Name: Alvi Akbar
Student No: 11118887
NSID: ala273

(1)

Assignment #02

(1)

Generator 11001

i.e $x^4 + x^3 + 1$ → degree (4)

```
11001 | 10111010000
        11001
        ---------
        011100
         11001
        ---------
        0010110
          11001
        ---------
          011110
           11001
        ---------
           0011100
             11001
        ---------
             00101   → REM
```

⇒ 10111010101 ← Actual Frame Transmitted

---

(2) (i)

$$\tau = \frac{32}{3 \times 10^8 \text{ m/s}} = 0.10667 \times 10^{-6} \text{ s}$$

$$t = 0.10667 \times 10^{-6} \text{ s} + \frac{128 \times 8 \text{ bits}}{100 \times 10^6 \text{ bits/s}}$$

$$= 10.3466 \times 10^{-6} \text{ s} = 10.34 \, \mu\text{s}$$

$$\frac{\tau}{t} = \frac{0.10667 \times 10^{-6} \text{ s}}{10.3466 \times 10^{-6} \text{ s}} = 0.0103$$

CSMA seems more reasonable

(b) $\tau = 32 \times 10^3 \text{ m} / (3 \times 10^6 \text{ m/s}) = 1.067 \times 10^{-5} \text{ s}$

$$t = 1.067 \times 10^{-5} \text{ s} + \frac{128 \times 8 \text{ b}}{100 \times 10^6 \text{ b/s}}$$

$$= 2.091 \times 10^{-5} \text{ s} = \underline{20.91 \text{ ms}}$$

∴ ALOHA is more reasonable

(c) $$\tau = \frac{32 \text{ m}}{3 \times 10^8 \text{ m/s}} = 0.10667 \times 10^{-6} \text{ s}$$

$$t = 0.10667 \times 10^{-6} \text{ s} + \frac{64 \times 8 \text{ b}}{5 \times 10^7 \text{ b/s}}$$

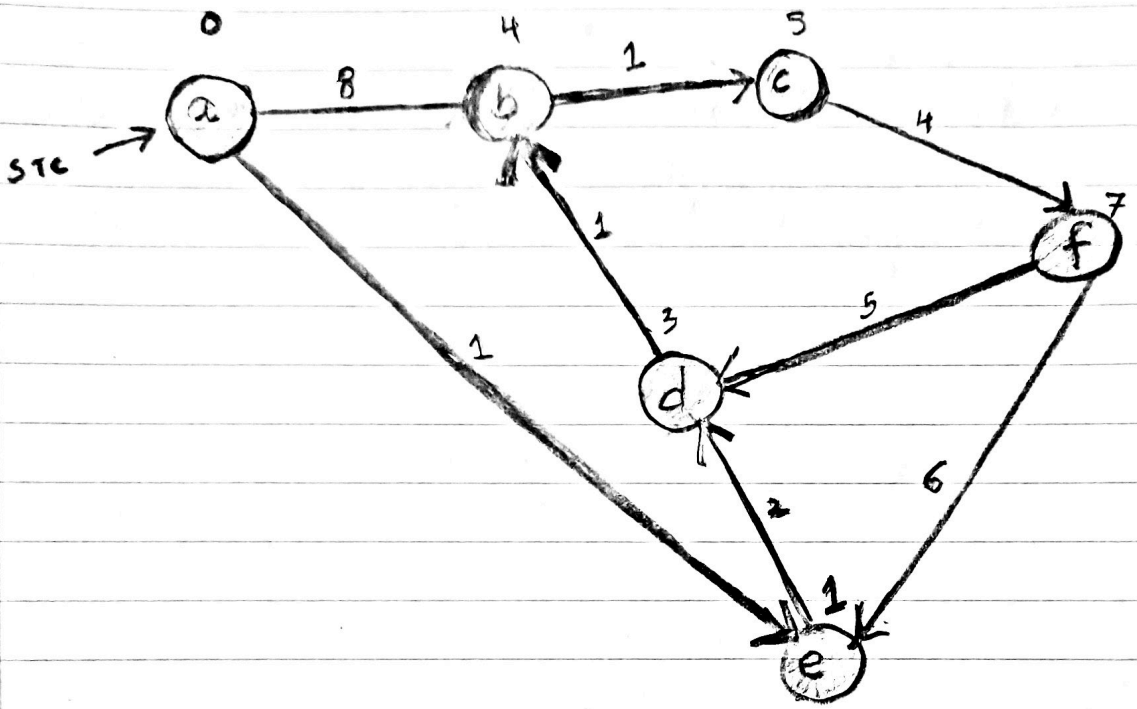
$$= 2.09 \times 10^{-7} \text{ s}$$

$\tau / t = 0.52$ ALOHA is more appropriate.

(3) Dijkstra's Algorithm

Algorithm

- Extract Node N with lowest distance
- Add Node to N to shortest path tree
- Relax the distance of neighbours of N by lowering any better estimate



Initialization	N	D(b)	D(c)	D(d)	D(e)	D(f)
Initialization	{a}	8	∞	∞	1	∞
after iteration 1	{a,e}	8	∞	3	1	7
after iteration 2	{a,e,d}	4	∞	3	1	7
after iteration 3	{a,e,d,b}	4	5	3	1	7
after iteration 4	{a,e,d,b,c}	4	5	3	1	7
after iteration 5	{a,e,d,b,c,f}	4	5	3	1	7
after iteration 6	{a,e,d,b,c,f,e}	4	5	3	1	7

(3)

Distance Vector Algorithm

(b) **Algorithm**

- Initialize vector with 0 (zero) cost to self, ∞ (infinite) to other destinations
- Periodically send vector to neighbours
- Update vector for each destination by selecting the shortest distance heard, after adding cost of neighbour link
  - Use the best neighbour for forwarding.