



CHAPTER IV

EXPERIMENTAL RESULTS

4.1 Data transfer method - Network Coding-Based Relay for IEEE 802.16j Multi-hop Relay Network

In order to prove the efficiency of the proposed NC-BR mechanism and the frame structure, QualNet 4.5 was extended to support the multi-hop relay function. The simulation results to demonstrate the performance of the proposed scheme are elaborated as follow.

4.1.1 Simulation model

In order to represent a realistic IEEE 802.16j Multi-hop Relay Network, the simulation model comprise of six scenarios which are 3-hop, 4-hop, and 5-hop scenarios, with and without the NC-BR scheme. Each MR-BS/RS covers 20 MSs. Each MS communicates to the MR-BS with the similar traffic. Figure 3.6 shows an example of 5-hop scenario configuration. The following parameters are used in the simulation: 300s of traffic flows generated, 64QAM modulation, 20ms for frame duration, 10 μ s for TTG/RTG, 4 μ s for SSTG, 32-2048 bytes for the data size with Pareto distribution, and the arrival time interval with 0.1ms mean in Poisson distribution. The simulation properties are shown in Table 4.1.

Table 4.1: Simulation properties.

Term	Values
MAC Layer Properties	
Frame Duration	20 ms
Base Station MAC propagation delay	1 us
Transmit/receive Transition Gap (TTG)	10 us
Receive/transmit Transition Gap (RTG)	10 us
SS Transition Gap (SSTG)	4 us
PHY Layer Properties	
Transmission power	15 dBm
Antenna efficiency	0.8
Antenna model	Omni directional
Traffic Model Properties	
Data size (min-max)	Pareto distribution, 32-2048 bytes
Arrival, mean interval	Poisson, 0.1 ms

4.1.2 Simulation results

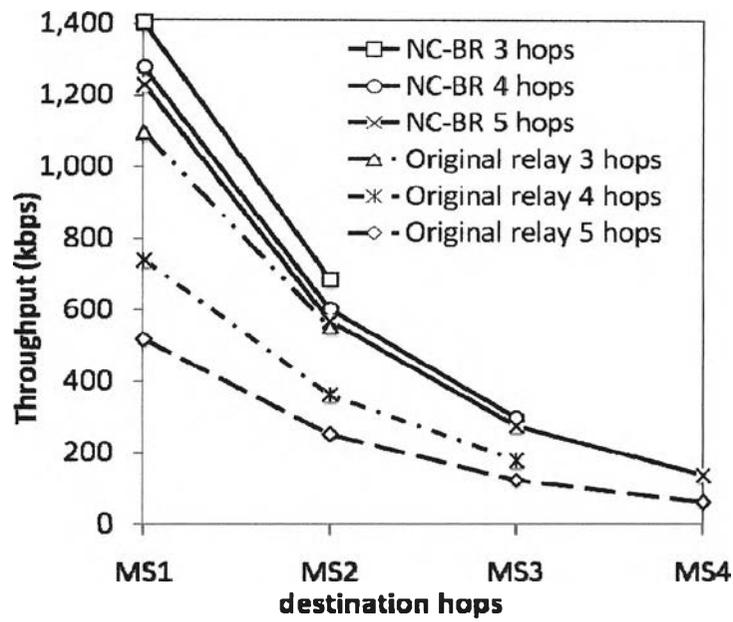


Figure 4.1: Throughputs of 3-5 hops scenarios of the NC-BR comparing to the original relay schemes.

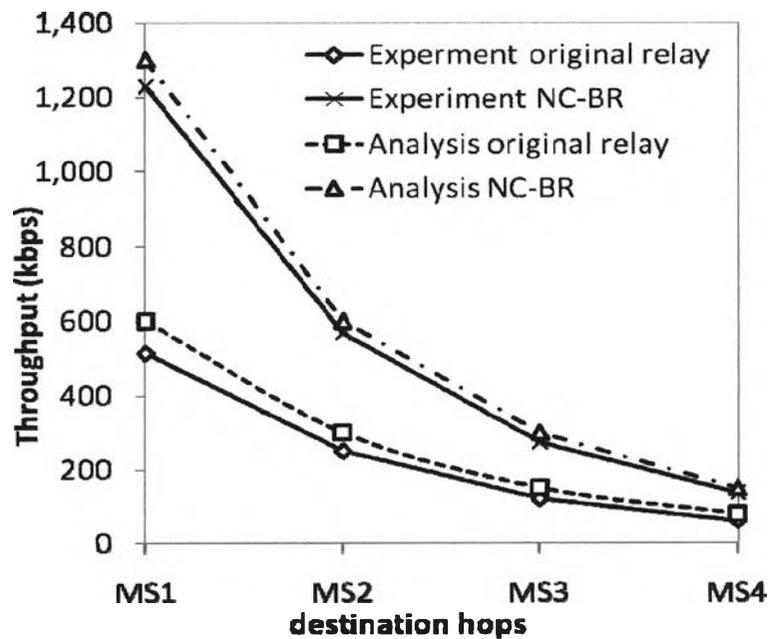


Figure 4.2: Throughputs of the original and the NC-BR comparing to analysis results of 5 hops scenario.

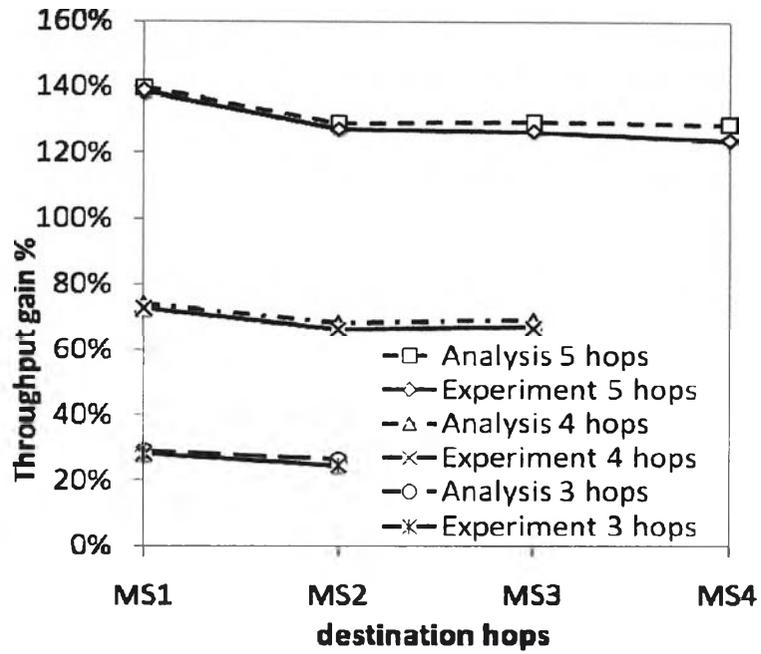


Figure 4.3: Percentages of the throughput gained under 3-5 hops scenarios.

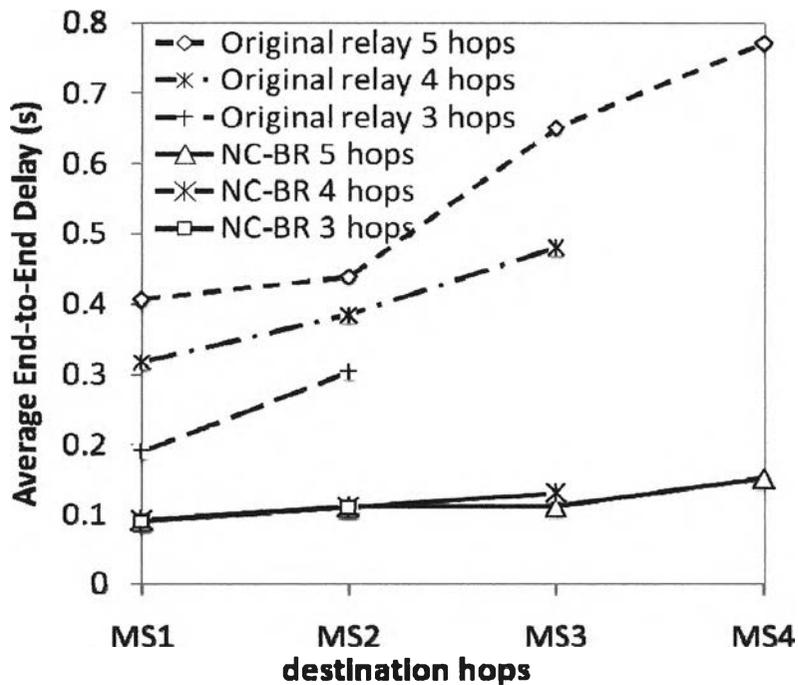


Figure 4.4: Average end-to-end delays of original and NC-BR of 3-5 hops scenarios.

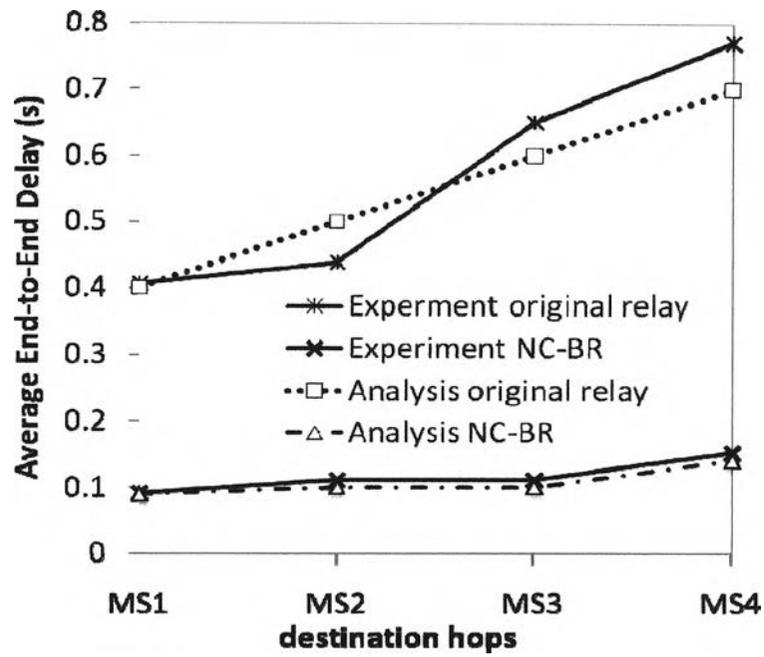


Figure 4.5: Average end-to-end delays of the original and the NC-BR comparing to analysis of 5 hops scenario.

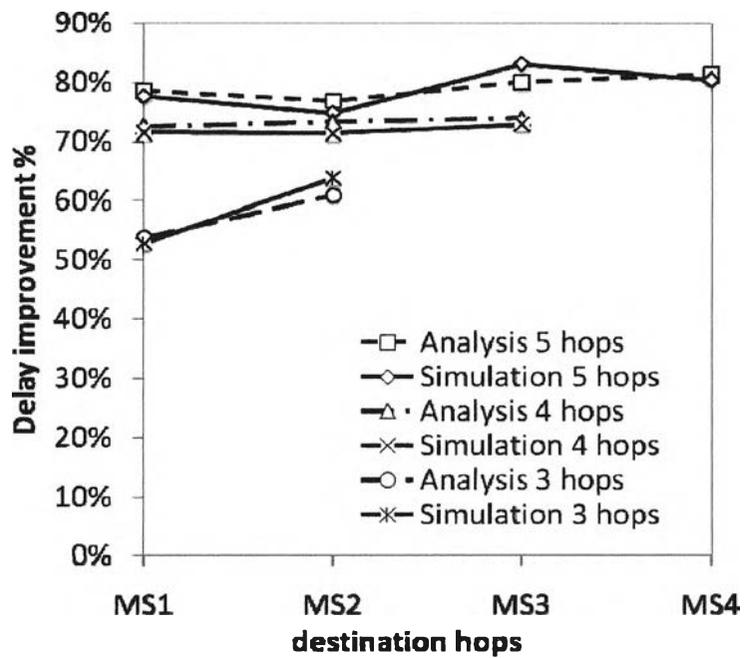


Figure 4.6: Percentages of the delay improvement under 3-5 hops scenarios.

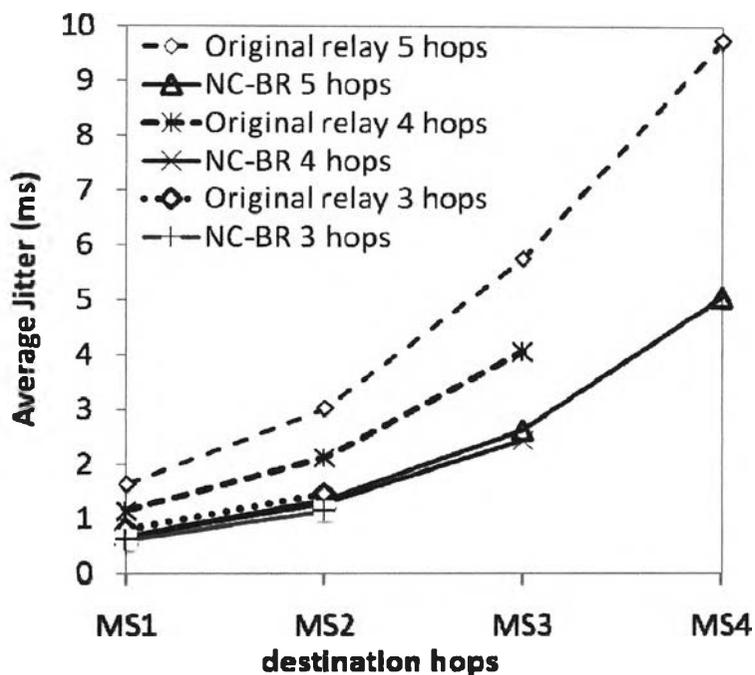


Figure 4.7: Average jitters of the original and the NC-BR under 3-5 hops scenarios.

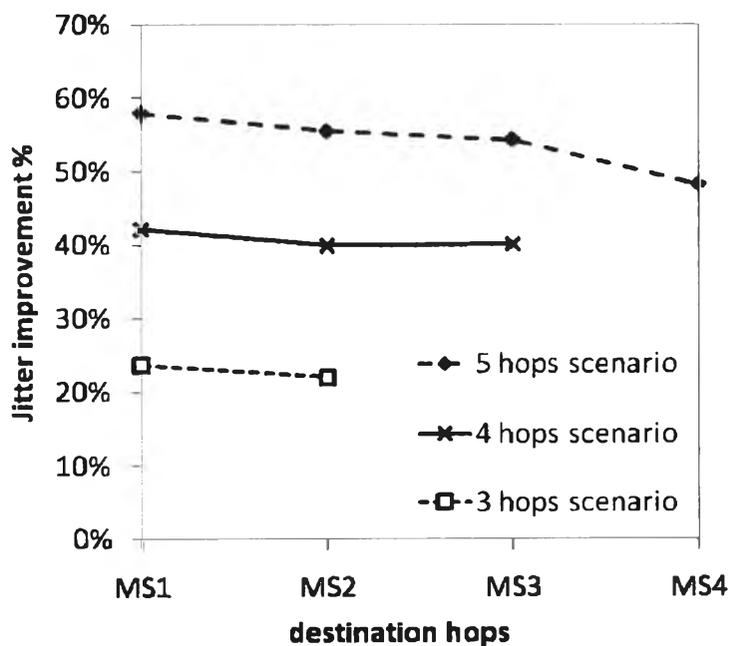


Figure 4.8: Percentages of jitter improvement under 3-5 hops scenarios.

According to the experimental results, throughputs were measured in Figure 4.1, Figure 4.2, and Figure 4.3. Figure 4.1 shows comparisons between throughputs of 3-5 hops scenarios of the original relay against the NC-BR. The results show a significantly higher throughput of the NC-BR comparing with the original relay scheme.

Figure 4.2 shows the comparison between the theoretical values from the Equation (3) against the simulation results of 5-hop scenario, based on the original relay and the NC-BR. The results show that the analysis results and the simulation results are closed to each other; thus, the accuracy of the theoretical model (3) is confirmed.

Additionally, Figure 4.3 shows the percentages of the throughput gained from the NC-BR, comparing with the original relay approach. These percentages are derived from the results shown in Figure 4.1. Referring to Figure 4.3, the throughput gained are 24-28%, 67-73%, and 124-140% on 3-hop, 4-hop, and 5-hop scenarios respectively.

Delay was measured and shown in Figure 4.4, Figure 4.5 and Figure 4.6. According to Figure 4.4, the comparison between the average end-to-end delays of total six scenarios, 3-5 hops scenarios of the original relay comparing against the NC-BR. The results show that in the case of the number of hops is equal, the NC-BR scenarios provides significantly lower the end-to-end delay than the original relay scheme.

Figure 4.5 shows the comparison between the theoretical values calculated from the theoretical model (4) against the simulation results of 5-hop scenario, with the original relay and the NC-BR. The result shows that, on the NC-BR, there is a small difference between the analysis and the simulation results; this confirms the accuracy of the proposed theoretical model (4). Based on Figure 4.5, the difference between the simulation and theoretical values of the original relays is very clear because there is an unpredictable delay variation from the queue length and packet dropped at each hop during the simulation period.

As a consequent of the experiment, the jitter values are obtained. Figure 4.7 shows comparisons between 3-5 hops scenarios of the original relay against the NC-BR. Results show that when the number of hops is equal, NC-BR scenarios provide much lower jitter than the original relay approach.

In order to obtain a clear picture of the performance improvement after applying the NC-BR, improvement metrics, the percentages of delay improvement and percentages of jitter improvement, are used as the indicators as shown in Figure 4.6 and Figure 4.8. According to Figure 4.6, the delay improvement is 53-64% on 3-hop, 71-73% on 4-hop, and 75-83% 5-hop scenarios. Moreover, Figure 4.8 shows that the jitter improvement is 22-24% on 3-hop, 40-43% on 4-hop, and 49-58% on 5-hop scenarios.

4.2 Path finding - Adaptive Nearest Neighbor Algorithm on Hierarchical Weighted-Index Graph

With the proposed algorithms, the adaptive travel time path selection algorithms (ATTPS) and the Hierarchical Index on Graph Layering Architecture (HIRN) system design, prove that the algorithms run on the less number of computations, and more accurate results than other existing methods are presented. In the HIRN, the number of all vertices is N . Time complexity of the ATTPS is $O(mn)$, where n is the number of vertices that are located only on HIRN level 1 and m is the number of sets of potential vertices to be shortcut to all vertices of HIRN level 1. The proposed algorithms process much less vertices than existing methods because of the advantages from the combination of the hierarchy index structure, Euclidean distance and stored historical weight data. Thus, the total processing time is smaller than the referenced methods [1, 7-8, 10, 47-95].

4.2.1 Path finding problem: Case study

The one classical problem was selecting the best path to travel from one point to another by least travel-time as possible. This is the significant interest under the problem of individual travel and business logistic planning. In order to solve this problem, the

required preliminary information was a digital map of possible transportation path (e.g. the road map for land transportation, the marine/watercourse map for waterway transportation). Then calculations have been performed to find the shortest-path from the source to the destination according to the digital map, as the viewed directed weight graph. From the existing method, the weight of an edge connecting between a pair of vertices in the graph from the digital map was an actual distance. The result of this method is the shortest-distance rather than the fastest time. Thus, position-aware shortest-path algorithm can increase the efficiency. A recent method [1, 7-8, 10, 47-95] has performed conversion from the travel-distance to the travel-time using speed-limit of each transportation path or actual travel-time of an individual mobile unit that submitted to the datacenter.

In the real situation, especially in a high traffic congestion environment, various factors can influence the travel-time to the destination (e.g., congestion, driving behavior, vehicle limitation and special congestion event). Unfortunately, the existing methods cannot solve this problem.

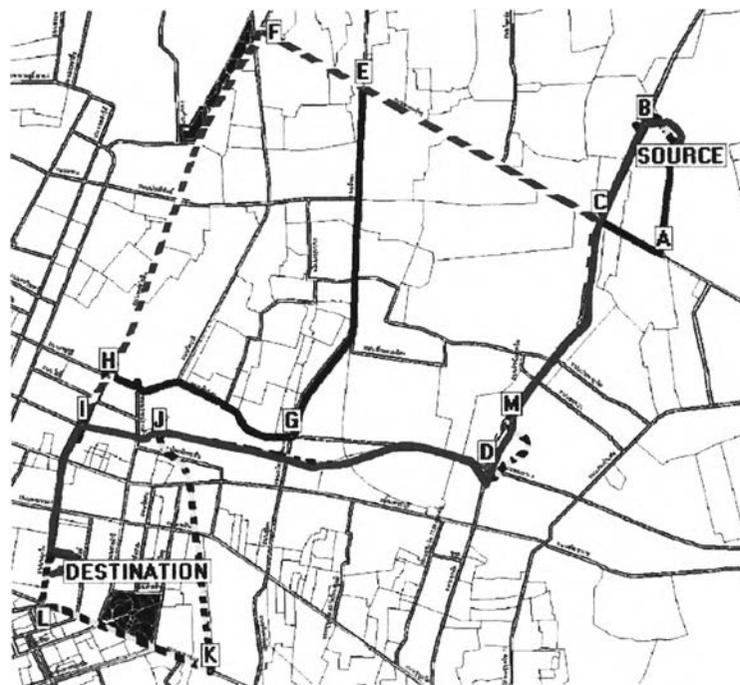


Figure 4.9: The 4 alternative paths from Bangkok city road network.

Table 4.2: Travel time information collected from Figure 4.9.

Path	Set of Vertices	Distance (Km.)	Travel time (Hrs.)	Average Speed (Km/Hrs.)
Path 1 (Gray line)	{Source, B, C, M, D, J, I, Destination}	16.837	1:01:02	16.547
Path 2 (Black line)	{Source, A, C, E, G, H, I, Destination}	18.103	1:08:05	15.950
Path 3 (Gray/White line)	{Source, B, C, E, F, H, I, Destination}	19.615	0:36:43	32.053
Path 4 (Gray/Black line)	{Source, B, C, M, D, J, K, L, Destination}	21.293	0:47:08	27.106

Considering Figure 4.9, the collecting information from a real road network of Bangkok City, shows that there are 4 alternative paths from the source to the destination point. Each path has different distances and details of distances, travel-time and average speeds, presented in Table 4.2.

From Table 4.2, it is cleared that the travel-time was not directly relevant to the distance. Path 1 was the shortest-path in term of the distance. On the other hand, the shortest-travel-time path, Path3, is what drivers expected because the mobile unit can travel with a higher speed through Path3 due to less congestion in the road network. Thus, in the real situation, the congestion can cause a large impact on the shortest-path selection in term of the travel-time. Beyond that, periods of time during a day and the days during a week can indicate a big difference on congestion of each road (e.g., in weekday rush hours, the road approach downtown will be very busy. In other hand, the same road will almost empty by late of night time.)

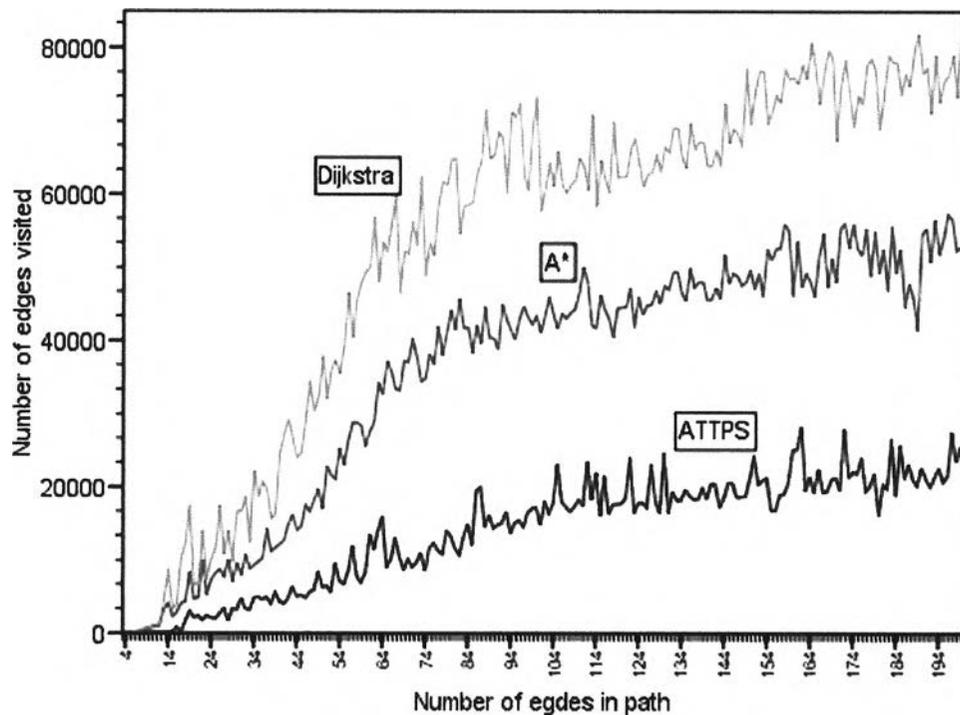


Figure 4.10: Comparison chart of simulation test, implement on data from TIGER/Line road map of California State.

4.2.2 Simulation Results

Consider Figure 4.10, the simulation was performed based on three algorithms: Dijkstra, A*, and ATTPS. The simulation programs of these three algorithms are implemented using Visual C++ .Net 2005. This experiment uses a graph from the TIGER/Line road map of CA that consists of 195,902 vertices (V) and 534,239 edges (E), which is large enough to represent the characteristics of the real world road network.

In the simulation situation of the proposed method, the HIRN was created for the used road map. Then, the ATTPS is performed on the HIRN index. For other two standard algorithms, the data were firstly loaded into the main memory before passing through the algorithms.

Referring to Figure 4.10, it is clear that under the situation of visiting a large volume of paths, the ATTPS has the number of the visiting vertices approximately four times less than the Dijkstra algorithm, and two times less than A* algorithm.

4.3 Integration of the path finding part and the data transfer part.

Referring to section 3.3, the integration of the path finding part and the data transfer part was proposed, figure 4.1 shows the experimental growth of the storage space in the centralized HIRN, the distributed HIRN and the existing path finding methods.

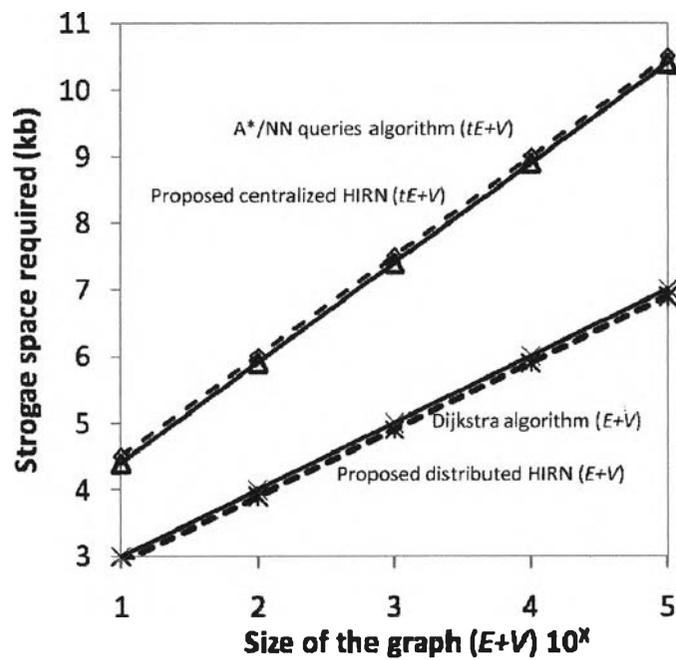


Figure 4.11: The growth of storage space in the centralized HIRN, the distributed HIRN, A*/NN queries algorithm and Dijkstra algorithm.