classmate
Date 3.9.2020
Page 1

ROUTNG

make evil go away.

verse 1

there is a world
that is virtual and different
it can be so cold
makes us stand up for what's right
our hope through our life
if we reset it to the start

chorus

here we are, going far
to save all that we love
if we give, all we've got
we will make it through
here we are, like a star
shining bright on your world
today (make evil go away)

mini-chorus

code lyoko we'll, reset it all
code lyoko be, there when you call
code lyoko we, will stand real tall
code lyoko stronger after all

verse 2

a world of machines
it can shadow human nature
all that we need
is the way to find the answer
and one thing is for sure
you can count on us for good.

chorus ...

verse 3

we'll do our best
to never let you down
we're up to the test
to turn this world around

mini-chorus ...

mini-chorus ...

by Noam Kaniel (vocals)

written by
Ygal Amar
Franck Keller

A WORLD WITHOUT DANGER
CODE LYOKO (June 6, 2004)

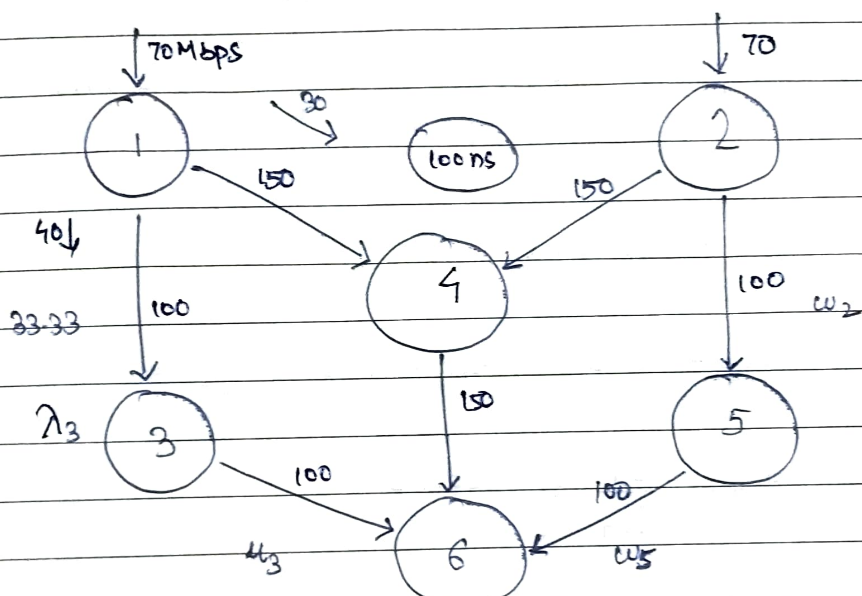
RECAP

M / M / m
M / M / m / m
M / G / 1
M / D / 1

} queuing system

M / M / 1

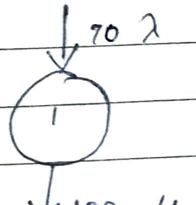
L, W, W_s, L_q, W_q, L_s tabular summary



routing 1 1 - 3 - 6 & 2 - 5 - 6

2 1 - 4 - 6 & 2 - 4 - 6

routing 1



$\rho = \frac{70}{100} = 0.7$

w_1 : delay from 1 → 3

$w_1 = \frac{L_1}{\lambda_1} = \frac{\rho_1}{\lambda_1(1-\rho_1)} = \frac{7}{3\times 70\times 10^6} = \frac{10^{-6}}{30} = \frac{100}{3}$ ns $= 33.3$ ns

$\lambda_3 = 70 \times 10^6$ $\rho_3 =$ $w_3 = 33.33$ ns.

$w_2 = w_5 = 33.33$ ns

<u>routing 2</u>

1 - 4 - 6 2 - 4 - 6

$\lambda_1 = 70$ $\mu_1 = 150$ $\rho_1 = \frac{70}{150}$

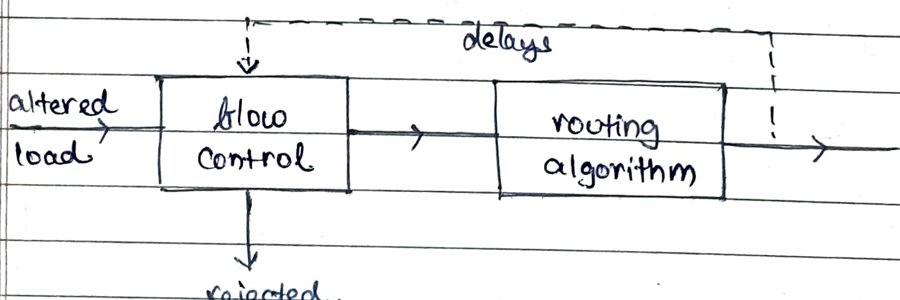
1 - 4 $w_1 = \frac{L_1}{\lambda_1} = \frac{70}{70\,(80) \times 10^6} = 12.5$ ns

2 - 4 $w_2 = 12.5$ ns

4 - 6 $\lambda_4 = 140$ $\mu_4 = 150$ $\rho_4 = \frac{140}{150}$

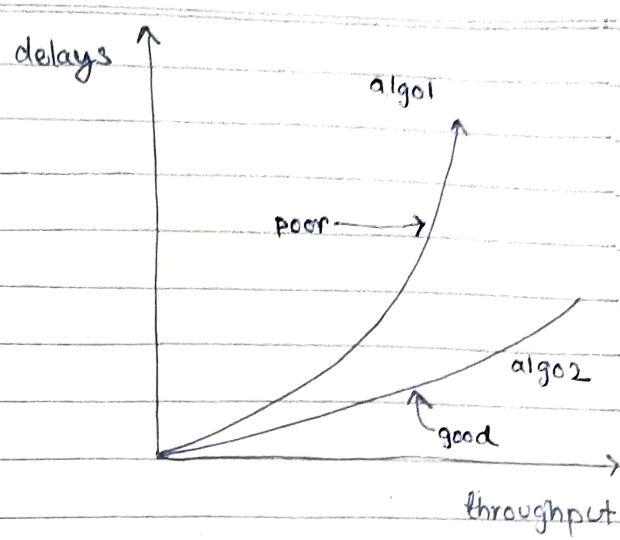
$w_4 = \frac{L_4}{\lambda_4} = 100$ ns

net delay $w_1 + w_4$ or $w_2 + w_4$ = <u>112.5 ns</u>



<u>throughput = offered load - rejected load</u>

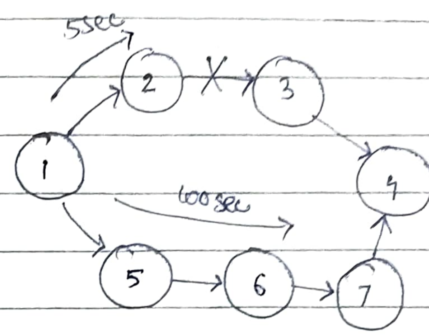
ROUTING



ensure delays are minimized
maximize throughputs

routing → Protocols
routing → algorithms

(i) static vs. dynamic
(ii) flat vs. hierarchical
(iii) ~~flat~~ intra-domain vs. inter-domain
(iv) distributed vs. centralized
(v) single vs. multipath
(vi) link state vs. distance vector



link state: need to know whole network (dijkstra)

(hierarchical routing)

distance vector: bellman-ford

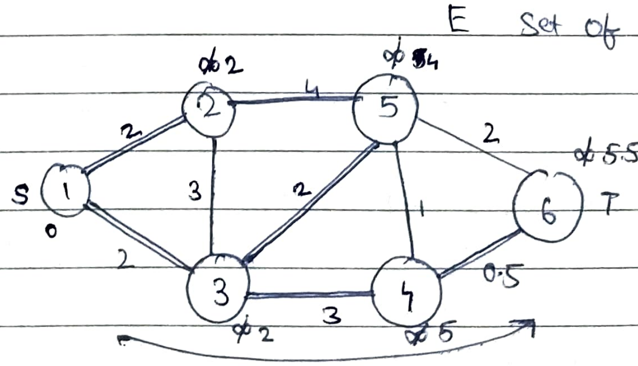
ROUTNG ALGORITHMS (shortest-path algorithms)

DIJKSTRA

$G = (V, E)$ V set of vertices
E set of edges

(i) dijkstra
(ii) bellman-ford
(iii) floyd-warshall

$S = \{1\}$



every step we add 1 node to our shortest path tree

$S = \{1\}$

initialization $D_1 = 0$ $D_j = d_{ji}$ $\forall j \neq 1$

$P = \{\}$

ALGORITHM

(1) $D_i = \min_{j \notin P} D_j$ $i \notin P$

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

(2) $D_j = \min\{D_j, D_i + d_{ij}\}$ $\forall j \notin P$

2 + 3

goto (1) until $P = G$

initialization $D_1 = 0$ $D_2 = 2$ $D_3 = 2$ $D_4 = \infty$ $D_5 = \infty$ $D_6 = \infty$

$P = \{1\}$

(1). $P = \{1, 2\}$

$D_2 = 2$, $D_3 = 2$ $D_4 = \infty$ $D_5 = 6$ $D_6 = \infty$

(2). $P = \{1, 2, 3\}$

$D_2 = 2$ $D_3 = 2$ $\underline{D_4 = 5}$ $\underline{D_5 = 4}$ $D_6 = \infty$

(3) $P = \{1, 2, 3, 5\}$

$D_2 = 2$ $D_3 = 2$ $D_5 = 4$ $D_4 = 5$ $D_6 = 5.5$

(4) $P = \{1, 2, 3, 4, 5, 6\}$

done.

<u>running time</u> $|V| = N$ # iterations = $N-1$

$O(N)$ in each iteration

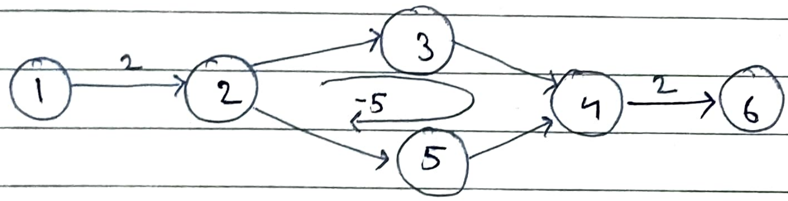
$O(N^2) \to O(N \log N)$ $O(|E| + |V| \log |V|)$

dijkstra <u>disadvantage</u>

(i) -ve edges are not allowed.

BELLMAN-FORD

-ve edges are allowed
but, no -ve wt. cycles. ← shortest path is not defined



<u>init.</u> ~~[illegible]~~ $D_1^n = 0$ $n = 1..n$

$D_i^0 = \infty$ $\forall i = 2..n$ $i \neq 1$

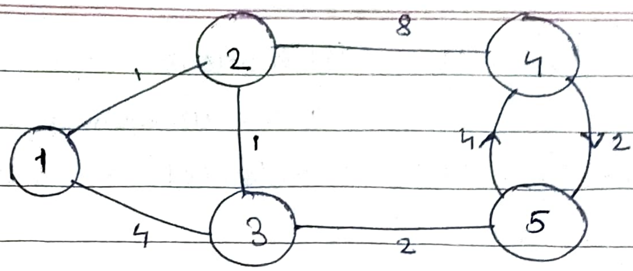
algorithm. $D_i^{n+1} = \min \{ d_{ij} + D_j^n \}$ $\forall i \neq 1$

terminate if $D_i^n = D_i^{n-1}$ $\forall i$

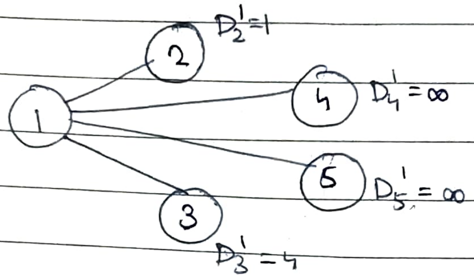
(D_j^n min. cost of reaching j using atmost n edges)

$D_i^{n+1} \leq \min_{j \neq 1} \{ d_j + D_i^n , D_i^n \}$

every step, shortest path with n edges to each node is calculated.



$D_1^1 = 0 \quad D_2^0 = D_3^0 = D_4^0 = D_5^0 = \infty$



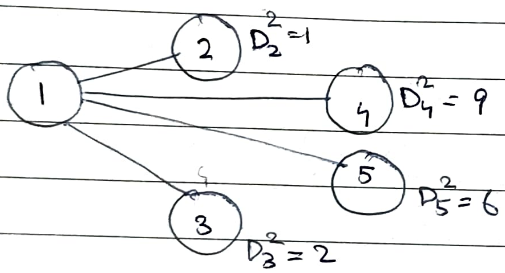
running time

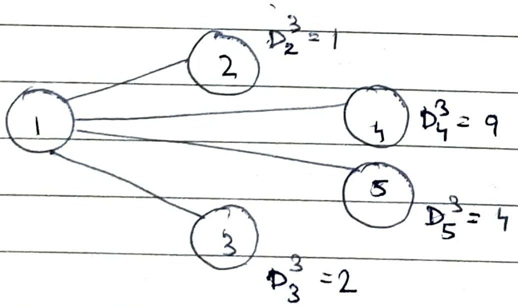
$O(|V||E|) \quad O(N^3)$

$|V| = N$

$O(m|E|)$

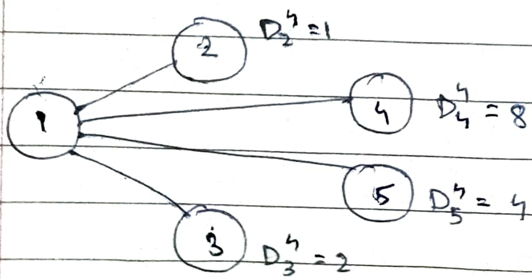
iteration required.





$m << N$

$|E| << N^2$



PROPOSITION

(a) D_i^h's ~~are~~ generated by the algorithm are equal to the shortest path from i to 1 of # edges $\leq h$.

(b) the algorithm terminates iff all cycles not containing node 1 are having non--ve costs. if algo terminates, it so in $h \leq N$ & at termination, D_i^h is the cost of shortest path from 1 to i (or i to 1).

HW $K \leq h$ prove $h+1$

$D_i^{K+1} \leq D_i^K \quad \forall K \leq h.$

FLOYD - WARSHALL
(all-pair shortest path)

$D_{ij}^0 = d_{ij} \quad \forall i, j \qquad$ for $h = 0 \ldots N-1$

$D_{ij}^{h+1} = \min_K \left[D_{ij}^h, \; D_{ik}^h + D_{kj}^h \right]$

iteration - h:

we have the shortest path cost from i to j using nodes $1, 2, \ldots h$ ($h = 0$, w/o any node)

$O(N^3) \quad O(|V|^3)$