Routing Protocol based on Floyd-Warshall Algorithm Allowing Maximization of Throughput

Kohei Arai

Graduate School of Science and Engineering
Saga University, Saga City
Japan

Abstract—**Routing protocol based on Floyd-Warshall algorithm which allows maximization of throughput is proposed. The metric function in the proposed routing protocol is throughput including not only send packets but also retransmission packets in order for improving effectiveness and efficiency of the network in concern. Through simulation studies, it is found that the proposed routing protocol is superior to the conventional Open Shortest Path First: OSPF based on Dijkstra algorithm for shortest path determination from the point of view of maximizing throughput. A routing protocol for Virtual Private Network (VPN) in Autonomous System (AS) based on maximizing throughput is proposed. Through a comparison between the proposed protocol and the existing protocols, OSPF (Widely used), it is also found that the required time for transmission of packets from one node to another node of the proposed protocol is 56.54% less than that of the OSPF protocol.**

Keywords—*Network routing protocol; virtual private network; autonomous system; open shortest path first; Floyd-Warshall algorithm; Dijkstra algorithm; throughput*

I. INTRODUCTION

There are two types of adaptive protocols (Interior Gateway Protocol: IGP) [1] used for path control in an autonomous system (Autonomous System: AS)[2]: one based on vector distance and one based on link state [3]. The latter is better. One of the latter is the Routing Information Protocol[4] (RIP)[5] , which is frequently used at present, which stores a table (Routing Table: RT) that indicates the next route to be connected every 30 seconds. This is a method of rewriting in accordance with the number of nodes (hop number) passing before reaching the network node.

As a result, it is possible to cope with a case where a failure occurs in the network or a case where the scale of the network is changed. In recent years, Open Shortest Path First (OSPF)[6] has been put to practical use and used frequently[7]. This is basically one of the adaptive protocols like RIP, and rewrites RT based on the link state (shortest distance). RIP has the danger of rewriting the table infinitely.

For example, when sending a packet from node A to C via B, if the bus to C fails, B's RT cannot be sent to C, so it is rewritten to send to A. In addition, even when the network topology changes, it takes time to transition to a stable state, and there is a risk of falling into an infinite loop. Further, there is a problem that a load is imposed on all nodes at the time of broadcasting. RIPv2, which compensates for this shortcoming, has also been proposed, in which information that avoids the above-mentioned problems is added to the header of the file indicating the link status.

A packet indicating the link state is broadcast to all nodes other than the target node. Based on this, the shortest route is obtained by Dijkstra Algorithm[8]: DA [1], the shortest route to all target nodes is found, and the method of rewriting RT is Link State Protocol: LSP. However, in LSP, the broadcast interval is 30 seconds, so if there is a change during that time, the tracking may be delayed.

If this interval is shortened, the following speeds up, but only the link state packet may occupy the network. Therefore, it has been considered to avoid this problem by adding time information to the link state bucket (Time Stamping[9]). OSPF is a device that adds such a device and supports the netmask (Variable-Length Sub-Net Masking: VLSM[10]. On the other hand, when constructing a virtual private network between earth observation data supply organizations and exchanging attribute information (metadata), etc. of acquired data with each other, a large amount of data is transmitted from a specific organization to a specific organization occurs.

In such a case, if the shortest path from any node to all the target nodes is obtained by the DA, the shortest path may not be between two specific nodes. Therefore, instead of DA, we introduce Floyd-Warshall [2], which guarantees the shortest path between two nodes. OSPF also believes that the shortest path leads to the shortest required time, the shortest delay time, and the lowest packet loss probability. However, here the link communication capacity, network throughput (link packet occupancy) and an optimal path control method considering the processing capability of the computer of the node in the evaluation function [3].

For the proposed route control method, the network topology was assumed, the packets were generated by random numbers, and the time required for the packets to arrive at the

---

[1] https://en.wikipedia.org/wiki/Interior_gateway_protocol
[2] https://en.wikipedia.org/wiki/Autonomous_system_%28Internet%29
[3] http://www.cla.kobe-u.ac.jp/Jouhou/95/Ishizaki/suuri.html
[4] https://ja.wikipedia.org/wiki/Routing_Information_Protocol
[5] http://www.acc.com/internet/whitepapers/iprouting.html
[6] https://ja.wikipedia.org/wiki/Open_Shortest_Path_First
[7] http://www.ecse.rpi.edu/Courses/S98/35696/illrip/index.htm
[8] https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm
[9] https://en.wikipedia.org/wiki/Trusted_timestamping
[10] http://www.vlsm.org/rms46/imho-e-vlsmtjt.html

destination node and the delay time at that time were evaluated. The results are compared with OSPF, and the superiority of the proposed method is confirmed.

The following section describes research background. Then the proposed method is described followed by experiment. After that, conclusion is described together with some discussions.

II. RELATED RESEARCH WORKS

Approach of improved topology development protocol in Ad Hoc network minimizing the number of hops and maintaining connectivity of mobile terminals which move from one to the others is proposed and attempt [4]. Then, routing approach with immediate awareness of adaptive path while minimizing the number of hops and maintaining connectivity of mobile terminals which move from one to the others is proposed and evaluated [5].

On the other hand, agent based approach of routing protocol minimizing the number of hops and maintaining connectivity of mobile terminals which move from one area to the other is attempted [6] together with approach of improved topology development protocol in ad-hoc network minimizing the number of hops and maintaining connectivity of mobile terminals which moves from one to the others [7].

Backup communication routing through Internet Satellite, WINDS, for transmission of disaster relief data is proposed and well reported [8] together with backup communication routing through Internet satellite WINDS for transmission of disaster relief data [9]. Also, back-up communication routing through Internet satellite WINDS for transmitting of disaster relief data is proposed and evaluated [10].

Meanwhile, service robot with communication aid together with routing controlled by human eyes proposed and well report with experiments [11]. More recently, with the spread of mobile terminals equipped with wireless LAN as standard equipment, research on mobile ad-hoc networks (MANET), which builds a network with only wireless terminals, is drawing attention. Many ad hoc routing protocols proposed in MANET do not consider the traffic state on the route when generating the route, so even if the number of relay hops is short, the route with relatively high load is selected. There is a problem of being lost. An ad-hoc routing protocol that is capable of route generation considering traffic on the route by extending OLSR (Optimized Link State Routing) is proposed by Akira Morisaki et al.[12].

SrcRR: A High Throughput Routing Protocol for 802.11 Mesh Network is also proposed by Daniel Aguayo, John Bicket, Robert Morris [13]. SrcRR uses its own transmit bit-rate selection algorithm based on medium-term loss rate measurements, replacing the algorithm built into the radio firmware. Mobile Ad hoc Networks (MANET) are wireless networks consisting of a collection of mobile nodes with no fixed infrastructure. Due to their decentralized, self-configuring and dynamic nature, MANETs offer many advantages and are easy to install. In the recent years, a lot of researches are going on in the area of Mobile Ad hoc Networks (MANETs). This network is an infrastructure -less network where nodes communicate with each other without any aid of centralized administration. In this paper, we are analyzing Throughput of DSR routing protocol [14].

High Throughput Cryptocurrency Routing in Payment Channel Networks is proposed by Vibhaalakshmi Sivarama et al. [15]. They proposed Spider, a routing solution that "packetizes" transactions and uses a multi-path transport protocol to achieve high-throughput routing in PCNs. Packetization allows Spider to complete even large transactions on low-capacity payment channels over time, while the multi-path congestion control protocol ensures balanced utilization of channels and fairness across flows.

III. PROPOSED ROUTING PROTOCOL

Considering the throughput Ti as "Ni", the number of packets arriving at an arbitrary node i in a unit time, the time t_1 from the arrival of an arbitrary packet at an arbitrary node i to another arbitrary node j is determined by nodes i, j If the communication capacity of the link between is α, it can be expressed by equation (1).

$$t_l = Ci / Ni = Ci / Ti \qquad (1)$$

The processing power of the node's computer is directly related to the time required for a packet to pass through the node. Considering the processing capacity Ai as the number MZ of packets passing through an arbitrary node i in a unit time, and considering the delay time Di at the node as the length of the packet queue [4],

$$t_n = (Ti + Di) / Az \qquad (2)$$

The shortest path that minimizes the time ti required to pass through these nodes and links is determined.

$$ti = ti + tn = \{CiAi + Ti^2 + TiDi\} / TiAi \rightarrow \min \qquad (3)$$

Here, the load Li is defined as follows.

$$Li = Ni / Ci = 1/t_1 = Ti / Ci \qquad (4)$$

The route control method proposed here seeks an optimal route between two nodes that optimizes the weighted average m of the load and the distance.

That is, it is a value obtained by normalizing the throughput by the communication capacity. The route control method proposed here seeks an optimal route between two nodes that optimizes the weighted average m of the load and the distance.

$$m = \omega_L Li + \omega_t ti \rightarrow \min. \qquad (5)$$

Here, $\omega_L + \omega_t = 1$. DA is a kind of polynomial time algorithm in dynamic programming [4], which guarantees the shortest path for all destination nodes. Here, an optimization method that guarantees the shortest path between any two nodes is necessary, and the algorithm of Floyd-Warshall: WF (see Appendix) is adopted as an algorithm that satisfies this.

In other words, using the link state information of the target node written in the RT, find the shortest path to the target node based on the processing capacity of the node and the communication capacity of the link known in advance, rewrite RT, and This is a method in which information is broadcast as link state packets every 30 seconds as in OSPF.

The differences from OSPF are summarized below.

- In OSPF, the shortest path from any node to all nodes is obtained by DA, but in the proposed method, the shortest path between two nodes from any node to the target node is obtained by WF.

- Find the optimal route based on the combination of link communication capacity and node processing capacity not considered in OSPF.

- The concept of throughput is newly introduced, and the optimized route is obtained by optimizing the weighted average of the load normalized by the communication capacity and the distance considered by OSPF.

IV. SIMULATION EXPERIMENTS

A. Method for Simulation

Since a universal simulation seems to be impossible, a special example (a network topology with 8 nodes) was considered here and the simulation was performed in the following order.

1) Randomly determine the network topology (8 and 16 nodes), source and destination nodes. At that time, the communication capacity of the link and the processing capacity of the node are determined at random.

2) A packet is generated from each node using a uniform random number during 2000 time slots (unit time) that does not hinder the calculation of throughput and communication distance, and the distance between all nodes and the load are calculated in advance.

3) At this time, the unit time is the length of the packet. This is equivalent to generating a fixed-length packet in a slotted time, such as an ATM (Asynchronous Transfer Mode[11]) cell. The occurrence probability is the same for each node and can be set arbitrarily. Furthermore, when the queue of each node exceeds 20, it is assumed that the node is disconnected until the next unit time.

4) After that, the weight of the distance and the load is optimized, and the optimal route between any two nodes is obtained by the WF algorithm.

5) Compare the result and the shortest path between all nodes determined by the DA algorithm based on only the distance with the path between the two nodes (corresponding to the shortest path obtained by OSPF).

B. Simulation Results

Fig. 1 shows the network configuration of the eight nodes used in this example.

In the figure, the processing capacity of the node and the communication capacity of the link are shown. Fig. 2 and 3 show the simulation results of the distance and load for the configuration of 8 nodes and 16 nodes. The load in the figure is the total load of 2000 time slots. In the case of 16 nodes, a new 8 node is added to the existing 8 nodes. At this time, the packet occurrence probability was set to 0.1. Here, the average

number of packets generated by each node in 2000 time slots is 200. Table I shows the case of the shortest path control of OSPF in the case of 8 nodes, and the difference of the path by the minimum load path control proposed here and the optimal path control with the shortest distance and the minimum load weighted.

In the table, the notations of 50:50, 90:10, and 99:1 correspond to load and distance weights of 0.5: 0.5, 0.9: 0.1, and 0.99: 0.01, respectively. In the case of the shortest route, the DA algorithm, the minimum load, and the weighted optimal route are obtained by the WF algorithm. When the weight is 0.5: 0.5, almost the same route as OSPF is selected. Although only 4-8 is different, in this case, the distance is the same for 4-1-2-6-8 and 4-7-8, but the load is the natural consequence of the latter being smaller.

When the weight is set to 0.9: 0.1, the route is different from OSPF. This situation is not so different from the case where the weight is set to 0.99: 0.01, so 0.9: 0.1 is adopted here as the weight. It was confirmed that this result was also applicable to the case of 16 nodes.
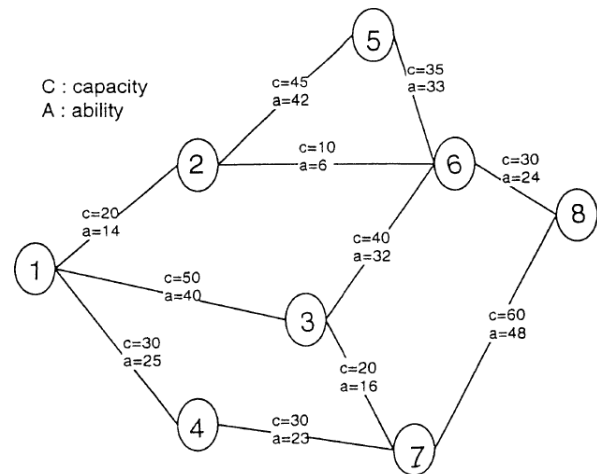

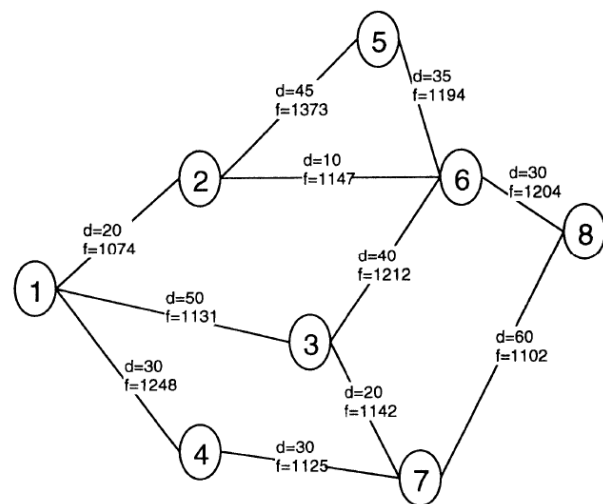Fig. 1. Network Configuration of Interest (8 Nodes).

Fig. 2. The Distance (d) and Accumulated Load (f) Between Nodes Over 2000 Time Slots for 8 Nodes of Network Configuration.

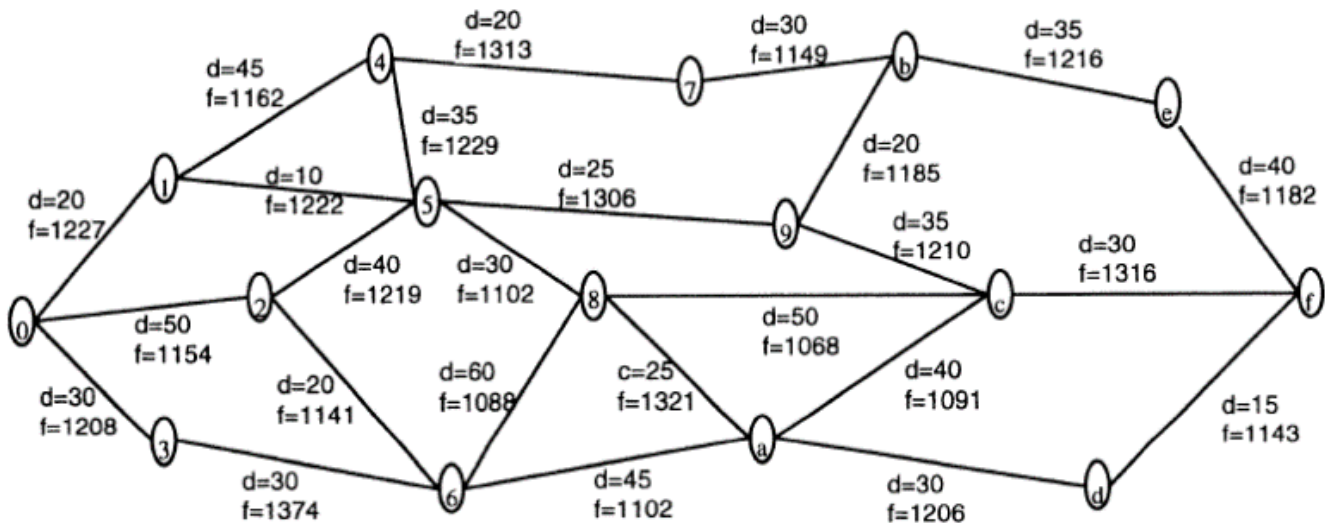[11] https://ja.wikipedia.org/wiki/Asynchronous_Transfer_Mode

Fig. 3.   The Distance (d) and Accumulated Load (f) Between Nodes Over 2000 Time Slots for 16 Nodes of Network Configuration.

TABLE I.   A Comparison of Optimum Route Among OSPF, Minimizing Load and Minimizing Weighted Average Between Distance and Load
for 8 nodes network configuration

start-goal	OSPF	Min. Load	Weight(0.5,0.5)	0.9:0.1	0.99:0.01
1-2	1-2	1-2	1-2	1-2	1-2
1-3	1-3	1-3	1-3	1-3	1-3
1-4	1-4	1-4	1-4	1-4	1-4
1-5	1-2-5	1-2-5	1-2-5	1-2-5	1-2-5
1-6	1-2-6	1-2-6	1-2-6	1-2-6	1-2-6
1-7	1-4-7	1-3-7	1-4-7	1-3-7	1-3-7
1-8	1-2-6-8	1-3-7-8	1-2-6-8	1-2-6-8	1-3-7-8
2-3	2-6-3	2-1-3	2-6-3	2-1-3	2-1-3
2-4	2-1-4	2-1-4	2-1-4	2-1-4	2-1-4
2-5	2-5	2-5	2-5	2-5	2-5
2-6	2-6	2-6	2-6	2-6	2-6
2-7	2-1-3-7	2-6-8-7	2-1-3-7	2-1-3-7	2-6-8-7
2-8	2-6-8	2-6-8	2-6-8	2-6-8	2-6-8
3-4	3-7-4	3-7-4	3-7-4	3-1-4	3-7-4
3-5	3-6-5	3-6-5	3-6-5	3-6-5	3-6-5
3-6	3-6	3-6	3-6	3-6	3-6
3-7	3-7	3-7	3-7	3-7	3-7
3-8	3-6-8	3-7-8	3-6-8	3-7-8	3-7-8
4-5	4-1-2-5	4-1-2-5	4-1-2-5	4-1-2-5	4-1-2-5
4-6	4-1-2-6	4-7-8-6	4-1-2-6	4-1-2-6	4-7-8-6
4-7	4-7	4-7	4-7	4-7	4-7
4-8	4-1-2-6-8	4-7-8	4-7-8	4-7-8	4-7-8
5-6	5-6	5-6	5-6	5-6	5-6
5-7	5-6-3-7	5-6-8-7	5-6-3-7	5-6-3-7	5-6-8-7
5-8	5-6-8	5-6-8	5-6-8	5-6-8	5-6-8
6-7	6-3-7	6-8-7	6-3-7	6-3-7	6-8-7
6-8	6-8	6-8	6-8	6-8	6-8
7-8	7-8	7-8	7-8	7-8	7-8

TABLE II.   MEAN ARRIVAL TIME FOR EACH 100 OF TIME-SLOT (APPROXIMATELY 56.54% OF IMPROVEMENT OF THE TIME REQUIRED FOR TRANSMISSION OF PACKETS IS EXPECTED FOR THE PROPOSED PROTOCOL COMPARED TO THE OSPF IN CASE OF 8 NODES

Time set	OSPF(min. distance)	Min.Load	Optimum weight for distance and load
100	9.78	9.78	7.96
200	7.87	Fail	7.87
300	16.23	16.23	16.23
400	Fail	Fail	9.94
500	16.57	16.57	7.34
600	10	35.98	7.84
700	9.2	9.2	9.19
800	Fail	Fail	7.45
900	8.42	26.73	8.42
1000	11.27	11.27	6.75
1100	10.32	10.32	7.08
1200	10.06	10.06	6.63
1300	41.03	41.03	10.85
1400	Fail	40.4	9.44
1500	7.77	7.77	7.77
1600	Fail	66.84	8.87
1700	16.08	16.08	7.12
1800	7.98	26.81	7.98
1900	24.9	24.9	17.88
2000	8.39	8.39	8.12
average	20.79	26.42	9.14

Furthermore, when the average time required for every 100 time slots from the generation of a bucket to the arrival of the packet when the packet generation node (Start) is 1 and the destination node (Goal) is 8 is as shown in Table II.

Therefore, comparing only the shortest path and the minimum load of OSPF, OSPF is superior except in the case of time slots 1400 and 1600. OSPF is inferior when the queue is broken and exceeds 20, and is avoided by minimum load control. The proposed route control with weight 0.9: 0.1 is not inferior to OSPF in all cases. It was confirmed that this conclusion was the same for the case of 16 nodes.

The biggest difference between OSPF and the proposed route control is that OSPF uses DA to minimize the distance from any node to all nodes, whereas the proposed method uses WF to determine the distance between any nodes. The point is to minimize it. To confirm the effect, we examined the time required to transmit a packet from node 1 to 8 for 8 nodes and from node 0 to f for 16 nodes. As a result, as shown in Table III, it was confirmed that the average arrival time over 2000 time slots can be reduced by 36.58% and 13.69% for 8 nodes and 16 nodes, respectively, by the proposed method.

TABLE III.   AVERAGED PACKET ARRIVAL TIME FOR THE NETWORK WITH 8 NODES (FROM NODE 1 TO 8) AND FOR THE NETWORK WITH 16 NODES (FROM NODE 0 TO f) OVER 2000 TIME SLOTS

Network Node	OSPF	Proposed	Percent Improvement
8 Nodes from Node 1 to 8	5.56	3.52	36.38
16 Nodes from 1 to f	7.23	6.24	13.69

V.   CONCLUSION

Routing protocol based on Floyd-Warshall algorithm allowing maximization of throughput is proposed. Through simulation studies, it is found that the proposed routing protocol is superior to the conventional Open Shortest Path First: OSPF based on Dijkstra algorithm for shortest path determination from the point of view of maximizing throughput.

A routing protocol for Virtual Private Network (VPN) in Autonomous System (AS) based on maximizing throughput is proposed. Through a comparison between the proposed protocol and the existing protocols, OSPF (Widely used), it was found that the required time for transmission of packets from one node to another node of the proposed protocol is 56.54% less than that of the OSPF protocol.

The proposed route control method in an autonomous system is based on a similar method (OSPF) that is already widely used, and introduces a new concept of throughput in the shortest distance that OSPF is based on, and calculates the weighted average of them. Is the new norm. By this, it was confirmed that it was possible to search for an optimal route with a light load, although not the shortest distance, but the distance was rather short.

In this paper, it is confirmed that 36.58 and 13.69% reduction in arrival time can be achieved with the network configurations of 8 and 16 nodes, respectively.

VI.   FURTHER RESEARCH WORKS

This time, the author has only given an example of the packet occurrence probability, but we plan to examine more cases in the future. In addition, automatic estimation of the optimal weight of distance and load is also a subject for the future.

VII. APPENDIX (FLOYD-WARSHALL'S ALGORITHM)

The algorithm generates a series of matrices by successively including new nodes that give the shortest path to each node pair. In the matrix Dk, the path between any two nodes uses only the nodes 1 to k. Therefore, if there are N nodes in total, a true shortest path is obtained when $k = N$. The algorithm is as follows:

1. Assuming that the total number of nodes is *N*, assign a number from 1 to *N* to each node in the network.
2. Define an $N \times N$ matrix M so that the element a_{ij} indicates the length of the path connecting the node *i* to the node *j*.
3. If there is no path connecting *i* and *j*, a_{ij} is set to infinity (∞) or set to a sufficiently large value (Max) for the purpose of calculation.
4. An $N \times N$ matrix Dk ($k = 0,..., N$) is sequentially generated as follows.
5. The matrix D^k is for all node pairs i, *j* such that the path between node *i* and node j number may include some node from the set (1,2, ..., *k*) as well as node *i, j* This can be explained as giving the shortest path length. (Therefore, matrix *D*- consists only of the nodes at the end of each path, which is similar to matrix *M*.) Matrix $D^{k + 1}$ determines whether or not to include node $k + 1$ in the already existing shortest path. It is obtained from the matrix D^k.

$$d_{i,j}^{k+1} = \min(d_{i,j}^{k1}, d_{i,k+1}^{k1} + d_{k+1,j}^{k1}) \qquad (6)$$

Here, the function min takes the minimum value among its arguments. That is, in the shortest path for each node, if the path including the node of $k + 1$ is shorter than the path of the matrix D^K, the node of $k + 1$ is included.

6. The matrix D^N gives the shortest path.

Acknowledgment

References

[1] Masao Fukushima, Introduction to Mathematical Programming (in Japanese), Asakura Shoten, Tokyo, 1996

[2] Toshihide Ibaraki, Masao Fukushima, Methods of Optimization (in Japanese), Kyoritu Shuppan, Tokyo, 1993. [Chinese edition: Translated by D.Z. Zeng, published by World Publishing Corporation, Beijing, 1997

[3] D. R. Cox and Walter L. Smith. Queues, Methuen, London; Wiley, New York, 1961.

[4] Kohei Arai, Lipur Sugiyanta, Approach of improved topology development protocol in Ad Hoc network minimizing the number of hops and maintaining connectivity of mobile terminals which move from one to the others, Anthony V. Stavros Edt., Advances in Communication and Media Research, Vol.8, ISBN 978-1-61324-794-5, 2011

[5] Kohei Arai, Lipur Sugiyanta, Routing Approach with Immediate Awareness of Adaptive Path While Minimizing the Number of Hops and Maintaining Connectivity of Mobile Terminals Which Move from One to the Others, International Journal of Computer Science and Information security, 9, 2, 94-101, 2011.

[6] Kohei Arai and Lipur Sugiyanta, Agent based approach of routing protocol minimizing the number of hops and maintaining connectivity of mobile terminals which move from one area to the other, Proceedings of the International Conference on Computational Science and Its Applications (ICCSA2010), LNCS part-III, 305-320,2010.

[7] Kohei Arai, Lipur Sugiyanta, Approach of improved topology development protocol in ad-hoc network minimizing the number of hops and maintaining connectivity of mobile terminals which moves from one to the others, Journal of Communication and Networks, 2, 6, 190-204, 2011.

[8] Kohei Arai, Kiyotaka Fujisaki, Hiroaki Ikemi, Masato Masuya, Terumasa Miyahara, Backup communication routing through Internet Satellite, WINDS, for transmission of disaster relief data, Proceedings of the International Symposium on WINDS Application Experiments, 2010.

[9] Kohei Arai, Backup communication routing through Internet satellite WINDS for transmission of disaster relief data, Proceedings of the International Symposium on WINDS, 2010

[10] Kohei Arai, Back-up communication routing through Internet satellite WINDS for transmitting of disaster relief data, International Journal of Advanced Computer Science and Applications, 2, 9, 21-26, 2011.

[11] Kohei Arai, Service robot with comminunication aid together with routing controlled by human eyes, Journal of Image Laboratory, 25, 6, 24-29, 2014

[12] Akira Morisaki et al., http://www.wata-lab.meijo-u.ac.jp/file/convention/2009/200907-DICOMO-Akira_Morisaki.pdf

[13] Daniel Aguayo, John Bicket, Robert Morris, http://pdos.csail.mit.edu/~rtm/srcrr-draft.pdf

[14] GirishTiwari1,*Ram Shiromani Gupta, Throughput Based Analysis of DSR Routing Protocol in MANET, IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 12, Issue 6, Ver. II P 56-62, 2017.

[15] Vibhaalakshmi Sivarama et al., High Throughput Cryptocurrency Routing in Payment Channel Networks, https://arxiv.org/pdf/1809.05088.pdf

Author's Profile

Kohei Arai, He received BS, MS and PhD degrees in 1972, 1974 and 1982, respectively. He was with The Institute for Industrial Science and Technology of the University of Tokyo from April 1974 to December 1978 also was with National Space Development Agency of Japan from January, 1979 to March, 1990. During from 1985 to 1987, he was with Canada Centre for Remote Sensing as a Post Doctoral Fellow of National Science and Engineering Research Council of Canada. He moved to Saga University as a Professor in Department of Information Science on April 1990. He was a councilor for the Aeronautics and Space related to the Technology Committee of the Ministry of Science and Technology during from 1998 to 2000. He was a councilor of Saga University for 2002 and 2003. He also was an executive councilor for the Remote Sensing Society of Japan for 2003 to 2005. He is a Science Council of Japan Special Member since 2012. He is an Adjunct Professor of University of Arizona, USA since 1998. He also is Vice Chairman of the Science Commission "A" of ICSU/COSPAR since 2008 then he is now award committee member of ICSU/COSPAR. He wrote 55 books and published 620 journal papers as well as 450 conference papers. He received 66 of awards including ICSU/COSPAR Vikram Sarabhai Medal in 2016, and Science award of Ministry of Mister of Education of Japan in 2015. He is now Editor-in-Chief of IJACSA and IJISA. http://teagis.ip.is.saga-u.ac.jp/index.html