Mansoura Engineering Journal

Volume 34 | Issue 3 Article 4

3-18-2021

Comparative Study among Routing Algorithms under Different Circumstances.

I. Zedan
Faculty of Engineering., Zagazig University., Zagazig., Egypt.

I. El- Henawy
Faculty of Computer and Informatics., Zagazig University., Zagazig., Egypt.

M. Abdallah
Faculty of Engineering., Zagazig University., Zagazig.,Egypt.

E. Mohamed
Faculty of Computer and Informatics., Zagazig University., Zagazig., Egypt.

COMPARATIVE STUDY AMONG ROUTING ALGORITHMS UNDER DIFFERENT CIRCUMSTANCES

دراسة مقارنة بين خوارزميات تحديد المسار تحت تأثيرالظروف المختلفة

I. E. Zedan[1], I. M. El- Henawy[2], M. I. Abdallah[3], E.R. Mohamed[4]

[1] Prof., Faculty of Engineering, Zagazig Univ.
[2] Prof., Faculty of Computer and Informatics, Zagazig Univ.
[3] Associate Prof., Faculty of Engineering, Zagazig Univ.
[4] Assistance teacher, Faculty of Computer and Informatics, Zagazig Univ.

الملخص العربي

ان الهدف من تقنيات تحديد المسار في شبكات الاتصال هو توجيه البيانات من المصدر الي المستقبل الصحيح. تقوم الشبكات باستخدام خوارزميات تحديد المسار لحساب وتحديد المسار المناسب لنقل البيانات . يجب ان تتفاعل خوارزميات تحديد المسار مع التغيرات التي تحدث في كل من تركيب الشبكة أوالاحمال الموجودة بدون توقف لوظائف العقد الموجودة في الشبكة.

يجب ان تتصف خوارزميات تحديد المسار بخصائص مثل القوة والعدل والثبات والمثالية.

يمكن تقسيم خوارزميات تحديد المسار الي نوعين رئيسيين , خوارزميات ثابتة وخوارزميات متكيفة . يمكن التعبير عن أداء هذه الخوارزميات بمقاييس مختلفة . كما يوجد كثير من العوامل التي تؤثر في أداء هذه الخوارزميات مثل حالة كل من المصدر والمستقبل والعقد البينية وكذلك حالة الشبكة.

ونتيجة لأن خوارزميات تحديد المسار لها تأثير كبير علي أداء الشبكة فان الهدف من هذا البحث هو دراسة جودة أداء هذه الخوارزميات تحت تأثير الظروف والعوامل المختلفة وبالتالي مساعدة مصممي الشبكات في اختيار الخوارزم الأمثل الذي يتناسب مع العوامل الموجودة . يتم ذلك عن طريق تقدير دالة ذات كفاءة مبنية علي العوامل المختلفة المؤثرة في أداء هذه الخوارزميات . والبحث يقدم طريقة لحساب هذه الدالة التي علي أساسها يستطيع المصمم اختيار الخوارزم الأمثل.

Abstract

The goal of routing in a communication network is to direct user traffic from a source to the correct destination in accordance with the network's service requirements. The routing algorithm is used by network to compute the path that would best serve to transport the data from the source to the destination. The routing algorithm should be able to cope with changes in the topology and traffic without requiring all jobs to be aborted in all hosts. The robustness, stability, fairness and optimality are main properties that the routing algorithms should often have.

The routing algorithms can be grouped into two major classes: non adaptive and adaptive. The performance of these routing algorithms may be expressed in terms of various criteria. There are many factors that affected the performance of these algorithms as source and destination capabilities, intermediate nodes capabilities and finally network parameters.

Because the routing algorithms have so much impact on the overall performance of any network, the objective of the present study is to compare between six routing algorithms by evaluating their performance under various circumstances. This is done by developing an efficient estimation function that depends on various network parameters.

Developing this function helps network designers to select the appropriate routing algorithm for their networks.

KEYWORDS

Routing algorithm - Performance evaluation - Estimated function- Network parameters

Accepted September 3, 2009.

1. Introduction

The routing algorithm is used to establish the appropriate routing paths or the equivalent routing table entries in each node along a path. The routing algorithms can be classified into two major classes; non adaptive and adaptive. Nonadaptive routing algorithms do not base their routing decisions on measurements or estimates of the current traffic and topology. Instead, the choice of route is computed in advance this procedure is some times called static routing. These algorithms are simple and work well in environments where network traffic is relatively predictable and where network design is relatively simple. The most famous Non adaptive algorithms are Bifurcated, Dijkstra and Flooding routing algorithms [1].

Adaptive routing algorithms can change their routing decisions to reflect changes in topology and the traffic. Adaptive algorithms differ in where they can get their information, when change the routes and what metric is used for optimization. This procedure is some times called dynamic routing. The most famous adaptive algorithms are distance vector, link state and Baran's routing algorithms [2, 3].

In general, a good routing algorithm should have the following properties:

1. Robustness with respect to failures and changing conditions.
2. Stability of the routing decisions.
3. Fairness of the resource allocation.
4. Optimality of the packet travel time.

Robustness means that the routing algorithm must adjust the routing decisions when equipment fails and when traffic conditions change. A robust routing algorithm rapidly detects change in operating conditions and reacts fast and appropriately to those changes either frequently or soon after a change is detected. The robustness requirement implies three tasks of the routing algorithm namely monitoring the network, making the routing decisions and implementing these decisions.

Stability means that algorithm should perform decisions smoothly to changes in operating conditions. That is, a small change in operating conditions should provoke a comparatively small change in routing decisions. Instability could result from the effect of the routing decisions on the operating conditions. That is, a change in conditions could modify the routing decisions, which would in turn change the operating conditions.

The routing algorithm is fair if it results in similar delays for the packets of different sources and destinations. This definition of fairness should be modified when the routing algorithm is designed to accommodate communication services with different qualities of services.

The routing algorithm is optimal if it maximizes the network designer's objective function while satisfying design constrains. A typical choice for the objective function is the rate revenues for the network when one assumes that the users pay a given cost per transmitted packet. The cost per packet depends on the quality of service provided to the user. Another commonly selected objective is to minimize the average delay per packet. Optimality and fairness are not always compatible. Such incompatibility may be caused by an objective function that doesn't reflect properly the operating cost of the network. [4,5].

The performance of routing algorithms is expressed in terms of the following criteria:

1. Performance criteria

The efficient measures of the performance of the routing algorithms are:

- Number of Hops
- Cost
- Average delay for packet
- Throughput
- Memory requirement
- CPU processing at nodes

2. Decision time and place

The routing algorithm can select the appropriate route for either each packet or for the entire session. Also the decision can be taken in each node or at central node or at originating one.

3. Information source

All nodes should recognize to all variations in the network status. This is performed by exchanging information with other nodes. The source of this information can be one of the following:

- All nodes
- local nodes
- Adjacent nodes
- Nodes along route
- None

4. Information update

The routing algorithm should update information to all nodes in the network. This is occurred:

- Continuous
- Periodic
- Major load change
- Topology change.

2. Routing algorithms

Routing algorithms can be classified into two major classes; Nonadaptive routing algorithms and adaptive routing algorithms.

2.1 Nonadaptive routing algorithms

These algorithms don't base their routing decisions on measurements or estimates of the current traffic, but the choice of route to get from any node I to any node J is computed in advance, off-line, and downloaded to all nodes when network is booted. Three nonadaptive routing are summarized here 1,4,5,6].

2.1.1 Flooding

Flooding is a very simple routing algorithm in which every incoming packet is sent out on every outgoing route except the one it arrived from. Flooding has two interesting characteristics that arise from the fact that all possible routes are tried. As long as there is a route from source to destination, the delivery of the packet is guaranteed. One copy of the packet arrives by the quickest possible route.

Flooding obviously generates vast numbers of duplicate packets. In fact, some measures are taken to damp the process. One such measure is to have a hop counter contained in the header of each packet. This counter is decremented at each hop, with the packet being discarded when the counter reaches zero. Ideally, the hop counter should be initialized to the length of the path from source to destination. If the sender does not know how long the path is, it can initialize the counter to the worst case, namely, the full diameter of the subnet.

An alternative technique for damming the flood is to keep track of which packets have been flooded to avoid sending them out a second time. One way to perform this goal is to have the source node put a sequence number in each packet it receives from its hosts. Each node then needs a list per source node telling which sequence numbers originating at that source have been seen. If an incoming packet is on list, it is not flooded. To prevent the list from growing without bound, each list should be augmented by a counter, k, meaning that all sequence numbers through k have been seen. When a packet comes in, it is easy to check if the packet is a duplicate. If so, it is discarded. Furthermore, the full list below k is not needed, since k effectively summarizes it.

Flooding is an extremely robust technique and would be particularly suitable for military applications, where large numbers of nodes may be blown to bits at any instant. In distributed database applications, it is sometimes necessary to update all the databases concurrently, in which flooding can be useful. Flooding always chooses the shortest path, because it chooses all possible routes in parallel. Consequently, no other algorithm can produce a shorter delay [1, 4].

2.1.2 Dijkstra Routing Algorithm

Dijkstra routing algorithm is widely used in many applications because it is very simple and easy to understand. This algorithm finds the shortest paths from a source to all other nodes. To do this it requires global topological knowledge (the list of all nodes in the network and their interconnections, as well as the cost of each link).

In most general the weight of each link could be computed as a function of the distance, bandwidth, average traffic, communication cost, mean queue length, average delay, and other factors [6].

Let D(v) be the distance (sum of links weights along any path) from source node 1 to node v.

Let L (I,j) be the given cost between node I and node j.

There are then two parts to the algorithm: an initialization step, and a step to be repeated until the algorithm terminates:

1. Initialization

Set N ={1} . For each node v not in Set N,

set D(v) =L(1,v).

The value ∞ is used for nodes that are not connected to node 1; any number larger than maximum cost or distance in the network would suffice.

2. At each subsequent step

Find a node w not in N for which D(w) is a minimum and add w to N. Then update D(v) for all nodes remaining that are not in N by computing

$$D(v) = Min\,[D(v),\, D(w)+L(w,v)]$$

Step 2 is repeated until all nodes are in N.

The Dijkstra algorithm is used in most popular routing protocols because of its simplicity and efficiency.

2.1.3 Bifurcated Routing Algorithm

This algorithm suggests the existence of multiple routing paths for packets to flow between each source-destination pair. The following procedure is done by each node in the network

1. Each node has a table that kept all possible routes to all other nodes.
2. Each route has a weight value.
3. For each transfer, the originating node selects the appropriate link based on an evaluation function (this function is related to traffic status, channel bandwidth, and other factors) [5].

2.2 Adaptive Routing Algorithms

In these algorithms, routing decisions are changed to reflect any changes in topology and traffic. Adaptive algorithms differ in where they get their information (e.g., locally, from adjacent nodes, or from all nodes), when they change the routes (e.g., periodic, every load change, or when the topology changed), and the metric they used to optimize performance (e.g., distance, throughput, or estimated time delay). Three adaptive routing algorithms are summarized here [5,7,8].

2.2.1 Distance vector algorithm

In this algorithm, each node maintains a table (vector) giving the best known distance to each destination and which line to use. These tables are updated by exchanging the information with the neighbors. The distance vector algorithm is sometimes called the distributed Bellman-Ford routing algorithm.

In this algorithm, each node maintains a routing table indexed by, and contains one entry for each node in the network. This entry contains two parts: the preferred outgoing line to use for this destination, and an estimate of metric to that destination. This metric may be number of hops, time delay, total number of packets queued along the path, or something similar. In this algorithm, the end-to-end path is computed for each packet at all nodes. Also, in this algorithm, each node sends all information of routing table to only neighboring nodes [5].

2.2.2 Link State Algorithm

This algorithm works within the same basic framework that distance vector algorithm does to find the lowest cost path. However, Link-state algorithm works in a somewhat more localized manner. Each node maintains a routing table for only its direct neighbors. Also, each node sends small updates (a portion of the routing table) to all other nodes. Consequently, this algorithm converges more quickly [7].

The idea behind this algorithm is simple and can be stated as five parts, each node must:

1. Discover its neighbors and learn their network addresses.
2. Measure the cost or the delay to each of its neighbors.
3. Construct a packet telling all it has just learned.
4. Send this packet to all other nodes.
5. Compute the shortest path to every other node.

2.2.3 Baran's hot potato routing algorithm

In this algorithm, when any node receives any packet, it attempts to dispose of it as quickly as possible by replacing it in the shortest output queue. Also, when any message arrives, the node counts the number of packets awaiting transmission in each output path. It then attaches the new packet to the end of the shortest path queue, without regard to where that link leads [8].

3. The parameters affected Routing Algorithms

There are many factors that influence the performance of the routing algorithms [9,10]. They can be classified into three major categories:-

1. Source and destination nodes parameters.
2. Intermediate nodes parameters.
3. Network status parameters.

However, each of these parameters differently affects the routing algorithms. Also, each main category has a different influence degree on these routing algorithms. This influence may be linear, or exponential, or other forms. The abbreviations shown in table 2.1 are used to stand for each routing algorithm. The abbreviations shown in table 2.2 are used to stand for the influence of each parameter on routing algorithms.

3.1 Source and destination nodes parameters

The capabilities of both transmit and receiver nodes are very important factors that noticeably affected the performance of all routing algorithms. Of course, the powerful transmit and receiver nodes result in perfect network performance. The CPU status, buffering and RAM are the main parameters that belong to this category.

3.1.1 The CPU of source and destination nodes

The CPU capability and number of CPUs of both source and destination nodes are very effective factors that the performance of all routing algorithms noticeably affected with.

Of course, improving the CPU capability leads to a positive influence on routing algorithms. Also, multiple powerful CPUs results in a positive influence on the routing algorithms. The degree of influence of this parameter on R_i has an anti-linear proportion to the increase of both data bus width and CPU's clock speed. Therefore, the following function reasonably describes this effect:

$$P_1 = \frac{1}{\sum_{i=1}^{n} s_i * d_i} * w_1(R_i) \qquad (3.1)$$

Where:

n is the number of CPUs

d is data bus width in bytes.

s is CPU speed in MHZ

$w_1(R_i)$ *is* the weight value that represents the effect of this parameter on each R_i.

3.1.2. The buffering capability of both source and destination nodes

The buffers are used in both source and destination nodes to store either originating or received packets then processed by the node. Of course, large and variable buffer size results in a positive response on the routing performance. Routing algorithms perform more adequately when the buffer size becomes increasingly larger. Hereunder is the function that reasonably describes this effect:

$$P_2 = \left(\frac{1}{\log_2 B} + 1\right) * w_2(R_i) \qquad (3.2)$$

Where:

B is the buffer size in MB.

$w_2(R_i)$ is the weight value that represents the effect of this parameter on each R_i

3.1.3 RAM size

The RAM size is a very important factor that mainly affected the routing algorithm response. Routing algorithms perform more adequately when the memory size becomes increasingly larger. Hereunder is the function that reasonably describes this effect:

$$P_3 = \left(\frac{1}{\log_2 R/2} + 1\right) * w_3(R_i) \qquad (3.3)$$

Where:

R is the RAM size in MB

$w_3(R_i)$ is the weight value that represents the effect of this parameter on each R_i.

3.2 Intermediate nodes capabilities

The intermediate nodes in any traffic path are mainly affected not only the routing, but also the overall network performance. Number of intermediate nodes, their processing capabilities and their buffering status are considered as the main parameters.

3.2.1 Intermediate nodes processing capabilities

The ingoing and outgoing traffic at the intermediate nodes can be processed rapidly and efficiently, if these nodes have very powerful CPUs. This case is similar to P_1 and hereunder is the function that reasonably describes this effect:

$$P_4 = \frac{1}{s * d} * w_4(R_i) \qquad (3.4)$$

(

Where:

d is data bus width in bytes

s is CPU speed in MHZ.

$w_4(R_i)$ the weight value that represents the effect of this parameter on R_i.

3.2.2 Number of intermediate nodes

Increasing number of these nodes in any traffic path leads to increase the cost of this path. Of course, a large negative influence on performance of routing algorithm is detected. According to pervious studies, the function *m*ln m* is selected to measure this influence [11, 12]. The function describes this effect cab be written as:

$$P_5 = m * \ln m * w_5(R_i) \qquad (3.5)$$

Where

m is the average number of intermediate nodes at each link.

$w_5(Ri)$ is the weight value that represents the effect of this parameter on each R_i.

3.2.3 Buffering status

Buffer size of the intermediate nodes plays a very important role in managing packets. Using adequate large and variable size buffers is the best choice that leads to a perfect traffic management.

Like P_2, the function that reasonably describes this effect can be written as:

$$P_6 = (\frac{1}{\log_2 K} + 1)^* w_6(R_i) \qquad (3.6)$$

Where

K is the average normalized buffer size for all intermediate nodes in MB.

$w_6(R_i)$ is the weight value that represents the effect of this parameter on each R_i

3.2.4 Average intermediate nodes delay

This parameter relies on bandwidth of links, number of nodes and traffic flow. There are many models that can determine average time delay. M/M/1 model is considered the simplest and efficient model that can perform this task. It is very intuitive that the more average node delay in the network, the more negative performance of the routing algorithms. Hereunder is the function that reasonably describes this effect:

$$P_7 = \frac{1}{D} * w_8(R_i) \qquad (3.7)$$

Where

D is the average node delay in msec

$w_7(R_i)$ is the weight value that represents the effect of this parameter on each R_i

The average node delay (D) used in M/M/1 model is given by:

$$D = \frac{1}{\mu * C - \lambda}$$

Where:

C is the average bandwidth of the links in KB/sec

μ is the average service rate in msec.

λ is the average arrival rate in msec.

3.3 Network status parameters

The packet length, network topology and network traffic are the most common parameters belonging to this category. These factors are almost changed; consequently their influences appear obviously on adaptive algorithms.

3.3.1 Network traffic

The network traffic parameter is very difficult to be evaluated. Therefore, the average number of packets per second passing through a node can be used as measure of this parameter. According to statistics obtained from real cases [11,12], the function that is selected to measure this influence on the performance of routing algorithms:

$$P_8 = x * \log_2 x * w_8(R_i) \qquad (3.8)$$

Where:

x is the average packets passing through each node per second.

$w_8(R_i)$ is the weight value that represents the effect of this parameter on each R_i

3.3.2 Packet length

Increasing the size of packet affected negatively the performance of routing algorithms. Based on pervious cases, the function $\sqrt{l}$ represents this influence [13]. Hereunder the function that describes this influence:

$$P_9 = \sqrt{l} * w_9(R_i) \qquad (3.9)$$

Where:

l is the average packets passing through each node measured in bytes.

$w_9(R_i)$ is the weight value that represents the effect of this parameter on each R_i

3.3.3 Network topology

The main basic types of network topologies are star, ring, bus, tree and full connected. Consequently, all various types of large network are developed based on them. The complexity of network topology leads to more difficulty in determining the appropriate efficient routing algorithm for this network [5].Of course, the adaptive routing algorithms are more sensitive to this parameter. The degrees of influence are very low (V.L), low (L), Fair (F), or high (H). Table 3.1 defines the degree of the influence for each topology on each routing algorithm [9].

Hereunder is the function that reasonably describes this effect:

$$P_{10} = I(T)^* w_{10}(R_i) \qquad (3.10)$$

Where:

I(T) is a constant value that express the influence of topology of network

$w_{10}(R_i)$ is the weight value that represents the effect of this parameter on each R_i

The following assumptions are used: V.L = 1, L = 4, M=7, and H=10.

3.4 Estimating a cost function Φ (R_i)

The pervious parameters described before exert an influence on the performance of routing algorithms. A cost function is considered for each algorithm that is mainly affected by some or all parameters mentioned. The cost function Φ (R_i) is expressed as follows:

$$(R_i) = f(P_1(R_i), P_2(R_i), \ldots\ldots\ldots P_{10}(R_i)) \quad (3.11)$$

4. Results and discussion

Table 4.1 shows 8 different cases that represent source and destination parameters, intermediate nodes parameters and network status parameters different network parameters.

The weight values that represent the influence of each parameter on the routing algorithms are shown in table 4.2 [14]. Considering these weight values, the influence functions P_j, $1 \leq j \leq 10$, and Φ (R_i) are computed for each routing algorithm.

4.1 Computing P_j

Using equation 3.1, table 4.1 and table 4.2, the influence function of CPU of source and destination nodes P_1 can be computed, the results are illustrated in table 4.3. It is noticed that the influence of P_1 is relatively small on all R_i.

Using equation 3.2, table 4.1 and table 4.2, the influence function of the buffering capability of both source and destination nodes P_2 is computed. Table 4.4 illustrates the results. It is noticed from the results that the influence of P_2 on both R_2 and R_3 are small relative to other R_i.

Using equation 3.3, table 4.1 and table 4.2, the influence function of RAM size of source and destination nodes P_3 is computed. It is noticed from results illustrated in table 4.5 that the effect of P_3 is relatively moderate for all R_i and R_2 is less sensitive to P_3, while R_1 and R_5 are more sensitive to P_3.

Using equation 3.4, table 4.1 and table 4.2, the influence function of CPU processing capability of intermediate nodes P_4 is computed. It is noticed from the results shown in table 4.6 that the effect of P_4 is relatively small on all R_i.

Using equation 3.5, table 4.1 and table 4.2, the influence function of average number of intermediate nodes P_5 is computed. The results are shown in table 4.7. It is noticed that the influence of P_5 is relatively high on all R_i and R_2 is less sensitive to P_5, while R_1 is more sensitive to P_5.

Using equation 3.6, table 4.1 and table 4.2, the influence function of buffering status of intermediate nodes P_6 is computed. From the results illustrated in table 4.8, it is noticed that the effect of P_6 is moderate on all R_i.

Using equation 3.7, table 4.1 and table 4.2, the average node delay of intermediate nodes P_7 is computed. Table 4.9 illustrates these results. It is found that P_7 has the most extreme effect on all R_i. R_2, R_3 and R_6 are affected by P_7 with small degree relative to R_1, R_4 and R_5.

Using equation 3.8, table 4.1 and table 4.2, the influence function of network traffic P_8 is computed and the results are in table 4.10. It is noticed from the results that the affect of P_8 is relatively high on all R_i.

Using equation 3.9, table 4.1 and table 4.2, the influence function of network traffic P_9 is computed the results are illustrated in table 4.11. It is noticed that the effect of P_9 is moderate and R_2 and R_3 are less sensitive to P_9 than other R_i.

Using equation 3.10, table 4.1 and table 4.2, the influence function of network topology P_{10} is computed and the results are illustrated in table 4.12. It is noticed from the results the effect of P_{10} is moderate on all R_i

4.2 Estimating Φ (R_i) using only source and destination nodes effect

In this case, only the source and destination nodes effect is considered and other parameters are neglected. By applying this assumption in equation 3.11 and taking the sum as the appropriate function, Φ (R_i) is calculated as:

$$\Phi\ (R_i) = \sum_{j=1}^{3} P_j(R_i) \quad (4.1)$$

Φ(R_i) values for the eight different cases are computed using equation 4.1 and tables 4.3, 4,4 and 4.5. Table 4.13 illustrates the results. It is noticed from the results that R_2 has a small cost relative to other R_i.

4.3 Estimating Φ (R_i) using only intermediate nodes effect

In this case, the intermediate nodes effect is considered and other parameters are neglected. By applying this assumption in equation 3.11 and taking the sum as the appropriate function, Φ (R_i) is calculated as:

$$\Phi\ (R_i) = \sum_{j=4}^{7} P_j(R_i) \quad (4.2)$$

$\Phi(R_l)$ values for the eight different cases are computed using equation 4.2 and tables 4.6, 4,7, 4.8 and 4.9. The results are illustrated in table 4.14. It is found that R_2 has a small cost relative to other R_l and R_1 has the higher cost value.

4.4 Estimating Φ (R_l) using only network parameters effect

In this case, only the network parameters effect is considered and other parameters are neglected. By applying this assumption in equation 3.11 and taking the sum as the appropriate function, $\Phi(R_l)$ is calculated as:

$$\Phi\,(R_l) = \sum_{j=8}^{10} P_j(R_l) \qquad (4.3)$$

$\Phi(R_l)$ values for the eight different cases are computed using equation 4.3 and tables 4.10 4,11 and 4.12. It is found from the results illustrated in table 4.15 that R_2 has a small cost relative to other R_l and R_1 has the higher cost value.

4.4 Overall network performance evaluation

Considering all categories effect in estimating the cost function Φ (R_l) with equal weight, Φ (R_l) is given by:

$$\Phi\,(R_l) = \sum_{j=1}^{10} P_j(R_l) \qquad (4.4)$$

(

The average cost and performance degree of all R_l are calculated. Table 4.16 illustrates these results. It is found from these results that R_l are highly effected by intermediate nodes parameters, consequently, the estimated $\Phi(R_l)$ based on these parameters is the higher. On the other side, it is found that the source and destination node parameters has the lowest effect on is R_l. Also, it is noticed that R_2 is considered to be the best routing algorithm based on all evaluations, while, R_1 is the worst for the eight cases. Table 4.16 shows the arrangement of the routing algorithms according their effects.

5. Conclusions

The performance of both adaptive and non-adaptive routing algorithms is significantly affected by all network parameters with different degrees. Routing algorithms are very sensitive to intermediate nodes parameters especially the average number of nodes and average node delay relative to other ones. While source and destination nodes parameters are less effective ones.

The selection of the appropriate routing algorithm for a certain network is based on an estimated function Φ (R_l) and performance levels, therefore, it is strongly recommended to use the Dijkstra algorithm in case the network traffic is relatively predictable and all network parameters are fixed. While Baran's hot potato routing algorithm is the appropriate choice in case of the repetitive variations in topology and the traffic status.

Network performance is mainly improved by selecting the appropriate network parameters, consequently, it is highly recommended for any user to

- Determine the network parameters.
- Analyze and evaluate the influence function for each parameter on each routing algorithm.
- Estimate a cost function based on the current network status.
- Compare and select the appropriate algorithm regarding to his network conditions.

6. References

1. M. Pioro, D.Medhe," Routing, Flow, and Capacity Design in Communication and Computer Networks", Morgan Kauffman series, 2004.
2. D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and principles of Internet traffic engineering", Internet RFC 3272, May 2002.
3. G. Trangmoe, "A comparative study of dynamic routing circuit-switched networks", University of Arizona, 1995
4. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin," Network Flows: Theory, Algorithms, and Applications", Prentice Hall, 1993.
5. M. Schwartz, "Telecommunication Networks Protocols, Modeling and Analysis", Addison-Wesley Publishing Company, 1989.
6. Anel A. Montes, " Network shortest path application for optimum track ship routing", California, 2005
7. S. F. Wu, F.-Y. Wang, Y. F. Jou, and F. Gong., "Intrusion detection for link-state routing protocols", In IEEE Symposium on Security and Privacy, 1997.
8. Michael T. Goodrich, "Efficient and Secure Network Routing Algorithms", Johns Hopkins University, 2001.
9. S.Yang, "Performance evaluation of routing algorithms under various network configuration parameters", Int. J. Network Mgmt., 183-197, 1997.

10. A. Ahmed, "Data Communication Principles for fixed and wireless networks", Kluwer Academic Publishers, Chapter 8-10, 2003.
11. G. L. Li and P. W. Dowd , " An analysis of network performance degration induced by workload fluctuations", IEEE/ACM Trans. on networking, 3, No. 4, 433-440,1995.
12. M.A. Mouhamed and A. A. Maasarani, " Performance evaluation of scheduling procedure constrained computations on message-passing systems", IEEE Trans. on Parallel & Distributed Systems, 5, No. 12, 1317-1322, 1994.
13. A. Kershnbaum, "Telecommunication Network Design Algorithms", McGraw-Hill, New York, No. 302, 488-524,1993.
14. D. Saha and A. Mukherjee, "Computational analysis of a routing strategy in a hierarchical computer communication network", Computer Communications journal, 18, No. 7, 507-511, 1995.

Table 2.1 Routing algorithms abbreviations

Abbreviation	Routing algorithm
R_1	The flooding routing algorithm
R_2	Dijkstra routing algorithm
R_3	Bifurcated routing algorithm
R_4	Distance vector algorithm
R_5	Link state algorithm
R_6	Baran's hot potato routing algorithm

Table 2.2 the influence of network parameters abbreviations

Abbreviation	Stand for
P_1	The influence of the CPU capability of source and destination nodes on R_i
P_2	The influence of buffering capability of the source and destination nodes on R_i
P_3	The influence of RAM size of source and destination nodes on R_i
P_4	The influence of the CPU of the intermediate nodes on R_i
P_5	The influence of the average number of the intermediate nodes on R_i
P_6	The influence of the buffering status of the intermediate nodes on R_i
P_7	The influence of the average node delay of the intermediate nodes on R_i
P_8	The influence of the traffic status on R_i
P_9	The influence of packet length on R_i
P_{10}	The influence of network topology on R_i

Table 3.1.The influence of network topology on the routing algorithms

Routing algorithm	Tree	Ring	Bus	star	full
R_1	L	V.L	L	V.L	L
R_2	L	V.L	L	V.L	L
R_3	L	V.L	L	V.L	L
R_4	M	L	H	L	H
R_5	M	L	H	L	H
R_6	M	L	H	L	H

Table 4.1 different network parameters used for evaluating the routing R_i

case#	Source and destination nodes parameters					Intermediate nodes parameters							Network parameters		
	CPU			B	RAM R	CPU		No of nodes	K	Node delay			x	L	topology
	N	s	d			s	d			μ	λ	C			
1	1	66	32	10	16	5	32	12	3	8	5	64	4	32	tree
2	1	100	32	15	32	10	32	15	4	9	6	100	8	48	bus
3	2	120 50	64 32	20	64	33	64	18	5	10	7	128	16	56	full
4	2	133 66	64 64	25	128	66	64	20	8	11	8	156	20	64	ring
5	2	266 100	64 64	32	256	133	64	25	10	12	8	200	32	100	bus
6	2	512 133	64 64	50	512	266	128	30	12	14	9	256	40	128	star
7	2	512 266	128 128	64	1024	512	128	32	15	15	10	512	64	150	full
8	2	512 512	128 128	128	2048	760	128	40	20	16	11	600	80	200	ring

Table 4.2 weight values for each parameter and routing algorithms

	R_1	R_2	R_3	R_4	R_5	R_6
P_1	9	3	5	6	8	7
P_2	7	1	3	5	5	6
P_3	7	1	3	5	8	6
P_4	7	2	3	7	7	9
P_5	8	1	5	7	5	6
P_6	6	1	3	6	5	9
P_7	8	1	1	5	4	1
P_8	10	8	6	5	5	6
P_9	10	2	2	7	6	6
P_{10}	10	7	8	4	4	3

Table 4.3 P_1 values versus R_i

case#	$P_1(R_1)$	$P_1(R_2)$	$P_1(R_3)$	$P_1(R_4)$	$P_1(R_5)$	$P_1(R_6)$
1	0.034091	0.011364	0.018939	0.022727	0.030303	0.026515
2	0.0225	0.0075	0.0125	0.015	0.02	0.0175
3	0.007759	0.002586	0.00431	0.005172	0.006897	0.006034
4	0.005653	0.001884	0.003141	0.003769	0.005025	0.004397
5	0.003074	0.001025	0.001708	0.002049	0.002732	0.002391
6	0.001744	0.000581	0.000969	0.001163	0.00155	0.001357
7	0.000723	0.000241	0.000402	0.000482	0.000643	0.000562
8	0.000549	0.000183	0.000305	0.000366	0.000488	0.000427

Table 4.4 P_2 values versus R_i

Case #	$P_2(R_1)$	$P_2(R_2)$	$P_2(R_3)$	$P_2(R_4)$	$P_2(R_5)$	$P_2(R_6)$
1	9.10721	1.109803	3.90309	6.50515	6.50515	7.80618
2	8.791706	1.113744	3.767874	6.27979	6.27979	7.535748
3	8.619647	1.116014	3.694135	6.156891	6.156891	7.388269
4	8.507368	1.117545	3.646015	6.076691	6.076691	7.29203
5	8.4	1.119048	3.6	6	6	7.2
6	8.240287	1.121355	3.531551	5.885919	5.885919	7.063103
7	8.166667	1.122449	3.5	5.833333	5.833333	7
8	8	1.125	3.428571	5.714286	5.714286	6.857143

Table 4.5 P_3 values versus R_i

Case #	$P_3(R_1)$	$P_3(R_2)$	$P_3(R_3)$	$P_3(R_4)$	$P_3(R_5)$	$P_3(R_6)$
1	9.333333	1.333333	4	6.666667	10.66667	8
2	8.75	1.25	3.75	6.25	10	7.5
3	8.4	1.2	3.6	6	9.6	7.2
4	8.166667	1.166667	3.5	5.833333	9.333333	7
5	8	1.142857	3.428571	5.714286	9.142857	6.857143
6	7.875	1.125	3.375	5.625	9	6.75
7	7.777778	1.111111	3.333333	5.555556	8.888889	6.666667
8	7.7	1.1	3.3	5.5	8.8	6.6

Table 4.6 P_4 versus R_i

Case #	$P_4(R_1)$	$P_4(R_2)$	$P_4(R_3)$	$P_4(R_4)$	$P_4(R_5)$	$P_4(R_6)$
1	0.35	0.1	0.15	0.35	0.35	0.45
2	0.175	0.05	0.075	0.175	0.175	0.225
3	0.026515	0.007576	0.011364	0.026515	0.026515	0.034091
4	0.013258	0.003788	0.005682	0.013258	0.013258	0.017045
5	0.006579	0.00188	0.00282	0.006579	0.006579	0.008459
6	0.001645	0.00047	0.000705	0.001645	0.001645	0.002115
7	0.000854	0.000244	0.000366	0.000854	0.000854	0.001099
8	0.000576	0.000164	0.000247	0.000576	0.000576	0.00074

Table 4.7 P_5 versus R_i

Case #	$P_5(R_1)$	$P_5(R_2)$	$P_5(R_3)$	$P_5(R_4)$	$P_5(R_5)$	$P_5(R_6)$
1	324.966	40.62075	203.1038	284.3453	203.1038	243.7245
2	479.3172	59.91465	299.5732	419.4025	299.5732	359.4879
3	887.2284	110.9035	554.5177	776.3248	554.5177	665.4213
4	1180.441	147.5552	737.7759	1032.886	737.7759	885.3311
5	1564.809	195.6012	978.0058	1369.208	978.0058	1173.607
6	1965.285	245.6607	1228.303	1719.625	1228.303	1473.964
7	2379.157	297.3947	1486.973	2081.763	1486.973	1784.368
8	2590.493	323.8116	1619.058	2266.681	1619.058	1942.87

Table 4.8 P_6 versus R_i

Case #	$P_6(R_1)$	$P_6(R_2)$	$P_6(R_3)$	$P_6(R_4)$	$P_6(R_5)$	$P_6(R_6)$
1	7.673658	1.278943	3.836829	7.673658	6.394715	11.51049
2	7.5	1.25	3.75	7.5	6.25	11.25
3	7.388269	1.231378	3.694135	7.388269	6.156891	11.0824
4	7.2	1.2	3.6	7.2	6	10.8
5	7.127411	1.187902	3.563705	7.127411	5.939509	10.69112
6	7.074313	1.179052	3.537157	7.074313	5.895261	10.61147
7	7.015763	1.169294	3.507881	7.015763	5.846469	10.52364
8	6.949078	1.15818	3.474539	6.949078	5.790898	10.42362

Table 4.9 P_7 versus R_i

Case #	$P_7(R_1)$	$P_7(R_2)$	$P_7(R_3)$	$P_7(R_4)$	$P_7(R_5)$	$P_7(R_6)$
1	1752	219	219	1095	876	219
2	2256	282	282	1410	1128	282
3	3144	393	393	1965	1572	393
4	4336	542	542	2710	2168	542
5	5312	664	664	3320	2656	664
6	7096	887	887	4435	3548	887
7	15280	1910	1910	9550	7640	1910
8	32680	4085	4085	20425	16340	4085

Table 4.10 P_8 versus R_i

Case #	$P_8(R_1)$	$P_8(R_2)$	$P_8(R_3)$	$P_8(R_4)$	$P_8(R_5)$	$P_8(R_6)$
1	80	64	48	40	40	48
2	240	192	144	120	120	144
3	640	512	384	320	320	384
4	864.3856	691.5085	518.6314	432.1928	432.1928	518.6314
5	1600	1280	960	800	800	960
6	2128.771	1703.017	1277.263	1064.386	1064.386	1277.263
7	3840	3072	2304	1920	1920	2304
8	5057.542	4046.034	3034.525	2528.771	2528.771	3034.525

Table 4.11 P_8 versus R_i

Case #	$P_9(R_1)$	$P_9(R_2)$	$P_9(R_3)$	$P_9(R_4)$	$P_9(R_5)$	$P_9(R_6)$
1	56.56854	11.31371	11.31371	39.59798	33.94113	33.94113
2	69.28203	13.85641	13.85641	48.49742	41.56922	41.56922
3	74.83315	14.96663	14.96663	52.3832	44.89989	44.89989
4	80	16	16	56	48	48
5	100	20	20	70	60	60
6	113.1371	22.62742	22.62742	79.19596	67.88225	67.88225
7	122.4745	24.4949	24.4949	85.73214	73.48469	73.48469
8	141.4214	28.28427	28.28427	98.99495	84.85281	84.85281

Table 4.12 P_8 versus R_i

Case #	$P_{10}(R_1)$	$P_{10}(R_2)$	$P_{10}(R_3)$	$P_{10}(R_4)$	$P_{10}(R_5)$	$P_{10}(R_6)$
1	40	28	32	28	28	21
2	40	28	32	40	40	30
3	40	28	32	40	40	30
4	10	7	8	16	16	12
5	40	28	32	40	40	30
6	10	7	8	16	16	12
7	40	28	32	40	40	30
8	10	7	8	16	16	12

Table 4.13 Φ (R_i) values as a function of source and destination nodes parameters

Case #	Φ (R_1)	Φ (R_2)	Φ (R_3)	Φ (R_4)	Φ (R_5)	Φ (R_6)
1	18.47463	2.4545	7.922029	13.19454	17.20212	15.8327
2	17.56421	2.371244	7.530374	12.54479	16.29979	15.05325
3	17.02741	2.3186	7.298445	12.16206	15.76379	14.5943
4	16.67969	2.286096	7.149156	11.91379	15.41505	14.29643
5	16.40307	2.262929	7.030279	11.71633	15.14559	14.05953
6	16.11703	2.246936	6.90752	11.51208	14.88747	13.81446
7	15.94517	2.233801	6.833735	11.38937	14.72286	13.66723
8	15.70055	2.225183	6.728877	11.21465	14.51477	13.45757

Table 4.14 Φ (R_i) values as a function of intermediate nodes parameters

Case #	Φ (R_1)	Φ (R_2)	Φ (R_3)	Φ (R_4)	Φ (R_5)	Φ (R_6)
1	2084.99	260.9997	426.0906	1387.369	1085.848	474.685
2	2742.992	343.2146	585.3982	1837.078	1433.998	652.9629
3	4038.643	505.1425	951.2232	2748.74	2132.701	1069.538
4	5523.654	690.759	1283.382	3750.1	2911.789	1438.148
5	6883.943	860.7909	1645.572	4696.342	3639.952	1848.306
6	9068.361	1133.84	2118.841	6161.701	4782.2	2371.578
7	17666.17	2208.564	3400.482	11638.78	9132.821	3704.893
8	35277.44	4409.97	5707.533	22698.63	17964.85	6038.294

Table 4.15 Φ (R_i) values as function of network parameters

Case #	Φ (R_1)	Φ (R_2)	Φ (R_3)	Φ (R_4)	Φ (R_5)	Φ (R_6)
1	176.5685	103.3137	91.31371	107.598	101.9411	102.9411
2	349.282	233.8564	189.8564	208.4974	201.5692	215.5692
3	754.8331	554.9666	430.9666	412.3832	404.8999	458.8999
4	954.3856	714.5085	542.6314	504.1928	496.1928	578.6314
5	1740	1328	1012	910	900	1050
6	2251.908	1732.644	1307.89	1159.582	1148.268	1357.145
7	4002.474	3124.495	2360.495	2045.732	2033.485	2407.485
8	5208.964	4081.318	3070.81	2643.766	2629.624	3131.378

Table 4.16 Φ (R_l) and performance degree for routing algorithms

	Evaluation using first category		Evaluation using second category		Evaluation using third category		Overall evaluation	
	Average Φ (R_l	degree	Average Φ (R_l	degree	Average Φ (R_l	degre e	Average Φ (R_l	degree
R_1	16.73	6	10410	6	1929	6	12355.73	6
R_2	2.29	1	1301	1	1484	5	2787.29	1
R_3	7.17	2	2014	2	1125	3	3146.17	2
R_4	11.95	3	6864	5	9989	2	16864.95	5
R_5	15.49	5	5385	4	9889	1	15289.49	4
R_6	14.34	4	2200	3	1162	4	3376.34	3