$$

Why?

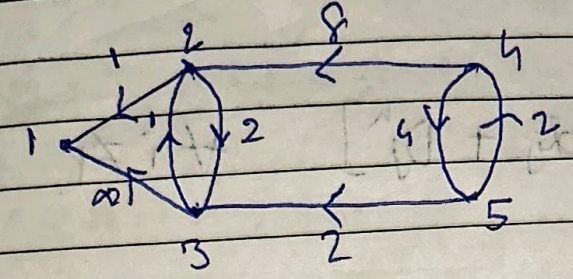
$$D_i^{h+1} = \min_j [d_{ij} + D_j^h]$$

$$= \min_j [d_{ij} + D_i^{h-1}]$$

$$= D_i^h$$

Teacher's Signature.....

So (*) $\Rightarrow D_i^{h+1} = D_i^h \quad \forall h \quad \forall i$



$$\begin{cases} D_1^1 = 0 \\ D_2^1 = 1 \end{cases} \qquad \begin{matrix} D_4^1 = \infty \\ D_3^1 = 4 \end{matrix} \qquad D_5^1 = \infty$$

$D_1^2 = 0 \quad D_2^2 = 1 \quad D_3^2 = 2 \quad D_4^2 = 9 \quad D_5^2 = 6$

$D_1^3 = 0 \quad D_2^3 = 1 \quad D_3^3 = 2 \quad D_4^3 = 9 \quad D_5^3 = 4$

$D_1^4 = 0 \quad D_2^4 = 1 \quad D_3^4 = 2 \quad D_4^4 = 8 \quad D_5^4 = 4$

Bellman Ford Algorithm

Theorem: Consider the B-F algorithm with initial conditions $D_i^0 = \infty \quad \forall i \neq 1$, then

1. The N scalars D_i^h generated by the algorithm are ~~equally~~ equal to the shortest ($\leq h$) walk lengths from node i to node 1.

2. The alg. terminates after a finite number of iterations. Furthermore, number of iterations needed is $\leq N$ (no. of nodes) and at termination D_i^h is the shortest path length from i to 1.

Teacher's Signature.....

PAGE NO.:
DATE: / /

Proof: By induction

Recall,

$$D_i^{h+1} = \min_j [d_{ij} + D_j^h] \quad \forall i \neq 1$$

$$D_1^h = 0 \quad \forall h$$

$$D_i^0 = \infty \quad \forall i \neq 1$$

Note, $D_i^1 = d_{i1} \quad \forall i \neq 1$

So, $D_i^1 =$ shortest (≤ 1) walk from i to 1

Suppose D_i^k is equal to shortest ($\leq k$) walk length from i to 1 $\forall k \leq h$. We shall show that D_i^{h+1} is the shortest ($\leq h+1$) walk length from i to 1.

Observe, shortest ($\leq h+1$) walk from i to 1 either consists of less than $h+1$ links ~~with the first being (i,j) for some $j \neq 1$ followed by an h link walk from j to 1 where node 1 is not repeated~~ in which case its length is equal to D_i^h, or else it consists of $h+1$ links where first is (i,j), $j \neq 1$ followed by an h-link walk from j to 1 where node 1 is not repeated. The latter walk must be a shortest ($\leq h$) walk from j to 1.

Thus,

shortest ($\leq h+1$) walk length $= \min\left\{D_i^h, \min_{j \neq 1} [d_{ij} + D_j^{h-1}]\right\}$

Teacher's Signature.....

By induction
$D_j^k \leq D_j^{k-1} \quad \forall\, k < h$

[Since the set of (k-1) walks from node j to 1 contains the corresponding set of ($\leq k-1$) walks.]

Therefore,

$$D_i^{h+1} = \min_j [d_{ij} + D_j^h]$$

$$\leq \min_j [d_{ij} + D_j^{h-1}]$$

$$= D_i^h$$

Also $D_i^h \leq D_i^1 = d_{i1} = d_{i1} + D_1^h$

So from (#)

Shortest ($\leq h+1$) walk length $= \min \{ D_i^h, \min_j [d_{ij} + D_j^h] \}$

$= \min \{ D_i^h, D_i^{h+1} \}$

Hence,
shortest ($\leq h+1$) walk length $= D_i^{h+1}$

This proves that D_i^h generated by the algorithm is the shortest ($\leq h$) path from i to 1.

Next, we observe that if BF terminates under h iterations we must have $D_i^k = D_i^h \;\forall i$ and $k > h$ so we can not have the length of shortest path reduced by allowing more/add more links.

Teacher's Signature.....

Now, $\forall i$ and h $\exists$ a path that is shortest ($\leq h$) walk from i to 1 (search in a finite set) and the corresponding length is D_i^h by part (a) it is paths ~~have~~ has no repeated nodes to be shortest it must be of length $\leq N-1$

Hence, it follows that the
$D_i^N = D_i^{N-1}$ $\forall i$

So, alg. terminates in step $\leq N$

HW Q. What is complexity of BF algorithms?

Routing → Broadcasting (Flooding)
→ ~~Point to point~~ shortest path (Bellman-Ford)

Bellman-Ford continued

1) Complexity
2) Shortest path construction
3) ~~Convergence~~. Uniqueness

Complexity Theorem:
Number of computations required for BF algorithm is $O(N^3)$ where N is the number of nodes.

Proof: In every step one ~~computer~~ computes ~~for~~ D_i for N nodes, via minimization over almost $N-1$ neighbours. So each steps

Teacher's Signature.....

takes $O(N^2)$ computations.

- The algorithm converges in atmost N steps. So total no. of computations $O(N^3)$

Shortest-Path Construction in Bellman Ford

Bellman-Ford Eqn: (*)

$$D_i = \min_j [d_{ij} + D_j]$$

$D_1 = 0$

BF gives a solution to

(*) Path from i to 1: For each i. Pick a neighbour j s.t $j \neq$ D_j is node for in BF algo.

So basically apart from arranging a value D_i to each node i, BF assigns a neighbour j to each i (j is the one that gives the minimization in $D_i = \min_j [d_{ij} + D_j]$

To get the path from i to 1 for each ~~update~~ vertex go to the neighbour designated by BF alg. —(#)

Thm: (#) gives a path from each node to 1:

If: Enough to show $\forall$ i once the process (#) starts, no ~~need to~~ nodes get repeated.

By contradiction: Assume $i, j_1, j_2, \cdots j_k, j_{k+1}, \cdots j_L$ appears where $j_k = j_L$

Teacher's Signature.....

Note:

$$D_{jk} = d_{j_k j_{k+1}} + D_{j_{k+1}}$$

$$= d_{j_k j_{k+1}} + d_{j_{k+1} j_{k+2}} + D_{j_{k+2}}$$

$$D_{jk} = d_{j_k j_{k+1}} + d_{j_{k+1} j_{k+2}} + \cdots + d_{j_{t-1} j_t} + D_{j_t}$$

$$D_{jk} = d_{j_k j_{k+1}} + d_{j_{k+1} j_{k+2}} + \cdots + d_{j_{t-1} j_t} + D_{j_k}$$

$$\because D_{j_t} = D_{j_k}$$

$$\Rightarrow d_{j_k j_{k+1}} + d_{j_{k+1} j_{k+2}} + \cdots + d_{j_{t-1} j_t} = 0$$

Contradiction!!

Hence in almost in step, reaches to 1.

Theorem

(#) gives shortest length path from each i to 1.

Proof:

Proceed earlier that D_i gives length of shortest path from i to 1 and observe that (#) gives a path of length D_i.

Q. Potentially BF eqn may have multiple solutions?

Ans. No!!

Teacher's Signature.....

He will give 1 or 2 results from BF alg or problems which he will do next week.

PAGE NO.:
DATE:

Theorem: BF eqn have unique solutions.

Proof: Suppose it has another set of solutions $\tilde{D}_i$

So $\tilde{D}_i = 0$

As from $\tilde{D}_i$ too one can construct a path from i to i; $\tilde{D}_i \geq D_i \ \forall i$

Now it is enough to show
$D_i \geq \tilde{D}_i \ \forall i$

Run BF alg. first with usual initial conditions
$D_i^0 = 0$, $D_i^0 = \infty$ to get $D_i \ \forall i$

Next Run BF alg. with initial conditions
$D_j^0 = \tilde{D}_i \quad \forall i$ ——— (1)

Convergence in 1st step it self at $\hat{D}_i$ is (1)
$D_i \geq \tilde{D}_i$

Step 1
$$D_i^1 = \min_j [d_{ij} + D_j^0] \geq \tilde{D}_i^1 = \min_j [d_{ij} + \tilde{D}_j^0]$$

$$D_i^2 = \min_j [d_{ij} + D_j^1] \geq \tilde{D}_i^2 = \min_j [d_{ij} + \tilde{D}_j^1]$$

Teacher's Signature.....

PAGE NO.:
DATE: / /

All edges have +ve weights.

- First of the shortest path to node 1 must be edge from some neighbour of 1.
- The next shortest of the shortest paths will either be the single edge path from the next closest neighbour of 1 or shortest 1 edge path through the previously chosen closest neighbour.

- To formalise this we view each node i as being labelled with an estimate D_i of the shortest path lengths to node 1.

When the estimate gets certain, we regard the node as being permanently labelled.

Let P be the set of potentially labelled nodes
$d_{ij} = \infty$ if i and j not.

~~Shortest~~ w.

start with $P = \{1\}$
$D_1 = 0$
$D_j = d_{j1}$ for $j \neq 1$

Step 1 Find $i \notin P$
s.t $D_i = \min_{j \notin P} D_j$

$P := P \cup \{i\}$

Teacher's Signature.....

V

PAGE NO.:
DATE: / /

Dijkstra's Algorithm

Target Node : 1 All edges have +ve weights.

- The shortest of the shortest path to node 1 must be in edge from some neighbour of 1.
- The next shortest of the shortest paths will either be the single edge path from the next closest neighbour of 1 or shortest 1 edge path through the previously chosen closest neighbour.
- To formalise this we view each node i as being labelled with an estimate D_i of the shortest path lengths to node 1.

When the estimate gets certain, we regard the node as being permanently labelled.

Let P be the set of potentially labelled nodes
$d_{ij} = \infty$ if i and j not .

~~Shortest~~ w.

start with $P = \{1\}$
$D_1 = 0$
$D_j = d_{j1}$ for $j \neq 1$

Step 1 Find $i \notin P$
s.t $D_i = \min_{j \notin P} D_j$

$P := P \cup \{i\}$

Teacher's Signature.....

If P contains all nodes
STOP

Step2 $\forall j \notin P$ (updation of labels)

$D_j := \min[D_j, d_{ji} + D_i]$

Go To STEP 1

Why the algorithm works?

At the ~~beginning~~ beginning of each iteration of step1

(a) $D_i \leq D_j \quad \forall i \in P$ and $j \notin P$

(b) D_j ~~$\forall j$~~ is the shortest distance (path) from j to 1 using all nodes from P ~~except possibly those belonging to P~~ other than j

Cond (a) is satisfied initially and since $d_{ji} \geq 0$ and $D_i = \min_{j \notin P} D_j$ it is preserved by the formula

$D_j := \min[D_j, d_{ji} + D_i] \quad \forall j \notin P$

To show cond (B) by induction

It holds initially suppose that it holds at the beginning of some iteration of step 1.

Let i be the node added to P in that step and let D_k the label of each node k at the beginning

Teacher's Signature.....

Dijkstra's Algorithm contd.

of that iteration then (b) holds for $j=i$ by induction. It also holds $\forall j \in P$ by (a) and induction. Finally for $j \notin P \cup \{i\}$ consider a path from j to i which is shortest among those with all nodes except j, $j \in P \cup \{j\}$ and D'_j be the corresponding shortest distance.

Such a path must ~~contain~~ consists of an edge (j,k) for some $k \in P \cup \{j\}$ followed by a shortest path from k to i with nodes in $P \cup \{i\}$

Since we just saw that length of their path from k to i in D_k

$$D'_j = \min_{k \in P \cup \{i\}} [d_{jk} + D_k]$$

$$= \min \left[\min_{k \in P} [d_{jk} + D_k], d_{ji} + D_i \right]$$

Induction implies,

$$D_j = \min_{k \in P} [d_{jk} + D_k)$$

So we get,

$$D'_j = \min [D_j, d_{ji} + D_i]$$

Thus in step 2

D'_j is the shortest distance D_j from j to i using all nodes in $P \cup \{i\}$ except j.

Teacher's Signature.....

PAGE NO.:
DATE: / /

Remark: $\forall i \in P$, D_i is the shortest distance from i to 1.

PF exercise.

Exe. If # of nodes is N then show that # of computations needed is $\leq O(N^2)$

Bellman Ford

$D_i^0 = 1$

$D_i^0 = \infty$

$D_i^{h+1} = \min_j [d_{ij} + D_j^h]$

keep iterating

solution in atmost # node steps.

Exe.: Read about synchronous distributed Bellman Ford Algorithm and its difficulties.

Distributed Asynchronous Bellman Ford Algorithm

Set up: 1) Target Node: 1

2) At each time t and each $i \neq 1$

$D_j(t)$:= Estimate of the shortest distance of each node j ($j \in N(i)$) that has been communicated to i.

$D_i(t)$ = Estimate of shortest distance of node i

Teacher's Signature.....

which was last computed at node i according to B-F iteration.

3) Process starts at t_0 $\forall$
$D_i(t) = 0 \quad \forall t \geq t_0$
$D_i^j(t) = 0 \quad \forall t \geq t_0$ and $j \in N(i)$

4) Each node i has d_{ij} $\forall j \in N(i)$ which is assumed to remain constant between t_i's:
$t_0 < t_1 < t_2 < t_3 < \cdots$
$t_m \to \infty$ as $m \to \infty$

Algorithm:

1) i updates $D_i(t)$ according to:

$$D_i(t) := \min_{j \in N(i)} [d_{ij} + D_j^i(t)]$$

and leaves $D_j^i(t)$, $j \in N(i)$ unchanged

2) Node i receives from one or more neighbours $j \in N(i)$ the values of D_j which was computed at ~~same time~~ some earlier time, updates the estimate D_j^i and leaves all other estimates unchanged.

Let T^i be the set of times for which an update by node i as in case 1 occurs and T_j^i be the set of times when a message is recieved at i from node j, as in case 2.

Teacher's Signature.....

PAGE NO.:
DATE: / /

Proof:

We assume:

Assumption 1: Nodes never stop updating their own estimates and receiving messages from neighbours [ie T_i^i and T_j^i have an infinite no. of elements $\forall i \neq 1$ and $j \in N(i)$]

Assumption 2: All distance information is eventually purged from system [ie. for $\bar{t} > t_0$ $\exists$ $t > \bar{t}$, D_j computed before $\bar{t}$ are not received after $\bar{t}$]

Teacher's Signature.....

Bellman Ford or Asynchronous Bellman Ford

$d_{ij} \geq 0$

BF : $D_i = \min_j [d_{ij} + D_j]$ } (#)

$D_1 = 0$

BF iteration : $D_i^{h+1} = \min_j [d_{ij} + D_j^h]$

$D_1^h = 0$

we only discared the nodes when the network has no -ve cycles as well as undirected
For general cases, refer books.

We saw

1. (*) Converges in atmost no. of vertices steps to D_i's that are minimum length
2. ~~(#) has a unique sol^n.~~
2. (#) has unique solution
3. Bellman Ford path $\forall i$ takes i that gives

$$D_i = \min_j [d_{ij} + D_j]$$

4. BF path has no repetitions

$D_i = d_{ij} + D_j$

$= d_{ij} + d_{jj_1} + D_{j_1}$

$D_i = d_{ij} + d_{jj_1} + \cdots + d_{jj_2} + D_i$ (here $d_{jj_1} > 0$, $d_{jj_2} > 0$)

$= 0$

(*)

5. ~~$D_i^{h+1} = \min$~~ From here we observed that

$D_i^{h+1} = \min_j [d_{ij} + D_j^h]$, $D_1^h = 0$ (**)

$D_i^\Delta = c_i$

Teacher's Signature.....

Reference → Gallagard Book

PAGE NO.:
DATE: / /

Converges then it must converge to the correct soln. i and min length paths !!

Asynchronous distributed Bellman Ford

$t_0 < t_1 < t_2 < \cdots < t_m < \cdots$ $\quad m \to \infty$ $\quad t_m \to \infty$

~~(i) Then~~

(i) Things remain same between t_m, t_{m+1}

(ii) Computing happens at t_m's

(iii) Each vertex i, has the values $d_{ij}\ \forall\ j \in N(i)$ and gets information when d_{ij} changes. Also gets D_j from all its neighbors and computes $D_i = \min [d_{ij} + D_j] \quad j \in N(i)$

$D_i(t)$ = Estimate of the distance of j to 1 ~~was~~ available to i at time t.

$D_1(t) = D_i(t)$

$D_i(t) = 0 \quad \forall\ t \geq t_0$

$D_i'(t) = 0 \quad \forall\ t \geq t_0$ and $i \in N(1)$

At time t_m's one of 3 things happen:

1) i updates $D_i(t) = \min_{j \in N(i)} [d_{ij} + D_j(t)]$

and $D_i'(t)$ unchanged

2) Updates D_i after getting ~~changed~~ valued of D_j for some neighbours and bcos D_j''s unchanged for other neighbours.

Teacher's Signature.....

3) i sits idle

T^i = set of times when l occurs

T_j^i = set of times 2 occurs

Assumption: Nodes do not stop sending its D_i's and calculating its D_i's, ie, T^i and T_j^i both are infinite sets.

Assumption: Old information eventually deleted, ie $\forall \bar{t} > t_0$ $\exists$ $\tilde{t} > \bar{t}$ s.t. D_j computed at node j prior to time $\bar{t}$ are not received at any neighbour node i after time $\tilde{t}$.

Fact: $D_i(t)$'s converge to correct distance in finite time.

We did this by constructing $\{\underline{D_i^k}\}$ and $\{\overline{D_i^k}\}$ s.t

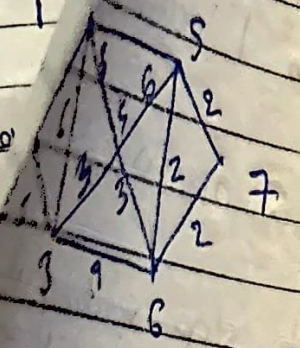
$$\underline{D_i^k} \leq \underline{D_j^{k+1}} \leq D_i \leq \overline{D_i^{k+1}} \leq \overline{D_i^k}$$

Teacher's Signature.....

Converges th
soln. i

Asynchrono

to 2



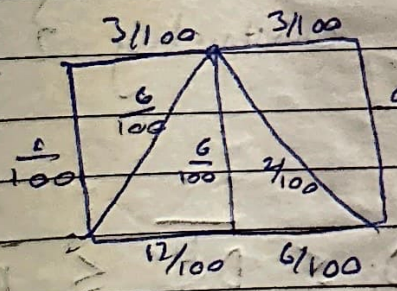
Run BF and Dijkstras algorithm

(i) Th

(i) Thi BF alg on a tree. For each node i+1,
(ii) the node j s.t j gives the minimum h
(iii)

$$D_i^{hi} = \min_j [d_{ij} + D_j^{hi-1}]$$

hi is the largest h s.t $D_i^h \neq D_i^{h-1}$.
Consider the subgraph consisting of all ij and show that this is a spanning tree consisting of shortest paths.



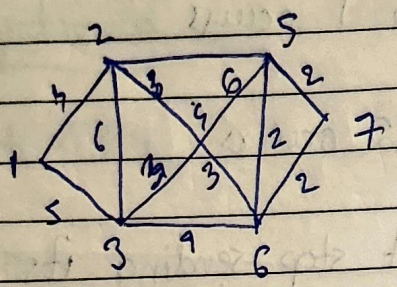
The number shown on each link is the probability of the link failing during the lifetime. It is assumed that links fail independently of each other. Find the most reliable path between each pair of nodes.

Teacher's Signature.....

PAGE NO.:
DATE: / /

Problems

1. Consider



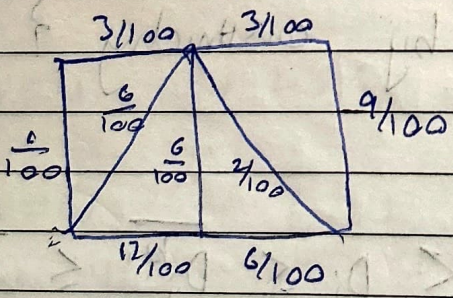
Run BF and Dijkstras algorithm

2. Consider BF alg on a tree. For each node i≠1, take the node j_i s.t j_i gives the minimum in

$$D_i^{hi} = \min_j [d_{ij} + D_j^{h_i - 1}]$$

where h_i is the largest h s.t $D_i^h \neq D_i^{h-1}$. Consider the subgraph consisting of all ij_i and show that this is a spanning tree consisting of shortest paths.

3.

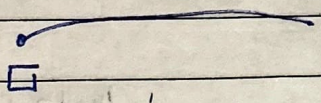


The number shown on each link is the probability of the link failing during the lifetime. It is assumed that links fail independently of each other. Find the most reliable path between each pair of nodes.

Teacher's Signature.....

4. Can you think of a distributed version of Dijkstras algorithm

Crypt Analysis



A = {message}
B = {encrypted message}

$A \xrightarrow{f} B$

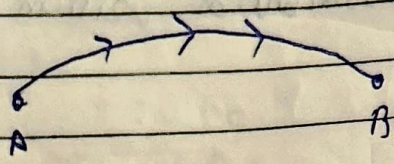
f 1-1 onto

Have to find $\boxed{f^{-1}}$

Teacher's Signature.....

PAGE NO. :
DATE : / /

Network Security



Encryption function f_1
Decryption function f_2

Message set M
Decrypt message set E $|M| = |E|$ and f_1, f_2 are 1-1 onto

$f_1 : M \longrightarrow E$ $\quad f_2 : E \longrightarrow M$

s.t $f_2 \cdot f_1(x) = x \quad \forall x \in M$

RSA Encryption System

Public Key Crypt

We need 3 numbers

d, p, q
p, q primes (very large)

$\gcd(d, (p-1)(q-1)) = 1 \qquad n = pq \qquad M = \{0, 1, \ldots, n-1\}$

We will find ~~d s.t~~ $\mathbb{R}$ s.t

$\mathbb{R}\ d = 1 \bmod \phi(n)$

$\phi \leftarrow$ euler ϕ function

Teacher's Signature.....

PAGE NO.:
DATE: / /

$\phi(n)$: # +ve numbers co-prime to n.

$f_1(x) = x^e \bmod n$

$f_2(y) = y^d \bmod n$

Thm (RSA): $\boxed{x^{ed} \bmod n = x}$ $\forall x \neq p, q$

$\frac{\mathbb{Z}}{pq} \to n$ $\bar{e} \in \mathbb{Z}/pq$ is an inverse if $\gcd(\bar{e}, pq) = 1$

Why the works

$n = pq$

$\phi(n) = \phi(pq) = \phi(p)\phi(q)$
$= (p-1)(q-1)$
$= n - p - q + 1$

$\frac{\mathbb{Z}}{pq} \cong \frac{\mathbb{Z}}{p} \times \frac{\mathbb{Z}}{q}$

As $e \cdot d \equiv 1 \bmod \phi(n)$

$ed = k\phi(n) + 1$

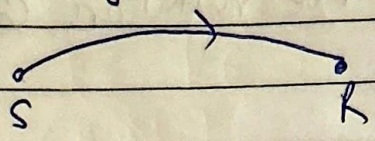
So $x^{ed} = x^{k\phi(n)+1} = x \cdot x^{k\phi(n)}$ $\times y$

$|(\mathbb{Z}/n\mathbb{Z})^\times| = \phi(n)$ $\bar{y} \cdot \bar{a} = \bar{1}$ $ya + nb = 1$

Teacher's Signature.....

PAGE NO.:
DATE: / /

RSA Crypto System



- S sends messages to R
- Every communication is public
- Set of messages M
- R gives S a "key" e and a no. N s.t $M = \{0, \dots, N-1\}$

- By default e and N also become public
- Set of 'encrypted messages' E.
  Here $E = \{0, \dots, N-1\}$

$\exists$ a function $f_e : M \to E$

S → R f_e uses e

S takes a message x, encrypts using e and sends to R.

- R had created e and N in the following way:

R has picked two "very large" primes. p and q $N = pq$

R has chosen a number d s.t
$\gcd(d, (p-1)(q-1)) = 1$

$$\boxed{e = d^{-1} \bmod (p-1)(q-1)}$$

Teacher's Signature.....

~~pq-p-q+1~~

PAGE NO.:
DATE: / /

R has given S only e and N. Has not mentioned anything about p, q and d

This allows R to have a function $f_d : E \to M$ s.t

we expect $\forall x \in M \quad f_d \cdot f_e(x) = x$

$f_e(x) = x^e \bmod N$

$f_d(y) = y^d \bmod N$

It is very hard to find d just from e and N

This is RSA cryptosystem.

Why and How this works?

Euler ϕ function

$\forall$ +ve int n

$\phi(n) = |k|$ $\quad 0 \le k \le n-1$, $\gcd(k,n) = 1$

$\phi(pq) =$  pq-1 - # of terms divisible by either p or q between 1 to pq-1.

$\phi(pq) = (p-1)(q-1)$

$(pq-1) - (q-1+p-1)$

$= pq - p - q + 1$

$= (p-1)(q-1)$

$\mathbb{Z}/pq \simeq \mathbb{Z}/p \times \mathbb{Z}/q$

Teacher's Signature.....

Chinese Remainder Thm

CRT : See from internet

Any n and $k \in \{1, \ldots, n-1\}$ we have

$\gcd(k, n) = 1$

iff

$\bar{k} \in \left(\frac{\mathbb{Z}}{n\mathbb{Z}}\right)^*$

Why?

gcd $\Rightarrow$ unit

$\exists a, b$

s.t $ak + bn = 1$

$\Rightarrow \bar{a} \cdot \bar{k} = 1$ in $\mathbb{Z}/n\mathbb{Z}$

unit $\Rightarrow$ gcd

$\bar{a} \cdot \bar{k} = \bar{1}$

$\Rightarrow ak - 1 = bn$ for some b

$\Rightarrow ak + bn = 1$

$\Rightarrow \nexists$ no $p \neq 1$ that divides k and n for that will imply $p/1$ *

So $\left(\frac{\mathbb{Z}}{pq}\right)^* = \phi(n)$

but $\mathbb{Z}/p$ and $\mathbb{Z}/q$ both fields.

But in fields all non-zero elements have inverses !!

$(a, b) \in F_1 \times F_2$ where F_i's are fields has a inverse iff $a \neq 0_{F_1}$ and $b \neq 0_{F_2}$

Teacher's Signature.....

PAGE NO.:
DATE: / /

So $|F_1 \times F_2|^* = F_1^* \times F_2^*$

Hence

$$|\mathbb{Z}/p \times \mathbb{Z}/q|^* = (\mathbb{Z}/p)^* \times (\mathbb{Z}/q)^* = (p-1)(q-1)$$

$f_e(x) = x^e \bmod N$

$f_d f_e(x) = x^{ed} \bmod N$

Now $ed = 1 \bmod (p-1)(q-1)$

So $ed = 1 + k(p-1)(q-1)$

$= 1 + k\phi(N)$

$x^{ed} = x \cdot x^{k\phi(N)}$

$= x \cdot (x^{\phi(N)})^k$

$x^{ed} \bmod N = x(x^{\phi(N)})^k \bmod N$

as $x \in \{0, \ldots, N-1\}$

If $x = 0$ then obvious

$x^{ed} = x \bmod N$

$x^{\phi(N)} = 1 \bmod N$

If x coprime N, then

$\bar{x} \in (\mathbb{Z}/pq)^*$ and hence $\boxed{\bar{x}^{\phi(n)} = \bar{1}}$

What if x not co-prime to pq?

In CRT, $\frac{R}{IJ} \cong \frac{R}{I} \times \frac{R}{J}$ — (*)

Teacher's Signature.....

$x^{ed} \mod N$

$= x^{1+k\phi(N)} \mod N$

$= x^{1+k(p-1)(q-1)} \mod N$

To show that

$x^{k(p-1)(q-1)} \mod N$

$= 1 \mod N$

If $x \neq mp$

$x^{p-1} = 1 \mod p$

as $\gcd(x, p) = 1$

Hence, $x^{k(p-1)(q-1)} \mod p = 1 \mod p$

by symmetry, $x^{k(p-1)(q-1)} \mod q$

$= 1 \mod q$

when $x \neq nq$

If $x \neq mp, nq$ then

$\bar{x} = \bar{1}$ in $\mathbb{Z}/p$

$\bar{x} = \bar{1}$ in $\mathbb{Z}/q$

Then by (*) $\bar{x}$ must be $\bar{1}$ in $\mathbb{Z}/pq$

If $x = mp \in \{1, \ldots, N-1\}$ then

$(\bar{x}, \bar{x})$ in $\mathbb{Z}/p \times \mathbb{Z}/q$ is $(\bar{0}, \bar{1})$

if $x \in \{0, \ldots, N-1\}$

$x = mp \Rightarrow x \neq mq$

Teacher's Signature.....

So $\gcd(x, q) = 1$

So $\bar{x} = \bar{1} \bmod q$

Also as $x = mp$ $\quad x = 0 \bmod p$

But $(\bar{0}, \bar{1})^* = (\bar{1}, \bar{0})$. Hence $h = x \bmod N$

This forms the proof of RSA

- Note: that direct check to compute $\phi(N)$ takes $O(\sqrt{N})$
- If you know $\phi(N)$ and the fact $N = pq$ for 2 primes, then you know p and q

$$\phi(N) = N - p - q + 1$$

So both pq and $p+q$ will be known and hence p and q.

$17^{13} \bmod 19$

$= 17 \cdot 14^6 \bmod 19$

$= 17 \cdot 6^3 \bmod 19$

$= 17 \cdot 216 \bmod 19$

$= 17 \cdot 7 \bmod 19$

$=$

$17 \cdot (289)^6 \bmod 19$

$289 \bmod 19 = 14$

$14^2 = 196 \bmod 19$

$= 6$

Find inverse of $17 \bmod 30$

$2 \cdot 17 - 30 = 4$

$(4 \cdot 8 - 30 = 2) \times 10$

$10(4 \cdot 8 - 30) = 20$

$3 \cdot 17 - 30 = 21$

$-10(10(2 \cdot 17 - 30) \cdot 8 - 30$

$+ 3 \cdot 17 - 30 = 1$

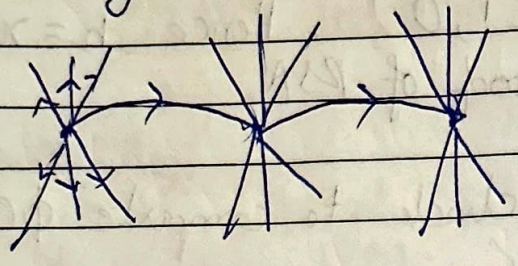
Teacher's Signature.....

PAGE NO.:
DATE: / /

Extended Euclidean Algo

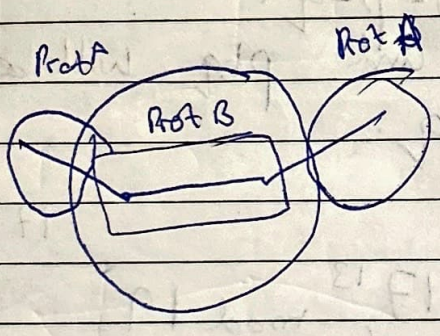
$-153 \cdot 17 = 1 \bmod 30$
$\Rightarrow -7 \cdot 17 = 1 \bmod 30$
$\Rightarrow 23 \cdot 17 = 1 \bmod 30$

Internetworking



Different networks use different protocols.
Eg. Vodafone in India calling AT&T in California.

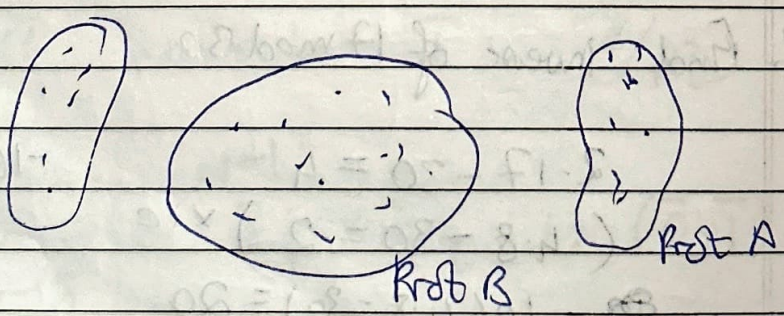
How to communicate??



Tunneling

- See in Internet

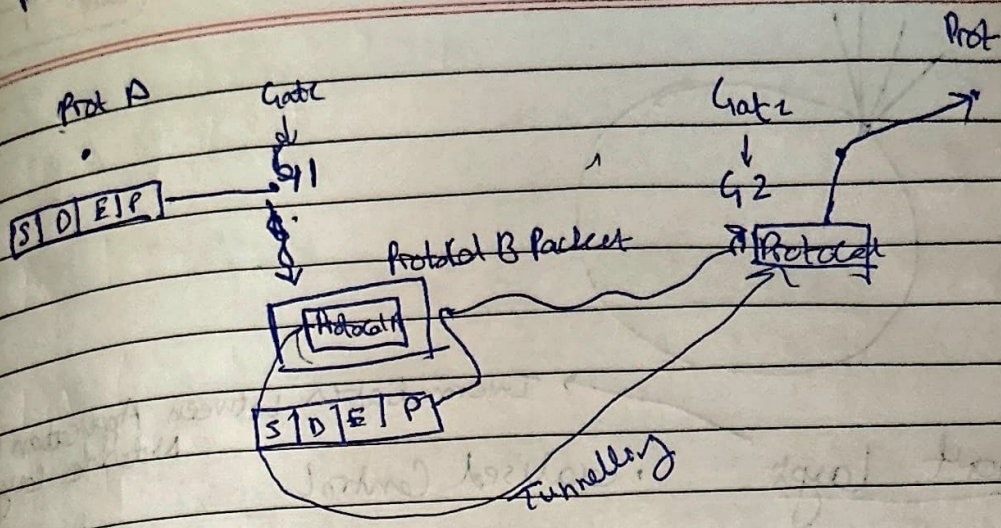
For both the topics, Tanenbaum



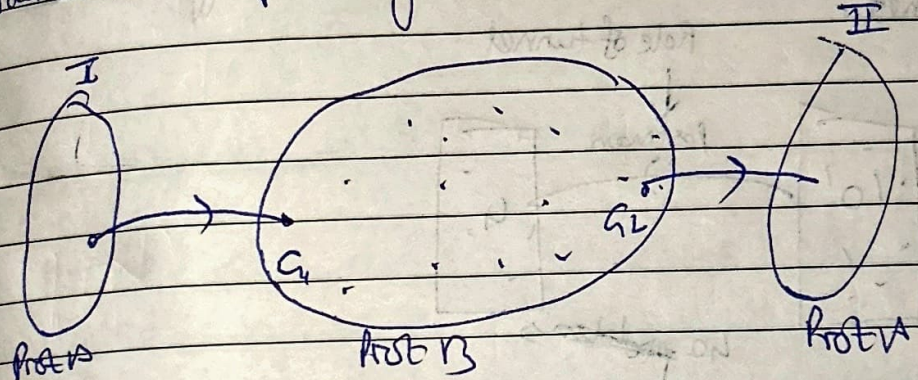
Teacher's Signature.....

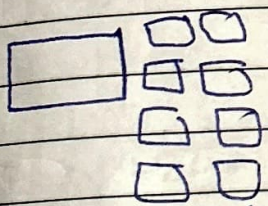
S - Source
D - Dest
E - Error Ctrl
P - Protocol

PAGE NO. :
DATE : / /



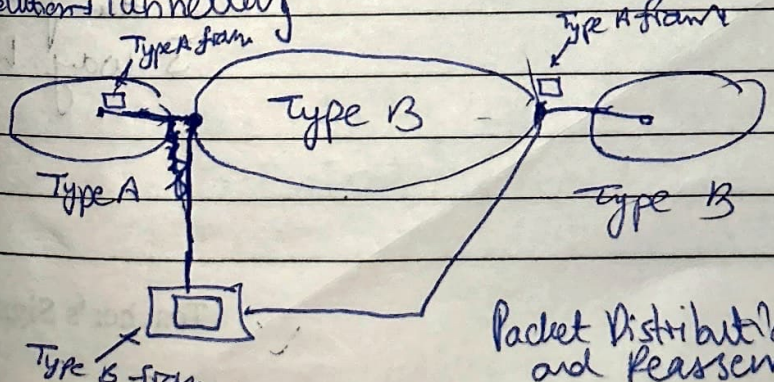
- Packet ~~Error~~ Splitting





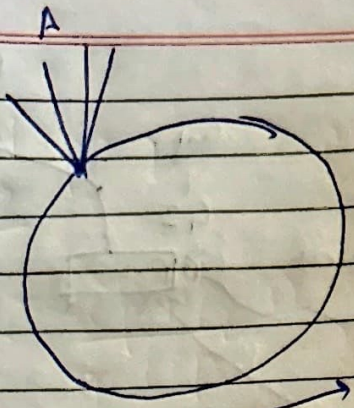
↑
Bit used for ordering or ordering

Problem
1. Internetworking
Solution: Tunnelling



Packet Distribution and Reassembling

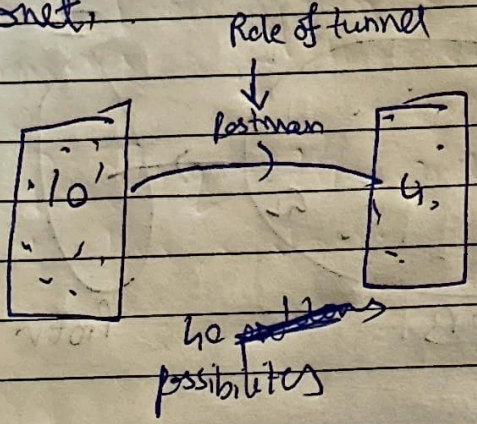
Teacher's Signature.....

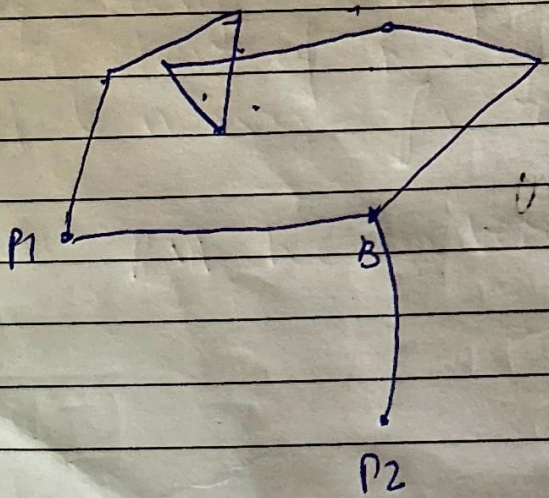


→ Intermediates between Application Layer and Network Layer.

Transport layer , Localised Control.

- Deals with communication b/w different networks of internet.





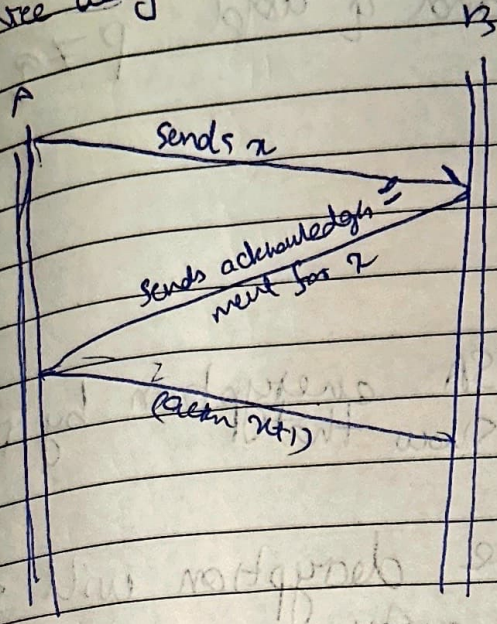
Delay and Repeat Problem

↑

Solution to this problem is 3-way handshake

Teacher's Signature.....

Three way Handshake



Some problems on RSA

1. Let $N = 221$ and $e = 77$. What will be the encrypted message if the original message is 27. If the encrypted message is 95 Then what is the original message.

2. Let $N = 224,551$. You may assume N has no prime ~~factor~~ factor ≤ 59 (may cross check by hand). You may also assume $\sqrt{N} = 473.85$ $\sqrt[3]{N} = 60.78$

Prove that if N is not a prime the N must be a product of two primes p_1 and p_2, $p_1 < p_2$ with $61 \leq p_1 \leq 467$

Suppose one tells you that $0 \leq p_1 \leq 450$. Find the factorisation of N ?

Teacher's Signature.....

PAGE NO.:
DATE: / /

3. For $N = p * q$, p and q odd $p \neq q$, define

$$\lambda(N) = \frac{(p-1)(q-1)}{\gcd(p-1, q-1)}$$

Suppose we verify RSA encryption by $ed \equiv 1 \bmod \lambda(N)$. Show that

(1) Prove that encryption e decryption will still be inverses in the system

(2) If $p = 37, q = 79$ and $e = 7$, then compute d in this system.

4. $N = pq$ encryption key e and decryption key Find the number of messages x s.t the encrypted message for x is also x.

5. Show that RSA becomes very insecure if $p = q$

https://crypto.stackexchange.com/questions/53105/why-do-p-and-q-have-to-be-different-in-rsa

Teacher's Signature.....