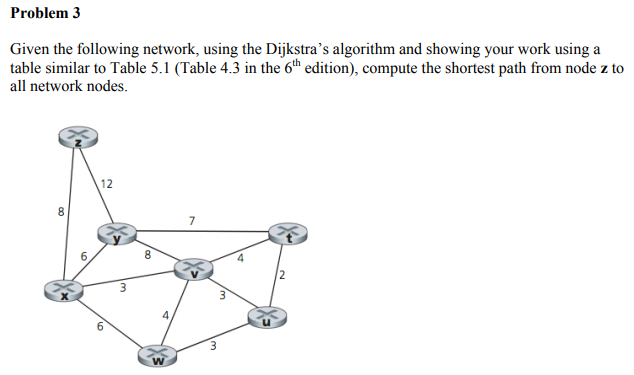


1. R = 11010 to get this is did the long division of 1110001100000/110011
2. Frame being transmitted is D + R = 1110001100000+11010 = 1110001111010
3. The receiver knows if the frame is correct or not by doing the long division again with G and the D+R value and if the remainder is 0 then there are no errors

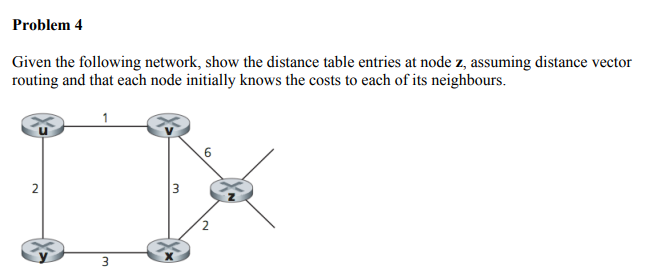


1. R = 110
2. D+R = 111001100000 + 110 = 111001100110
3. Same thing if we check the long division and the remainder is 0 then there are no errors



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Step | N | D(t)p(t) | D(v)p(v) | D(u)p(u) | D(x)p(x) | D(w)p(w) | D(y)p(y) |
| 0 | Z | Inf | Inf | Inf | 8,z | Inf | 12,z |
| 1 | Zx | Inf | 11,x | Inf | 8,z | 14,x | 12,z |
| 2 | Zxv | 15,v | 11,x | 14,v | 8,z | 14,x | 12,z |
| 3 | Zxvu | 15,v | 11,x | 14,v | 8,z | 14,x |  |
| 4 | Zxvuy | 15,v |  | 14,v |  |  |  |
| 5 | Zxvuyw |  |  |  |  |  |  |
| 6 | zxvuywt |  |  |  |  |  |  |

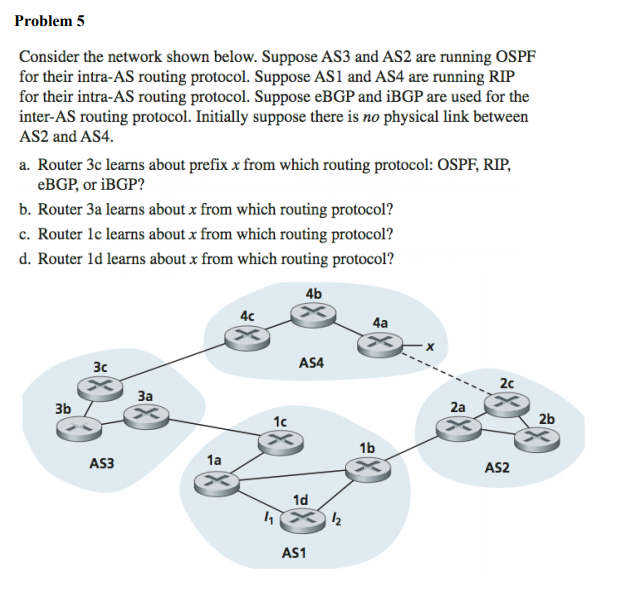
We can see from this graph that all nodes are covered and we can see that to node z the shortest path from every node is listed so I will add a path that shows the fastest routes. Meaning that to get to T from Z would take 8+3+4 =15 cost or u to z would be 8+3+3 = 14 etc.



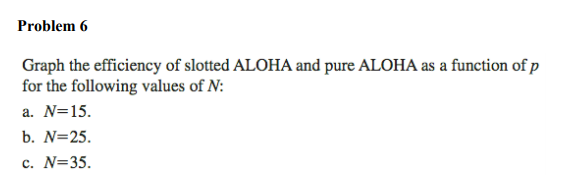
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Cost |  |  |  |
|  |  | **u** | **v** | **x** | **y** | **z** |
| From | **v** | Inf | Inf | Inf | Inf | inf |
|  | **x** | Inf | Inf | Inf | Inf | Inf |
|  | **z** | Inf | 6 | 2 | Inf | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Cost |  |  |  |
|  |  | **u** | **v** | **x** | **y** | **z** |
| From | **v** | 1 | 0 | 3 | Inf | 6 |
|  | **x** | inf | 3 | 0 | 3 | 2 |
|  | **z** | Inf | 6 | 2 | Inf | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Cost |  |  |  |
|  |  | **u** | **v** | **x** | **y** | **z** |
| From | **v** | 1 | 0 | 3 | 3 | 5 |
|  | **x** | 4 | 3 | 0 | 3 | 2 |
|  | **z** | 6 | 5 | 2 | 5 | 0 |

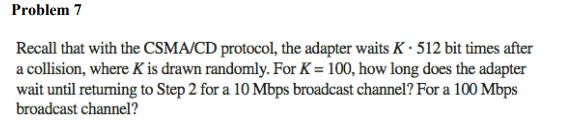


1. Router 3c would learn about x from eBGP
2. Router 3a would learn about x from iBGP
3. With eBGP
4. iBGP



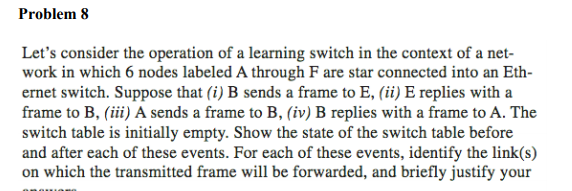
|  |  |  |  |
| --- | --- | --- | --- |
| p | e |  |  |
|  | n = 15 | n = 25 | n = 35 |
| 0 | 0 | 0 | 0 |
| 0.1 | 0.343151 | 0.199 | 0.09734 |
| 0.2 | 0.131941 | 0.0236 | 0.003549 |
| 0.3 | 0.030520 | 0.0014 | 5.682E-05 |
| 0.4 | 0.00470 | 4.74E-05 | 4.011E-07 |
| 0.5 | 0.000457 | 7.45E-07 | 1.018E-09 |

|  |  |  |  |
| --- | --- | --- | --- |
| p | e |  |  |
|  | n = 15 | n = 25 | n = 35 |
| 0 | 0 | 0 | 0 |
| 0.1 | 0.07850 | 0.01590 | 0.00270 |
| 0.2 | 0.00580 | 0.000111 | 1.7997E-06 |
| 0.3 | 0.000206 | 2.752E-07 | 3.0750E-10 |
| 0.4 | 3.684E-06 | 2.245E-10 | 1.149E-14 |
| 0.5 | 2.793E-08 | 4.440E-14 | 5.929E-20 |



Since K \* 512 bits we can see that its 51200 bits so for a 10 Mbps signal it would be bit time / bps

Which would be 5.12 ms, other wise for a 100 Mbps broadcast channel the wait time would be 0.512 ms



i) **B sends a frame to E**

State: Switch learns interface corresponding to mac of b

Link forwarded to: a,c,d,e and f

* The switch doesn’t know the interface corresponding to mac of e since switch table is empty

ii) **E replies with a frame to B**

State: Switch learns interface corresponding to mac of e

Link forwarded to: b

* The switch knows interface corresponding to mac of b

iii) **A sends a frame to B**

State: Switch learns the interface corresponding to mac of a

Link forwarded to: b

* The switch already knows the interface corresponding to mac of b

iv) **B replies with a frame to A**

State: Switch table state remains the same

Link forwarded to: a

* The switch already knows the interface associated with mac of a