- **Bellman Ford vs Asynchronous Bellman Ford :-**

$d_{ij} \geq 0$ and if i, j are not connected then $d_{ij} = \infty$.

BF says : $D_i = \min_{j \in N(i)} [d_{ij} + D_j]$ ——— (#)

$D_1 = 0$

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

$D_1^h = 0$

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

We only discussed the probs when the network has no cycle as well as undirected.

[For general case, reference : 1) any graph algo book
2) Data network
— R. Gallager et. al. books]

We saw:

1) (*) converges in atmost number of vertices steps to D_i's that are minimum length paths from i to 1.

2) (#) has a unique soln.

3) Bellman ford path: $\forall i$ take j that gives min. in $D_i = \min_j [d_{ij} + D_j]$

4) BF path has no repetation

$$D_i = d_{ij} + D_j$$

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

$$\vdots$$

$$= \underbrace{d_{ij} + d_{jj_1} + \cdots + d_{j_{k-1}j_k}}_{0} + D_i$$

5) From here we observed that if

$$D_i^{h+1} = \min_j \left[d_{ij} + D_j^h \right] ; \quad D_1^h = 0$$

$$D_i^0 = c_i$$

converges then it must converge to the correct solutions i.e. min. length paths.

- ADBF (Asynchronous distributed Bellman-Ford):-

$$t_0 < t_1 < t_2 < \cdots < t_m < \cdots$$

as $m \to \infty$, $t_m \to 0$

(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 neighbours & computes $D_i = \min_{j \in N(i)} [d_{ij} + D_j]$

$D_j^i(t)$ = Estimate of the shortest distance of $j \in N(i)$ j to 1 available to i at time t.

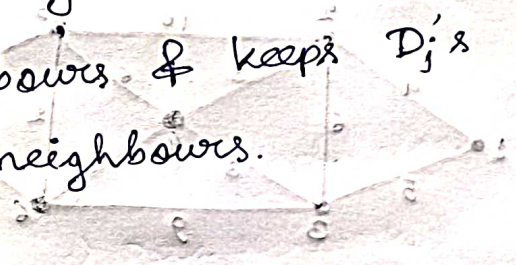
$D_i(t) = D_i^i(t)$ Estimate of min. dist. from i to 1 computed via BF iteration at time t

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

$D_1^i(t) = 0 \quad \forall t \geq t_0 \;\&\; i \in N(1)$

At time t_n's one of 3 things happen:

1. i updates $D_i(t) = \min_{j \in N(i)} [d_{ij} + D_j]$ & keeps $D_j^i(t)$ unchanged.
2. Updates D_j^i after getting values of of D_j ; for some neighbours & keeps D_j^is unchanged for other neighbours.



3. i sets idle

T^i = set of times when 1 occurs

T_j^i = set of times 2 occurs

- Assumption 1 : Nodes do not stop sending its D_i^is & calculating its D_i^is i.e. T^i & T_j^i both are infinite sets.
- Assumption : Old informations eventually deleted i.e. $\forall \bar{t} \geq t_0$, $\exists\, \tilde{t} > \bar{t}$ s.t. D_{ij} computed at node j prior to time $\bar{t}$ are not received at any neighbour node i after time $\tilde{t}$.

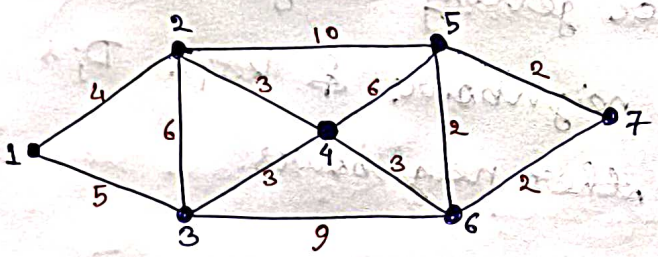
- **Fact:** $D_i(t)$'s converge to correct distance in finite time.

We did this by constructing $\{\underline{D_i^k}\}$ & $\{\overline{D_i^k}\}$ s.t. $\underline{D_i^k} \le \underline{D_i^{k+1}} \le D_i \le \overline{D_i^{k+1}} \le \overline{D_i^k}$

(Proved in last class)

Problems:-

1) Consider



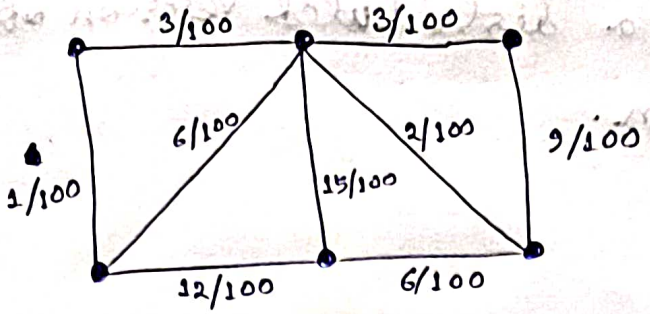
Run BF and Dijkstra algorithms.

2) Consider BF algo on a tree. For each node $i \neq 1$, take the node j_i s.t. j_i gives the minimum in $D_i^{h_i} = \min_j \left[d_{ij} + D_j^{h_i - 1} \right]$

where h_i is the largest h s.t. $D_i^h \neq D_i^{h-1}$.

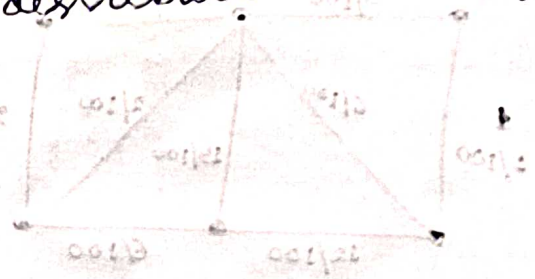
Consider the subgraph consisting of all $i j_i$ (edge) & show that this is a spanning tree consisting of shortest paths.

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.

4) Can you think of a distributed version of Dijkstra's algorithm.



Network Security :-

Crypt Analysis

A = {messages}
B = {encrypted messages}
$A \xrightarrow{f} B$
f is 1-1, onto

How to find f^{-1}?

Encryption function f_1

Decryption function f_2

Message set M

Encrypted message set E.

$f_1 : M \to E$, $f_2 : E \to M$

$|M| = |E|$

f_1, f_2 are 1-1 and onto,

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

Modular arithmetic

• RSA Encryption System :

(Public key cryptography)

RSA
↓
Rivest, Shamir, Adleman.

d, p, q

p, q → primes (very large)

d → $\gcd(d, (p-1)(q-1)) = 1$

$n = pq$, $M = \{0, \ldots, n-1\} = \mathbb{Z}_n$

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

We will find e s.t. $e \cdot d \equiv 1 \mod \phi(n)$

ϕ ← Euler ϕ function

$\phi(n)$ = # (+ve) numbers less than n and co-prime to n

- $f_1(x) = x^e \bmod n$
- $f_2(y) = y^d \bmod n$

- Theorem (RSA):-

$$x^{ed} \bmod n = x \quad ; \forall x \neq p, q$$

- I will give you e, n to find d. To find d we have to know p, q.

- $\frac{\mathbb{Z}}{(pq)} \to n$

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

- Why this works?

$$n = pq$$

$$\phi(n) = \phi(pq)$$

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

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

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

- As $ed \equiv 1 \bmod \phi(n)$,

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

$\Rightarrow x^{ed} = x^{k\phi(n)+1}$

So, $x^{ed} = x \cdot x^{k\phi(n)} \bmod n$

$\left(\frac{\mathbb{Z}}{n\mathbb{Z}}\right)^* \to$ set of units

Read Chinese Remainder Theorem, extended Euclidean Algo.

Chinese Remainder Theorem

CRT is used to solve a set of different congruent eqn's with one variable but different moduli which are relatively prime as shown below:

$X \equiv a_1 \pmod{m_1}$

$X \equiv a_2 \pmod{m_2}$

$\vdots$

$X \equiv a_n \pmod{m_n}$

(CRT states that the above eqn's have a unique soln if the moduli are relatively prime)

$X = (a_1 M_1 M_1^{-1} + \cdots + a_n M_n M_n^{-1}) \bmod M$

where $M = m_1 \times m_2 \times \cdots \times m_n$,

$M_i = \frac{M}{m_i}$, $M_i \times M_i^{-1} = 1 \bmod m_i$

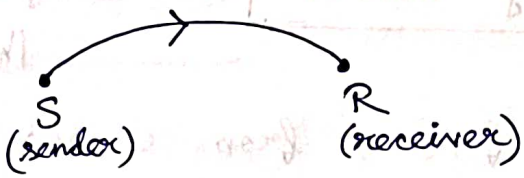
$|(\mathbb{Z}/n\mathbb{Z})^*|$

$\phi(n)$ $\hookrightarrow$ $\bar{y}$ belongs $\in (\mathbb{Z}/n\mathbb{Z})^*$

$x^{k\phi(n)}$ ⟹ $\bar{y}\cdot\bar{a} = \bar{1}$

$ya + nb = 1$

- **RSA crypto system :-**

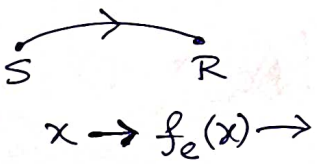


- S sends message to R.
- Every communication is public.
- Set of message M.
- R gives S a 'key' e and a no. n s.t.

$$M = \{0, n-1\}$$

- By default e & n also public.
- Set of 'encrypted messages' E. Hence $E = \{0, 1, \ldots, n-1\}$

$\exists$ a function $\boxed{f_e : M \to E}$



f_e uses e

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

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

R has picked two 'very large' primes p & q,

$n = pq$.

R has chosen a number d s.t. $\gcd(d, \underbrace{(p-1)(q-1)}_{\phi(n)}) = 1$

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

R (Receiver) gives S only e and N. Has not mentioned anything about p, q & d.

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

we expect $\underline{\forall x \in M}$, $\boxed{f_d \cdot f_e(x) = x}$.

$$\begin{cases} \boxed{f_e(x) = x^e \bmod N} & \text{(For encryption)} \\ \boxed{f_d(y) = y^d \bmod N} & \text{(For decryption)} \end{cases}$$

This is RSA cryptosystem

[⊛ It is very hard to find d from e and N.]

- Why & How this works??

- Euler-Phi-function :-

$\forall$ (+ve) integer n,

$\phi(n) = \{k : 1 \le k \le n-1, \gcd(k,n) = 1\}$

↳ # integers less than n & coprime to n.

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

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

$= pq - 1 - p - q + 2$

i.e. $\boxed{\phi(pq) = (p-1)(q-1)}$

$\therefore \mathbb{Z}/pq \cong \mathbb{Z}/p \times \mathbb{Z}/q$

(≅ : Ring)

Any n and $k \in \{1, \ldots, n-1\}$ we have $\gcd(k,n) = 1$ iff $\bar{k} \in (\mathbb{Z}/n\mathbb{Z})^*$

↳ set of units of $\mathbb{Z}/n\mathbb{Z}$

Why??

gcd = 1 ⇒ unit = ∃ a, b s.t. $ak + bn = 1$
$\Rightarrow \bar{a}\bar{k} + \bar{b}\bar{n} = \bar{1}$ [$\because \bar{bn} = \bar{0}$ in $\mathbb{Z}/n\mathbb{Z}$]
$\Rightarrow \bar{a} \cdot \bar{k} = \bar{1}$ in $\mathbb{Z}/n\mathbb{Z}$

∎

~~gcd ⇒ unit~~

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

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

$\Rightarrow ak + bn = 1$

$\Rightarrow \exists$ no $p \neq 1$ that divides k & n for that will imply $p \mid 1$.

So, $\left|(\mathbb{Z}/p\mathbb{Z})^*\right| = \phi(n)$

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

[But in fields ~~has~~ all non-zero elements have inverse

$(a, b) \in F_1 \times F_2$; where F_i are fields,

~~has a inverse~~ iff $a \neq 0_{F_1}$ & $b \neq 0_{F_2}$.

So, $(F_1 \times F_2)^* = F_1^* \times F_2^*$.]

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

Now, $f_e(x) = x^e \bmod n$

$\therefore f_d \cdot f_e(x) = x^{ed} \bmod n$

Now, $ed \equiv 1 \bmod (p-1)(q-1)$

So, $ed = 1 + k(p-1)(q-1)$
$= 1 + k \cdot \phi(n)$

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

$\therefore x^{ed} \bmod N = x \cdot (x^{\phi(N)})^k \bmod N$ as $x \in \{0, 1, \ldots, N-1\}$

If $x = 0$ then obviously $x^{ed} \equiv x \bmod N$

We want to show, $x^{\phi(N)} \equiv 1 \bmod N$,

if x is coprime to N then

$\bar{x} \in (\mathbb{Z}/pq)^*$ and $\bar{x}^{\phi(N)} \equiv \bar{1}$

↳ i.e. $\bar{x}$ is a unit

What about x not coprime to N.

In CRT

$$\frac{R}{I \cap J} \cong \frac{R}{I} \times \frac{R}{J}$$ the isomorphism $\bar{x} \mapsto (\bar{x}, \bar{x})$

└── ⊛

$I + J = (1)$

$\Rightarrow I \cap J = IJ$

$x^{ed} \bmod N$

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

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

To show that $x^{k(p-1)(q-1)} \bmod N = 1 \bmod N$

Take ⊛,

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

$= (x^{p-1})^{k(q-1)} \bmod p$

If $x \neq pm$

$x^{p-1} = 1 \bmod p$ as $\gcd(x, p) = 1$

$\therefore x^{k(p-1)(q-1)} \bmod p = 1 \bmod p$

By symmetry $x^{k(p-1)(q-1)} \bmod q = 1 \bmod q$ when $x \neq q^n$

If $x \neq mp, m'q$ then

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

So by (*), must $\bar{x}$ 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 $\bar{x} \in \{0, \ldots, N-1\}$, $x = mp$, $x \neq m'q$:

so $\gcd(x, q) = 1$ so $\bar{x} = \bar{1}$ mod q

Also $x = mp \Rightarrow \bar{x} = 0$ mod p)

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

— This proves the RSA

- Note that direct check to compute $\phi(N)$ take $\phi(\sqrt{N})$
- If you know $\phi(N)$ & the fact that $N = pq$ for 2 primes then you know p & q as $\phi(N) = N - p - q + 1$.
  $= (p-1)(q-1)$ [$\because N = pq$]
- So both pq and $p+q$ will be known & hence p & q.

$17^{13} \bmod 19$

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

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

$= 17 \cdot 7 \bmod 19$

$17^3 = 17 \cdot 17^{12}$
$= 17 \cdot (17^2)^6$

$19\overline{)289}(15$
19
99
95
4

Division algorithm

$30 = 17 \times 1 + 13$

$17 = 13 \times 1 + 4$

$13 = 4 \times 3 + 1$

$13 - 4 \times 3 = 1$

$13 - (17 - 13) \times 3 = 1$

$13 \times 4 - 17 \times 3 = 1$

$(30 - 17 \times 1) \cdot 4 - 17 \times 3 = 1$

$30 \times 4 - 17 \times 7 = 1$

9) Find inverse of 17 mod 30?

$2 \cdot 17 - 30 = 4$

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

$3 \cdot 17 - 30 = 21$

$30 \times 4 + 17(-7) = 1$

$17 \times (-7) = 1$

$-7 = \overline{23}$

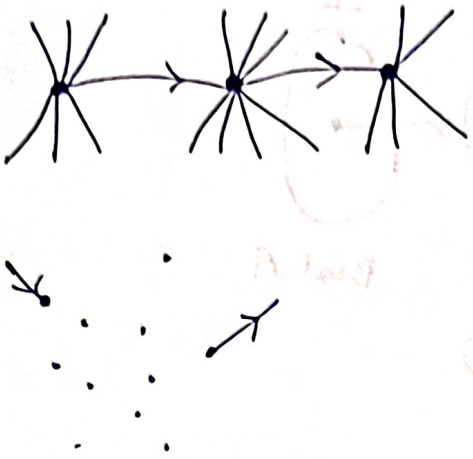
$$-10\left((2\cdot 17-30)\cdot 8-30\right)+3\cdot 17-30=1$$

$$\Rightarrow (-157\cdot)17 = 1 \mod 30$$

$$\Rightarrow -7\cdot 17 = 1 \mod 30$$

$$\Rightarrow 23\cdot 17 = 1 \mod 30.$$

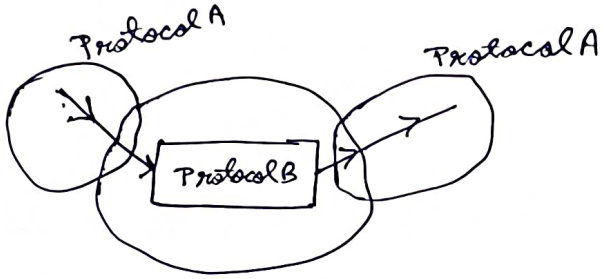
Internetworking :-



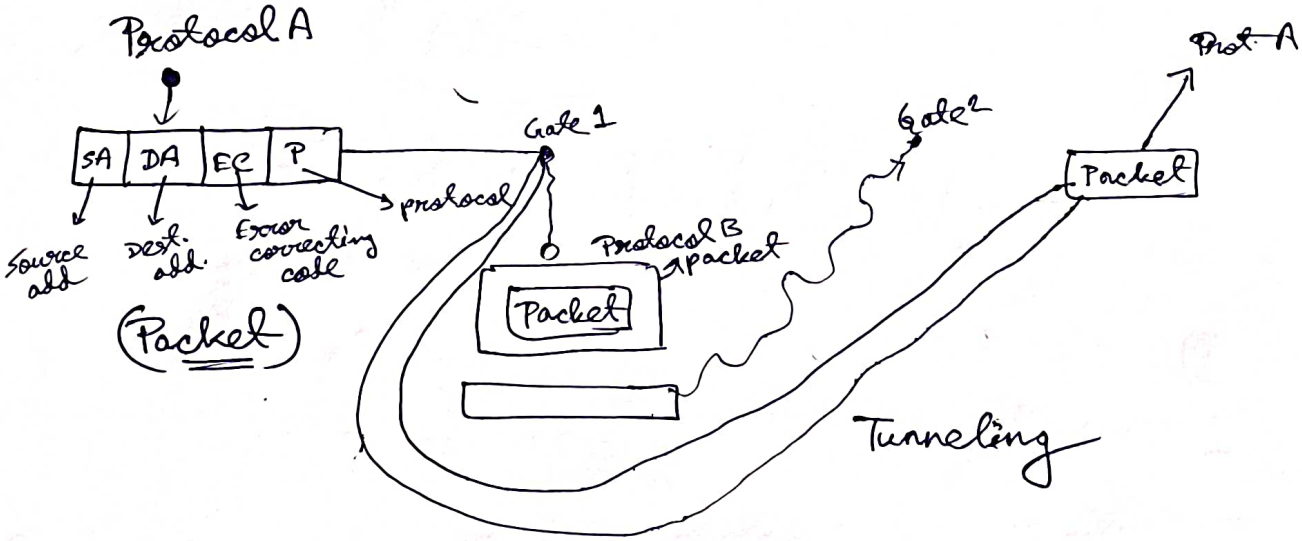
Different networks use different ~~protocols~~ protocols.

How to communicate?

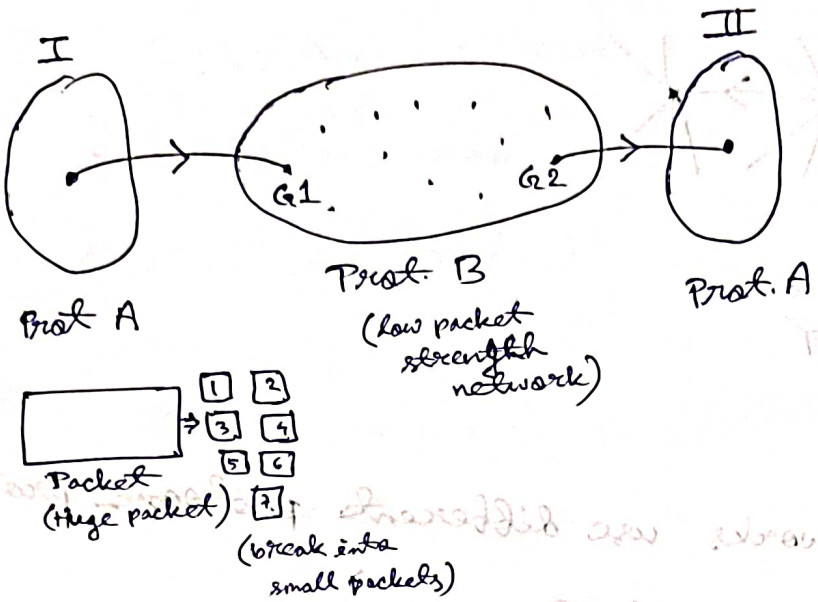
- **Tunneling :-**



Ref:-
Tanenbaum book



- Packet ~~Issues~~ ⇒ Spliting :-



(Tannenbaum)