

Lab4

311511034 黃聖偉

模擬結果

Avg PDR				
Packet Num			1	10000
Routing Protocol	Flow Count	Flow ID		
AODV	1	1	1	0.1728
DSDV	1	1	1	0.1791
AODV	2	1	1	0.0744
		2	1	0.0813
DSDV	2	1	1	0.0684
		2	1	0.1081

Avg System Throughput (Kbps)				
Packet Num			1	10000
Routing Protocol	Flow Count	Flow ID		
AODV	1	1	25.8658	171.233
DSDV	1	1	42.9365	174.873
AODV	2	1	26.3098	71.7255
		2	24.5375	78.3779
DSDV	2	1	53.567	65.9424
		2	46.5513	104.758

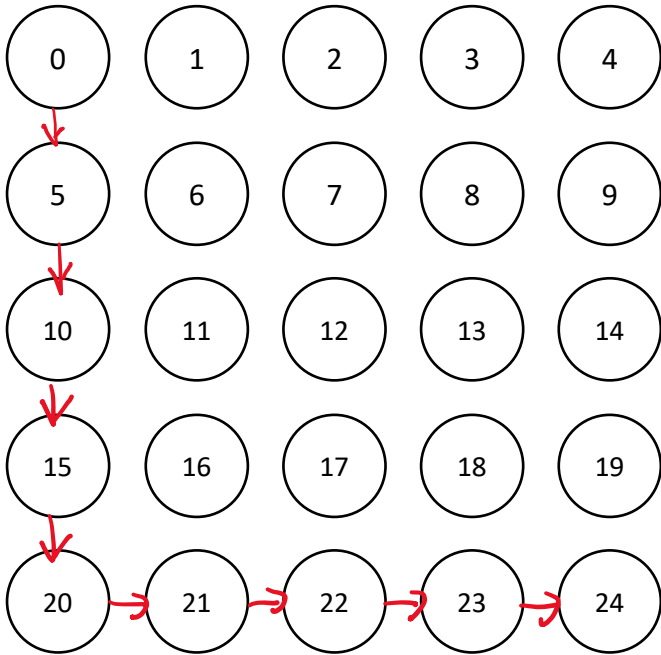
Avg Delay (s)				
Packet Num			1	10000
Routing Protocol	Flow Count	Flow ID		
AODV	1	1	0.159477	1.41717
DSDV	1	1	0.096072	1.42206
AODV	2	1	0.156786	2.09605
		2	0.16811	1.91802
DSDV	2	1	0.0770064	2.25922
		2	0.088612	1.90617

Routes

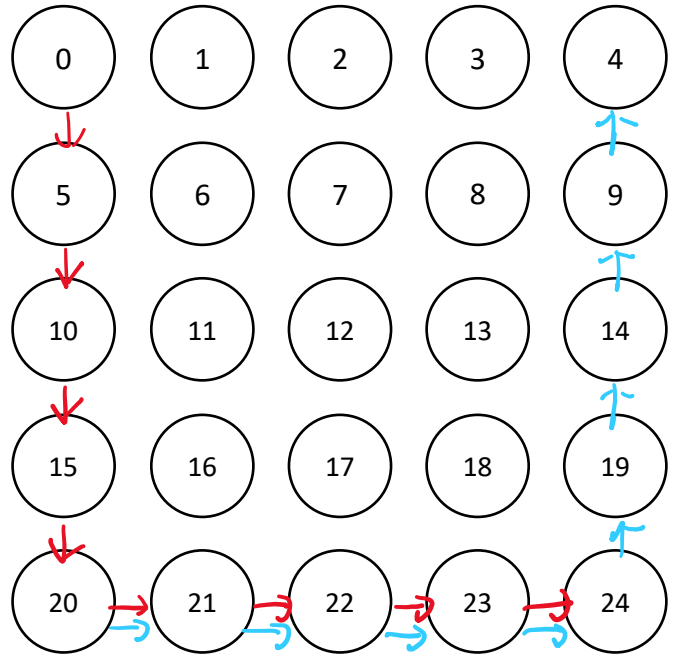
DSDV

→ flow 1 → flow 2

1 flow

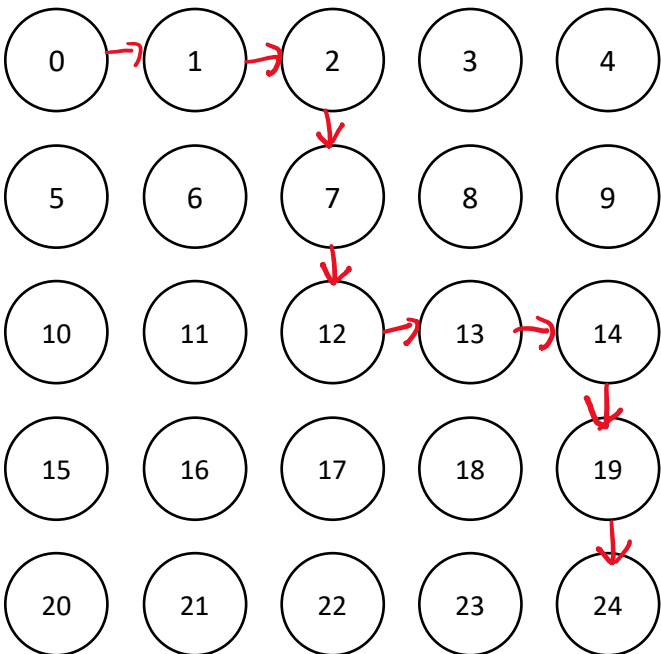


2 flow

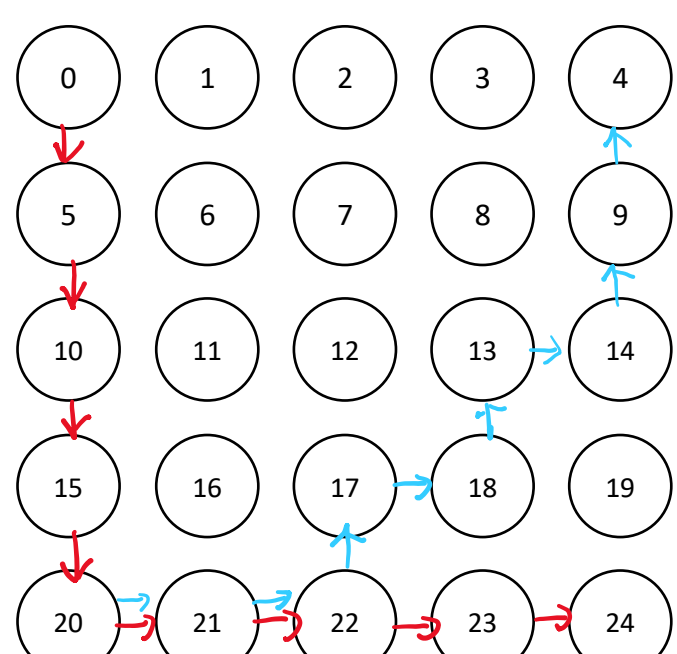


AODV

1 flow



2 flow



實驗結果分析

在傳輸一個 packet 時，不論 flow 數量是多少，可以很明顯地看到 DSDV 在 throughput 上比 AODV 高出許多，同時在 End-to-End delay 上 DSDV 也比 AODV 低了許多。我認為原因是 DSDV 在傳送 data packet 前就已經建好了 routing table，而每個 node 中也都有自己的 routing table，紀錄著所有 destination node 所對應的 next-hop 是哪一個 node。因此在 sender 收到 data packet 要準備轉傳時，不需要再去找 path，直接根據已經建好的 routing table 就可以將 packet 送到下個 node。AODV 則沒有預先建立 routing table，而是在 node 要送出 packet 時，才會開始找 path。因此 AODV 有比較高的 delay，在只送一個 packet 的情況下造成 throughput 也比較低。

在 10000 個 packet 而且只有一個 flow 時，可以看到 DSDV 與 AODV 在 throughput、End-to-End delay 與 PDR 都幾乎一樣。我認為這是因為 AODV node 只有在第一次發送 data packet 時才需要找 path，但當發送出第一個 data packet 後，在 forward path 中的所有 node 就會記錄這個找到的 path 中自己的 next-hop 是誰，因此之後再收到相同目的地的 node 就不用在重新找 path，所以在多個 packet 的情況下 DSDV 與 AODV 的效能是差不多的。

在 10000 packets 而且有兩個 flow 時，可以看到 DSDV flow1 與 flow2 的 throughput、End-to-End delay 與 PDR 效能差異較大，而 AODV flow1 與 flow2 的 throughput、End-to-End delay 與 PDR 效能差異較小。我認為這是因為 AODV 在發送 data packet 時才會尋找 path，這樣可以有辦法使 flow1 與 flow2 在傳輸的路徑上可以盡量不重疊，也就是盡量不經過相同的 node。而 DSDV 因為在發送前就尋找 path，因此造成 flow1 與 flow2 在傳輸的路徑上很容易經過相同的 node，也就更容易造成壅塞，導致其中一個 flow 先佔住 node 的資源，使兩個 flow 得到的結果相差較大。這個現象也可以從 routes 結果看到，在兩個 flow 時，DSDV 有比較多重疊的 node，AODV 則比較少。