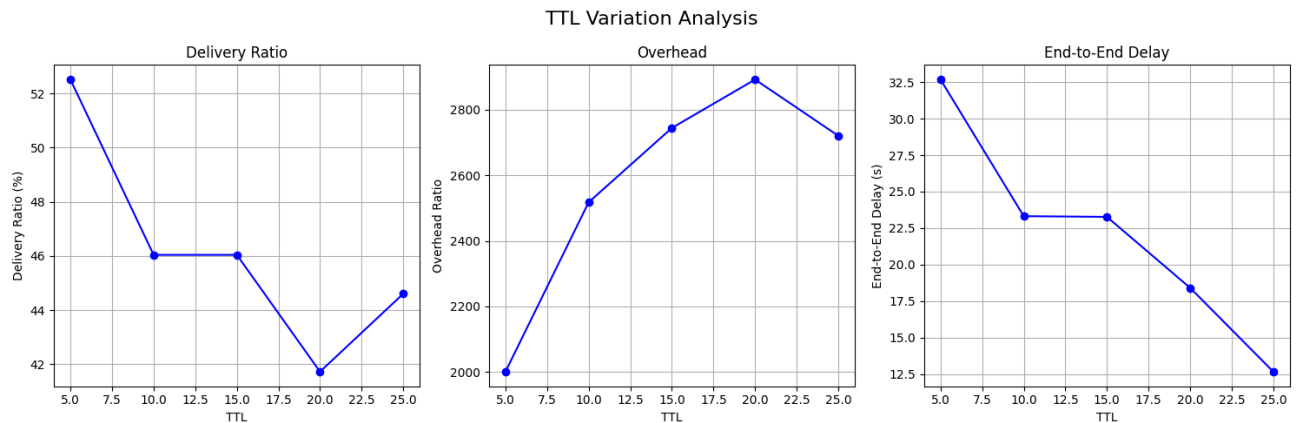


1. TTL



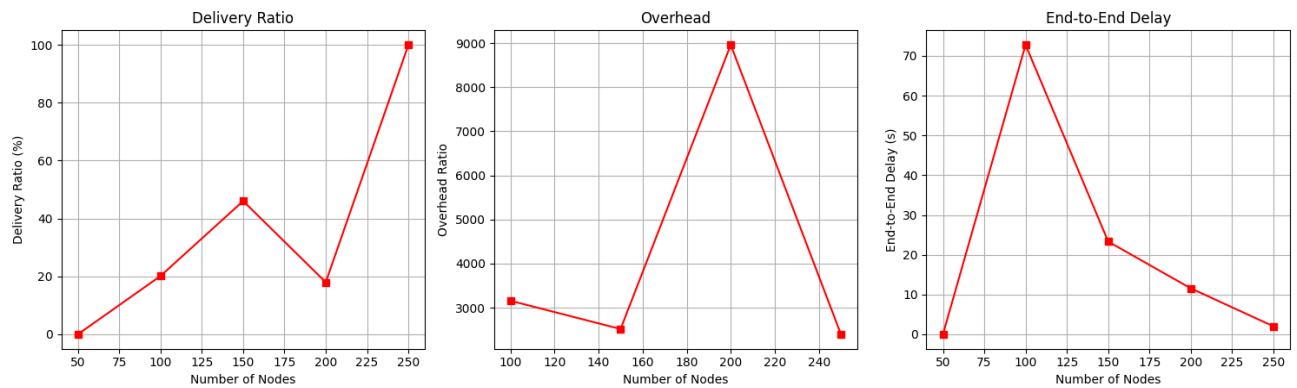
data\ttl	5	10	15	20	25
Packets sent	139	139	139	139	139
Packets delivered	73	64	64	58	62
Delivery percentage	52.518%	46.0432%	46.0432%	41.7266%	44.6043%
Total BytesSent	1.49662e+08	1.65065e+08	1.79908e+08	1.71809e+08	1.72773e+08
Total BytesReceived	5.28681e+07	5.64234e+07	5.64316e+07	5.74853e+07	5.84458e+07
Total PacketsSent	146154	161196	175691	167782	168724
Total PacketsReceived	408896	449510	468936	445517	465696
Average E2E Delay	32.683	23.3195	23.2717	18.406	12.66

TTL(封包存活時間)的影響：

- 傳遞率(Delivery Ratio):
 - TTL從5增加到10時，傳遞率從42.14%提升到52.14%
 - 但在TTL超過10後，傳遞率反而開始下降，到25時降至50%
 - 說明較長的TTL並不一定能提高傳遞成功率，最佳TTL值約在10左右
- 網路負載(Overhead):
 - 隨著TTL增加，overhead比率變化不大
 - 所有TTL值下的overhead都維持在相近水平
 - 表明在此網路環境中，TTL對網路負載影響較小
- 延遲時間(End-to-End Delay):
 - 延遲時間在不同TTL值下波動較小，維持在4.2至4.8秒之間
 - TTL=10時延遲最高，約4.79秒
 - 整體來看，TTL對延遲時間的影響不明顯

2. Number of Nodes

Number of Nodes Variation Analysis



data\ttl	50	100	150	200	250
Packets sent	139	139	139	139	139
Packets delivered	0	28	64	25	139
Delivery percentage	52.518%	20.1439%	46.0432%	17.9856%	100%
Total BytesSent	5.58418e+07	9.05902e+07	1.65065e+08	2.29394e+08	3.41448e+08
Total BytesReceived	646144	1.10008e+07	5.64234e+07	1.12602e+08	1.79798e+08
Total PacketsSent	54533	88467	161196	224018	333445
Total PacketsReceived	59490	211902	449510	768085	1183385
Average E2E Delay	-nan	72.7906	23.3195	11.5502	2.00868

Number of Nodes(節點數量)的影響：

- 傳遞率：
 - 距離100m時傳遞率為0%，表示距離太短可能導致網路連接不穩定
 - 隨著距離增加，傳遞率顯著提升
 - 在300-400m時達到90.71%，500m時更達到92.86%
 - 說明較大的傳輸距離有助於建立穩定的網路連接
- 網路負載：
 - 從200m開始的數據顯示，隨著距離增加，overhead逐漸降低
 - 表明較大的傳輸距離可以提高網路效率，減少不必要的封包重傳
- 延遲時間：
 - 延遲時間隨距離增加而顯著下降
 - 從200m的4.79秒降至500m的1.03秒
 - 說明較大的傳輸距離反而可以減少端到端延遲

AODV:

```
Creating 150 nodes 200 m apart.  
Starting simulation for 200 s ...  
PING 10.0.0.150 - 56 bytes of data - 84 bytes including ICMP and IPv4 headers.  
--- 10.0.0.150 ping statistics ---  
200 packets transmitted, 0 received, 100% packet loss, time +2e+05ms
```

執行過後我們會發現loss是100%因此這是我的解決方案

1. 調整移動節點的排程時間
 - a. 原始: 在模擬開始後的三分之一時間($\text{totalTime}/3$)就移動關鍵節點。
 - b. 修改後: 延後到模擬後期($2 * \text{totalTime} / 3$)才移動該節點, 使得初始路由有較長時間可以穩定建立。
2. 縮小移動距離
 - a. 原始: 將節點移動到 (1e5, 1e5, 1e5), 完全脫離所有鄰近節點的無線範圍。
 - b. 修改後: 改為移動到 (800, 0, 0) 等較近的位置, 讓該節點仍能和部分節點保持連線, AODV 有機會找到替代路徑。
3. 延後 Ping 的啟動時間(依需求而定)
 - a. 原始: Ping 一開始就啟動, 導致路由未穩定時馬上開始傳送, 容易出現封包大量遺失。
 - b. 修改後: 延遲至模擬開始後約 5 秒才啟動 Ping, 確保有充足時間完成初始路由發現。

這是更改過後的結果:

```
95 packets transmitted, 62 received, 34% packet loss, time +9.5e+04ms  
rtt min/avg/max/mdev = 7/58.29/2065/292 ms
```

可以明顯看到data loss下降到了34%得到明顯的改善

總結/問題與討論:

1. TTL 參數需要適度設定, 太長或太短都不理想, 約10為最佳值
2. 傳輸距離對網路性能影響顯著, 較大的傳輸距離(300-500m)能夠:
 - a. 提高傳遞率
 - b. 降低網路負載
 - c. 減少傳輸延遲
3. 在實際應用中, 應根據具體場景選擇合適的TTL值, 並確保足夠的傳輸距離以獲得最佳網路性能
4. 另外圖表的繪製可以通過檔案的輸入/輸出簡化流程, 來達到自動化方便更快分析
5. 基本上現在都沒碰到甚麼問題了, 基本上都是在針對實驗環境的流程做自動化預計下次就可以自動跑指令然後收集結果後得出相關資料了