
IP 네트워크에서 트래픽 레코드를 이용한 최단 거리 결정 기법

홍성혁*

Determining the shortest paths by using the history of IP network traffic records

Sunghyuck Hong*

요 약 본 연구는 정적 라우팅과 동적 라우팅의 장점을 활용하여 최단 거리를 산정하는 하이브리드 방식의 라우팅 기법이다. 네트워크는 휴먼 팩터에 의해 결정이 되기 때문에, 네트워크 트래픽은 항상 특정 시간에 몰리게 되고 네트워크 트래픽 레코드를 분석하면 각 시간대별로 네트워크 트래픽 추측이 가능하다. 따라서 RIP의 잦은 정보 교환으로 인해 발생하는 부하를 줄여 IP 네트워크에서 최단 경로를 산정하는 것이 이 논문이 목적이다.

주제어 : 정적 라우팅, 네트워크 트래픽, 최단 경로, 결정적 라우팅, 역사 기반 라우팅, 효율적인 라우팅.

Abstract There are many algorithms to improve the network traffic and to avoid losing packets in the network. This paper proposed determining the shortest paths for hops, which are in the middle of the source and destination. The shortest path in this paper means the fastest path between them. Recently, dynamic routing algorithm is currently used now but this paper suggests the fastest paths between the source and the destination is by using the record of the network traffic history. People are using the networking and the network traffic is always corresponding to how many people use the networking in specific time. Therefore, I can predict the network condition by referring to the history of network traffic record, and then the shortest path can be produced without using RIP too much. It will be helpful to improve the network traffic.

Key Words : Static routing, network traffic, the shortest paths, deterministic routing, history-based routing, efficient routing.

1. Introduction

The Internet began with ARPANET and just e-mail triggered rapid growth in ARPANET, so the Web has triggered explosive growth in the Internet. Today, there are 100 million hosts connected to Internet networks; many users are in the 100s of millions. Hundreds of countries access the Internet and the growth of Internet hosts and users are still increasing dramatically [8]. The growth of Internet hosts and users is much faster than the growth of network traffic

speed because of too many users. When we travel on the Internet, the network routers have to find out the path from source to destination and give the shortest path to users for destination. If we find the shortest path between them, we could get the result faster, unless it takes somewhat longer than the previous case.

Determining the shortest path (the fastest path) for hops is important to communicate with destination and source. According to *Roch Guerin ,and Ariel Orda* [1], finding minimum hops for all possible paths is the most efficient in order to get the shortest path in a certain

*백석대학교 정보통신학부 조교수 Sunghyuck.hong@gmail.com

논문접수: 2012년 4월 30일, 1차 수정을 거쳐, 심사완료: 2012년 5월 11일

network. However, the path weight has to be considered additive and bottleneck weights.

To get the shortest path routing, use Dijkstra's shortest path algorithm or Bellman - Ford algorithm, which is widely used for finding the shortest algorithm [2]. The shortest path algorithm is based on the graph theory. There are many algorithms to find the shortest path but they always pick the same path if their algorithms are correct. The only difference is the efficiency.

Let's consider the sending and receiving packets mechanism. Whenever senders send packets to a destination, they consider the shortest paths between them to reduce the cost and estimated time to delivery. To get the best shortest path, they could use RIP (Routing Information Protocol) that sends routing-update messages at regular intervals when the network topology changes [8]. Dynamic routing helps to find the shortest path. When a router receives a routing update that includes changes to an entry, it updates its routing table to reflect the new route. RIP routers maintain only the best route (the route with the lowest metric value) to a destination. After updating its routing table, the router immediately begins transmitting routing updates to inform other network routers of the change. These updates are sent independently of the regularly scheduled updates that RIP routers send [3].

That is the current scheme for a dynamic routing algorithm to find the shortest paths. However, this paper introduces the new scheme for the static routing path to determine the short path without using too much RIP. All routers update their routing table regularly. However, if routers use the history of networking traffic information in specific time in each routing table and store it, it is then possible to send packets from source to destination without any referencing RIP.

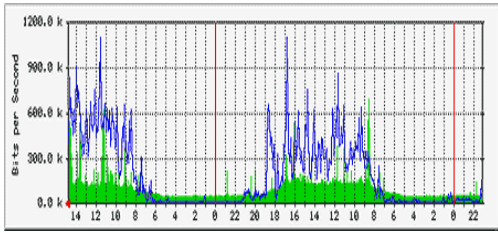
Therefore, the router sends packets to other destinations without exchanging too much unnecessary routing information, sending, receiving and updating the routing table. It maybe contributes to reduced

network traffic and reduced load of routing table updating processes because whether or not there are packets on the network, RIP information is exchanged automatically based on its timer. As shown in Figure 1, it is network traffic information for each time in a day. There are pick times and slow down periods in Figure 1. If many records of traffic history are collected as much as it can, the future network traffic will be able to be predicted without regularly using too much other routing information.

Also, this new scheme would use prediction theory, so it can forecast the future network traffic condition that is based on the history of the network traffic table for each time.

Network traffic depends on time and how many clients are connected in the local area network. Thus, it can be forecasted network traffic if there are many networking traffic information table in each day. Look at the network configuration, figure 2 and the network traffic information table, figure 2. If a mathematically predicted theory is used, it would be helpful to determine the shortest path for each time.

Therefore, it could deliver packets from source to destination in order to get the expected fast path without any too much routing information exchanging. It can also support routing algorithm and reduce network traffic. There must be peak time and idle time in a network path. Normally, weekdays in the morning and afternoon are peak time for networks and slow down is a little after business hours with a small increase at night between 8 to 11 pm. The pattern is quit predictable. Usually, network traffic factors could affect whole network traffic condition unexpectedly. However, it is not quite different between dynamic routing and static routing. Some packets may run fast and correct but some packets may slow down or sometimes be lost.



[Figure 1] Provided by CyberLynk Network, Inc

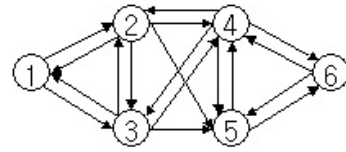
So overall speed of all packets represents as network speed. This paper focuses on increasing the overall speed in communication networks. The current routing algorithm is trying to increase packet speed by using a dynamic routing table that communicate to each router and update routing table to find out the fast path in each network. However, those kinds of efforts don't make significant change in networking traffic situations in comparison to static routing algorithm and sometime it could make network traffic worse because of exchanging unnecessary routing information and updating the routing table regularly with each router in the network.

The paper expected that the static routing algorithm that is based on the history of the network traffic information and using mathematical prediction theory in order to help determine the fast path would be better than the dynamic routing algorithm.

2. System model

Here is a network model, figure 2. Suppose node 1 is the source and node 6 is the destination. Each node has a routing table and updates its routing table regularly.

Also, they exchange RIP (Routing Information Protocol) every 60 seconds and after 180 seconds, if no response is detected, they regard as it is not connected. [8]. Figure 2 shows 1 hop network cost in 10:00 am, so calculating the shortest path between node 1 and node 6 depends on the network traffic table



[Figure 2] Network Model

It was actually tested in Castalia which is sensor network simulator tool. 6 network nodes are setup and given a weighting value (time spending) on each node to calculate network cost.

Date	Time	Node 1 to 2 weight value	Node 1 to 3	Node 2 to 3	...
4/1	10:00	1.8ms	1.9ms	1.8ms	...
4/1	10:10	1.9ms	1.8ms	1.0ms	...
4/1	10:20	2.0ms	2.1ms	2.0ms	...
4/1	10:30	2.1ms	2.1ms	2.0ms	...
4/1	10:40	2.2ms	2.0ms	2.2ms	...
4/1	10:50	2.2ms	2.2ms	2.5ms	...
4/1	11:00	2.3ms	2.5ms	2.5ms	...

[Figure 3] Network cost

Time (10:00)	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6
Node 1	-	1.9 (1.8)	2.1 (2.0)	-	-	-
Node 2	1.8 (1.9)	-	3(2.9)	3.1(3.2)	2.8 (2.7)	-
Node 3	1.5 (1.4)	2.8 (2.6)	-	2.9(3.0)	2.5 (2.5)	-
Node 4	-	1.8 (1.6)	1.6 (1.5)	-	1.7 (1.8)	-
Node 5	-	1.4 (1.3)	1.1 (1.2)	1.6 (1.4)	-	1.9 (2.0)
Node 6	-	-	-	2.3 (2.1)	2.5 (2.3)	-

*Figure 3, the values of inside are actual network speed.

*Each weight value is an average of everyday in specific time

[Figure 4] Network Cost

Time	The shortest path between node 1 and node 6	
	The shortest Path	Total time to reach
10:00	1 - 2 - 5 - 6	6.7ms
10:10	1 - 2 - 5 - 6	6ms
10:20	1 - 3 - 4 - 6	7ms
...
14:00	1 - 3 - 4 - 6	10ms
14:10	1 - 3 - 5 - 6	14ms
...
20:00	1 - 3 - 4 - 6	11ms
20:10	1 - 2 - 4 - 6	12ms
...

[Figure 5] The Elapsed Time for the Shortest Path

n	x	y	x ²	y ²	xy
1	1	1.8	1	3.24	1.8
2	2	1.9	4	3.61	3.8
3	3	2.0	9	4	6
4	4	2.1	16	4.41	8.4
5	5	2.2	25	4.84	11
6	6	2.2	36	4.84	13.2
7	7	2.3	49	5.29	16.1
Sum	28	14.5	140.0	30.2	60.3
Average	4	2.1	20.0	4.3	8.6

[Figure 6]

This paper can simulate to determine the shortest path based on the history of network traffic records and also it will show the difference between the actual the shortest path and calculate the shortest path. Figure 3 shows one hop network weight between each node at 10:00 am. The weight value between two nodes is the prediction value of weight at 10:00 am which means it collects records of weight values at every 10:00 am in during the week. Figure 4 shows how the prediction value has been calculated.

Therefore, according to Figure 4, the shortest path between node 1 and node 6 is going to be 1-2-5-6. The total cost is 6.7 ms. That is the expected cost, but it is can be a different current dynamic algorithm because the network condition always depends on how many users are using the network or how many packets are currently delivering on the network path. In fact the

actual shortest path may be different from this path, but it can be ignored because that is not going to be a big difference.

Suppose it sends packets on the network like figure 1-1, and we calculate the network cost such as when sending, receiving packets and exchanging RIP, updating routing table in every 60 seconds. New system model sends RIP information in every 10 minutes, and then updates the routing table and network traffic table when it gets RIP from its neighbor node.

Now, it is time to predict a certain time of network traffic information by using the mathematical method such as the linear least squares fitting technique because we have only a 10 minutes period between routing information. The linear least squares fitting technique [7],[4],[6],[5] is the simplest and most commonly applied form of linear regression and provides a solution to the problem of finding the best fitting straight line through a set of points if the relation between time and network traffic data are a line. In this case suppose time and network traffic are supposed to be a correlation from 10:00 am to 11:00. RIP information can be exchanged every 10 minutes, thus the routing table also updates every 10 minutes like figure 3. Therefore, if the source sends a packet to destination at 10:05, the network traffic needs to be calculated by the linear least squares fitting technique. Because we knew the time and network traffic but were not certain about the time's network traffic information, the network traffic at 10:05 could be measured by using another data set. 10:05 is in range of figure 4, so it can be calculated.

Let x be the time variable and y be a network traffic weight value.

Suppose that the data points are $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, where x is the independent variable and y is the dependent variable. The fitting curve $f(x)$ has the deviation (error) d from each data point, i.e. , $d_1=y_1-f(x_1), d_2=y_2-f(x_2), \dots, d_n=y_n-f(x_n)$. According to the method of least squares, the best fitting curve has

the property that:

$$\begin{aligned} \Pi &= d_1^2 + d_2^2 + \dots + d_n^2 = \sum_{i=1}^n d_i^2 = \\ &= \sum_{i=1}^n [y_i - f(x_i)]^2 = a \text{ minimum} \end{aligned}$$

In this case, it is supposed to have the linear relationship between time and network traffic from 10:00 to 11:00, so we can apply this to the least-squares line uses a straight line.

$y = a + bx$ (the linear relationship of time and network traffic)

to approximate the given set of data, $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, where $n \geq 2$. The best fitting curve $f(x)$ has the least square error, i.e.,

$$\Pi = \sum_{i=1}^n [y_i - f(x_i)]^2 = \sum_{i=1}^n [y_i - (a + bx_i)]^2 = \min$$

Please note that a and b are unknown coefficients while all x_i and y_i are given. To obtain the least square error, the unknown coefficients a and b must yield zero first derivatives.

$$\begin{cases} \frac{\delta \Pi}{\delta a} = 2 \sum_{i=1}^n [y_i - (a + bx_i)] = 0 \\ \frac{\delta \Pi}{\delta b} = 2 \sum_{i=1}^n x_i [y_i - (a + bx_i)] = 0 \end{cases}$$

Expanding the above equations, we have:

$$\begin{cases} \sum_{i=1}^n y_i = a \sum_{i=1}^n 1 + b \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i y_i = a \sum_{i=1}^n x_i + b \sum_{i=1}^n x_i^2 \end{cases}$$

The unknown coefficients a and b can therefore be obtained:

$$\begin{cases} a = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n \sum x^2 - (\sum x)^2} \\ b = \frac{n \sum xy - (\sum x)(\sum y)}{n \sum x^2 - (\sum x)^2} \end{cases}$$

where $\sum \dots$ stands for $\sum_{i=1}^n \dots i$.

Therefore, we get y line for time and network, so we can predict the network traffic information at 10:05 by using this method. We can get the results of $x, y, x^2,$

$y^2,$ and xy .

Figure 4 is based on Figure 3 for node 1 to node 2 only.

$$a = (14.5 \times 140 - (28 \times 60.3)) / ((7 \times 140) - 28 \times 28) = 1.7429$$

$$b = (7 \times 60.3 - 28 \times 14.5) / (7 \times 140 - 28 \times 28) = 0.0821$$

$$\text{Thus, } y = 1.7429 + 0.0821x$$

If it gets network traffic at 10:05, then it assigns x value as 1.5. 10:00 is assigned to 1, so 10:05 is between 10:00 and 10:10, so it will be 1.5. That means x is equal to 1.5, y is equal to $1.7429 + 0.0821 \times 1.5 = 1.8661$. According to Figure 3, this result sounds acceptable.

3. CONCLUSION

This paper focused on reducing unnecessary information and improvement of network traffic by using the linear least squares fitting technique, which is required to the linear relationship between factors, but fortunately in this case, I supposed that there is a linear relationship between time and network traffic, so it would be applied to this case. It should require that applying n^{th} Degree Polynomials will be the future work in this paper.

Reference

- [1] Azaron, Amir; Kianfar, Farhad (2003). Dynamic shortest path in stochastic dynamic networks: Ship routing problem, European journal of operational research. 144(1), 138
- [2] Bhide, Nilesh M; Sivalingam, Krishna M; Fabry-Asztalos (2001). Routing Mechanisms Employing Adaptive Weight Functions for Shortest Path Routing in Optical WDM Networks, 3(3), 227-236.
- [3] C. Hedrick, "Routing Information Protocol (1988). Rutgers University, June, <http://www.ietf.org/rfc/rfc1058.txt>

- [4] Franklin A. Graybill (1976). Theory and Application of the Linear Model Pacific Grove.
- [5] Lancaster, P. and Salkauskas (1986). K. Curve and Surface Fitting: An Introduction. London: Academic Press.
- [6] Nash, J. C (1990). Compact Numerical Methods for Computers: Linear Algebra and Function Minimisation, 2nd ed. Bristol, England: Adam Hilger, pp. 21-24.
- [7] Ronald R. Hocking (1996). Methods and Applications of Linear Models, Wiley Series In.
- [8] William S (2002). High-speed networks and Internets performance and quality of service”, 2nd Prentice Hall. 7-8.

홍 성 혁



Sunghyuck Hong received the Ph.D. degree from Texas Tech University in August, 2007 major in Computer Science. Currently, he works at Division of Information and Communication in Baekseok University as an assistant professor. Before he joined Baekseok, he

worked at International affairs in Texas Tech University as a senior programmer/analyst, and his jobs were development of ASP.NET web applications and maintenance of PC/Server. He is a member of editorial board in the Journal of Korean Society for Internet Information (KSII) Transactions on Internet and Information Systems. His current research interests include Secure Wireless Sensor Networks, Key Management, and Networks Security.

· E-Mail: shong@bu.ac.kr