International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.8, Issue.9, September 2020 E-ISSN: 2347-2693

Distributed Path Computation with Intermediate Variables (DPCIV) for Distributed Routing Algorithms to Guarantee Routing Decisions

K.V. Nishad[1], R. Vadivel[2*]

[1]Department of Information Technology, Bharathiar University, Tamil Nadu India
[2]Department of Information Technology, Bharathiar University, India

Corresponding Author: vlr_vadivel@yahoo.co.in

DOI: https://doi.org/10.26438/ijcse/v8i9.5963 | Available online at: www.ijcseonline.org

Received: 05/Sept/2020, Accepted: 12/Sept/2020, Published: 30/Sept/2020

***Abstract*—** Distributed routing algorithms can lead to a temporary elevation between path regeneration, which can cause major stability problems in high-speed networks. This paper introduces a new algorithm, Distributed Path Computing with Intermediate Variables (DPCIV), which can be integrated with any distributed algorithm to ensure that the directed graph caused by route decisions is always acyclic. An important contribution of DPCIV, in addition to its ability to work with any routing algorithm, is the update method using simple message exchanges between neighboring locations that ensure maximum ease at all times. DPCIV apparently outperforms existing loop blocking algorithms in key metrics such as the frequency of synchronized refresh and the ability to save paths during the transition. The simulation results that block these advantages in the context of a very short path are presented. In addition, the universal performance of DPCIV is demonstrated by studying its use of a functionally-oriented non-shortcut. In particular, the route seeks to counteract the power of failure by increasing the number of subsequent hops available at each destination.

***Keywords*—** Distance vector, shortest path, link state, routing algorithm.

I. INTRODUCTION

The main motive of this paper is to establish a distributed path computation with intermediate variables (DPCIV) for guarantee routing. Distributed transmission is the basic function of modern communication networks and is expected to remain the case, though some recent suggestions consider the use of very centralized solutions [1]. Depending on the information distribution mode and the following computer using distributed information, there are two broad categories of algorithms: (i) link-state algorithms (also known as topology scattering) and (ii) remote-vector algorithms [2]. In all three modes, nodes select successive (next-hop) destinations for each destination based solely on location information, with the intention that selected destination routes are more efficient in the sense - e.g., with lower costs. Because end-to-end approaches are constructed by combining outcome measures in specific areas, achieving a global goal means consistency in all areas of equilibrium in the complexity and knowledge on which those calculations are based. Thus, recognizing the benefits of remote sensing solutions, even in environments where they can be culturally similar, requires developing ways to overcome these problems. Such awareness is not new. Since the 1970s, a number of works have been performed referring to this purpose in the form of short computations of the method [3].

II. RELATED WORK

Link-state algorithms, which is a well-known OSPF protocol [4], which distributes the state of each geographical network (its state and its connection points) to all other network locations through reliable flooding. After receiving network status updates from all locations, each location independently calculates the route to all locations. The timing of information inconsistencies across locations is relatively small (a few tens of miles per unit in modern day networks [5]), so that the loops, if any, remain for a while. On the flip side, link-state algorithms can be more up-to-date in terms of communication (stream updates), storage (full network mapping), and complication (changes wherever the network triggers global views). These are just some of the reasons to investigate alternatives included in distance algorithms, which are the focus of this paper [6]. Distance-vector Expression Distribution Distributed data now contains the results of a specific design (e.g. its current cost per area) that distribute to its neighbors, who also work their own calculations before continuing to distribute any updated results to their neighbors [7]. The Bellman-Ford (DBF) distributed algorithm is a well-known example of the use of a remote-vector algorithm (cf. RIP [8], EIGRP [9]) that combines a tree of shortest path from a given point to a whole. Moving the distribution of information and computing can reduce storage requirements (only route information is stored), over connections (no flood packet transfers), and accounting (location change does not require distribution beyond the affected area). Thus, remote-vector algorithms avoid a number of the disadvantages of link-state algorithms, which may be desirable, especially in cases of switching to local topology and / or where high-level control is not required [8].

The cause of data transmission and data transmission is that the distribution of information is taken at computer speed because the node cannot send updates before completing its current calculation. This may extend to times when areas have incompatible data, which, as discussed earlier and shown in Section V, can lead to narrower and longer route townships. In addition, moving data to distribution and calculation can also result in slower connections. This is because each node relies on the (partial) integration effects of its neighbors, which can deliver cyclic dependencies that increase the number of steps needed to achieve a final, accurate result [9]. Indeed, when locations are inaccessible, the distance-based algorithm may also not come up with a specific number of steps. This is known as a count-to-infinity problem, which is absent in link-state algorithms where nodes compute paths independently. (In fact, when the cost-to-reach is high, the specified location is inaccessible and the calculation terminated). The research began with a renewed interest in finding solutions for large Ethernet computers. In particular, considering extending the vulnerability of Ethernet networks by introducing faster traffic algorithms into the existing distribution algorithm [10]. It is proposed to run state-of the-art solutions for developing Ethernet networks, or not for the sake of the population, and a registry solution looks like an attractive alternative.

a. Existing work

Incompatible information in different locations may have serious consequences beyond achieving the desired success. Of particular importance is the possible formation of temporary route loops, which can have a detrimental effect on network performance especially on wireless networks or limited access routes, eg, no Time-toLive (TTL) field in packet heads or TTL is set to a large value. In the presence of a routing loop, the packet hosted in the loop returns to the same locations many times, thus increasing the load of the bulk traffic in the affected communication channels [11].

The problem, the biggest problem with even unicast packets, is increasing more and more with broadcast packets, which not only get trapped in every layer on the network, but also generate duplicate packets across all network interfaces. The occurrence of a routing loop then often causes a wide network delay, which can result in the delay or delay of the same control (update) packet required to terminate the loop; thus creating a situation where a temporary problem has a lasting effect [12]. Avoiding the movement of temporary loops remains an important requirement for integrating traffic into existing and emerging network technologies, in recent discussions.

III. METHODOLOGY - DISTRIBUTED PATH COMPUTATION WITH INTERMEDIATE VARIABLES (DPCIV)

Distributed algorithms can lead to temporary loops during trace recovery, which can create major stability problems in high-speed networks. By introducing a new algorithm called Distributed Path Computing with Intermediate Variables (DPCIV), which can be integrated with any distributed algorithm to ensure that the directed graph caused by route decisions is always acyclic [13]. An important contribution of DPCIV, in addition to its ability to work with any routing algorithm, is the update method using simple message exchanges between neighbouring locations that ensure maximum ease at all times. DPCIV apparently outperforms existing loop-blocking algorithms in key metrics such as the frequency of synchronized refresh and the ability to save paths during the transition. The simulation results that block these advantages in the context of a very short path are presented. In addition, the universal performance of DPCIV is demonstrated by studying its use of a functionally-oriented non-shortcut. In particular, the route seeks to counteract the power of failure by increasing the number of subsequent hops available at each destination. Link-state algorithms, which is a well-known OSPF protocol; Then calculated the state of each connection in each of the other locations in the network using reliable flooding [14]. After receiving network status updates from all locations, each location independently calculates the route to all locations. The timing of information inconsistencies in all areas is so short that loops, if any, are short-lived. On the flip side, link-state algorithms can be more advanced depending on network storage, and computation. These are just some of the reasons to investigate alternatives included in distance algorithms, which are the focus of this paper.

A. Module Definition

The following modules are available for this research project.

- **Distributed Time-to-Live Module**
- Loop Free Routing Module
    - **Robust Routing Module**
    - **Shortest-path computation Module (or) shortest-path Simulation Module**

a. Distributed Time-to-Live Module

Time-to-Live (TTL) field in packet headers or TTL is set to a maximum value. When there is a loop for a route, the packet caught in the loop returns to the same locations many times, thus increasing the load of traffic more on the themes and objects involved. The problem, the biggest problem with even unicast packets, is increasing more and more with broadcast packets, which not only get trapped in every layer on the network, but also generate duplicate packets across all network interfaces.

b. Loop Free Routing Module

Free Loop information distribution and distribution can also lead to slow connections. This is because each node relies on the integration effects of its neighbors, which can introduce dependence on the cycle that increases the number of steps needed to achieve the final, relevant result. Indeed, when locations are inaccessible, the distance-based algorithm may also not come up with a specific number of steps.

c. Robust Routing Module

It shows the benefits of this reunion using the cost function that instead of a very long distance work, you want to maximize the number of subsequent hops available at all destinations. The availability of multiple hops ensures that the failure of any one link or neighbour does not hinder the node's ability to continue moving traffic to its destination. Failure results in the loss of at least one hop to the destination, so that the node can continue to send packets to the remainder without waiting for new calculation methods.

d. Shortest-path computation Module

The shortest path models are performed on a random graph with a 5-point scale, but in order to produce a wide range of configurations, a number of different parameters are used. Figure 1 shows the entire flow of DPCIV.

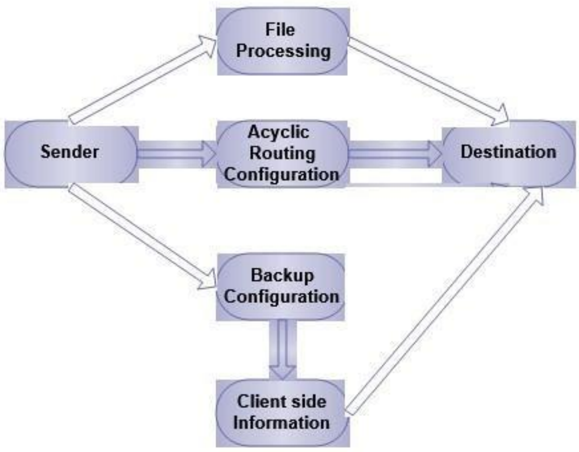

Figure 1. Routing is distributed via Intermediate variables (DPCIV)

IV. PERFORMANCE EVALUATION

This section presents simulation results by comparing the performance of DPCIV (in the standard mode used with the DBF to calculate the shortest paths) depending on the movement of loops, assembly times and the frequency of synchronous regeneration against DUAL. DBF operations other than DPCIV have also been presented as a reference. Use is done on random graphs with an average degree of 5. The number of nodes varies from 10 to 90 in 10 increments. For each graph size, 100 random graphs are generated. The cost of the link is derived from the bi-modal distribution: at 0.5 the probability of the cost of the connection is evenly distributed at [0,1]; and probability 0.5 is evenly distributed at [0,100] [12]. For each graph, an inexpensive 100-coordinate cost change is introduced, and then released from the same bi-modal stream. All three algorithms work on the same graph and sequence of variables [13]. Processing time for each message is fixed: it 2 s with probability 0.0001, 200 ms with probability 0.05, and 10ms with one. Algorithm Difficulty Update (Two): DUAL, which is part of the widely used EISRP for CISCO, is probably the most popular algorithm. In DUAL, each method continues, in each area, a set of neighbors called a success tracking set. A probability set of probabilities is calculated using a valid probability of incorporating distances that occur in a given area. There are a number of conditions for a few possible adverse events [14] that are tightly integrated into the short-circuit combination. Table1 shows the performance requirements.

Table 1. Performance Requirements

Operating System	Windows 8
Language	Java
JVM	J2SE 1.4
JDBC Vendor	Oracle
RAM Memory	2 GB RAM
Disk Space	1 GB

A. Results and Discussion

a. Processing Time

Processing time is the total time taken to complete the process of sending a packet from source to destination and it is otherwise called as execution time. Table 2 describes the processing time evaluation between DPCIV, DUAL and DBF. Figure 2 shows the graph for processing time. On seeing the table and graph it is clear that processing time is reduced by 91.34% in the proposed method.

Table 2. Processing Time Evaluation

Number of nodes	Processing Time (sec)		
	DPCIV	*DUAL*	*DBF*
20	0.873	1.357	1.879
40	0.923	1.964	2.638
60	1.456	2.032	2.845
80	1.957	2.681	3.064
100	2.539	3.572	3.168

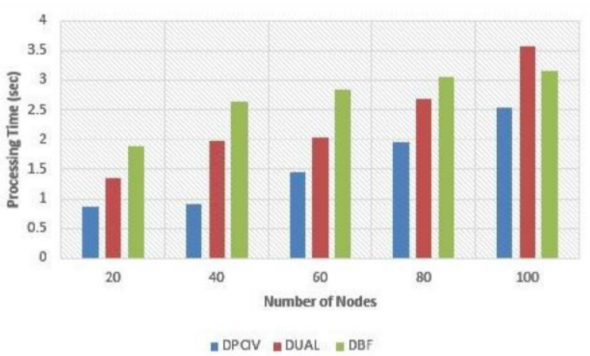

Figure 2. Graph for processing time

b. Packet Delivery Ratio

Packet Delivery Ratio (PDR) is the ration of packets received successfully to the destination. PDR depends upon the number packets send from source. Table 3 describes the PDR values obtained by proposed method and other existing methods. Figure 3 shows the graph view of PDR values. By analysing the graph the packet delivery ratio is reduced in the proposed method by 90.75%.

Table 3. Packet Delivery Ratio

Number of nodes	PDR (%)		
	DPCIV	DUAL	DBF
20	98.35	89.68	84.54
40	96.24	87.47	83.9
60	93.14	83.03	88.81
80	92.41	82.61	76.24
100	90.75	80.21	72.12

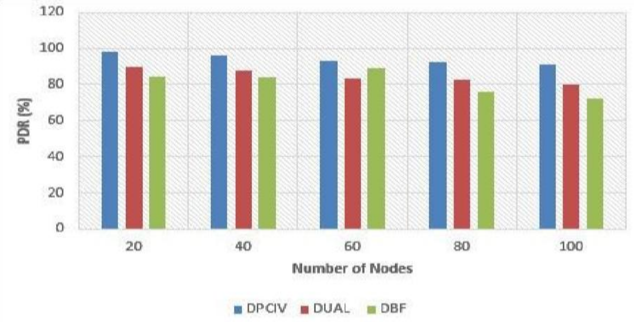

Figure 3. Graph for Packet Delivery Ratio (%)

c. Energy Consumption

Energy consumption is the battery power consumed for sending and received data packets. It is measured in Joule. Table 4 explains the energy consumed by the proposed DPCIV and DUAL and DBF. Figure 4 shows the energy consumption between proposed and existing methods. In the proposed method the energy consumed by the proposed method is reduced as 59.35 Joule.

Table 4. Energy Consumption

Number of nodes	Energy Consumption (Joule)		
	DPCIV	*DUAL*	*DBF*
20	73.287	89.465	92.073
40	69.365	83.658	95.389
60	62.699	90.426	96.83
80	60.221	93.57	97.273
100	59.35	100.455	103.465

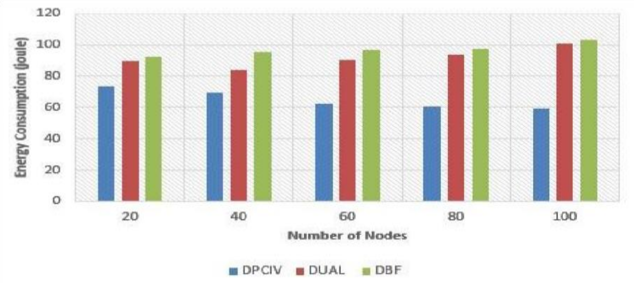

Figure 4. Graph for Energy Consumption

d. Delay

Delay is defined as the time taken for a data packet to reach the destination node. It is measured in milliseconds. In table 5 the time delay made by the proposed DPCIV, DUAL and DBF methods. Figure 5 shows the graph for delay happened during packet sending and receiving. The delay is reduced in the proposed method of about 68 milliseconds.

Table 5. Delay Evaluation

Number of nodes	Delay (milliseconds)		
	DPCIV	*DUAL*	*DBF*
20	96	132	148
40	87	120	128
60	80	111	119
80	72	96	103
100	68	93	95

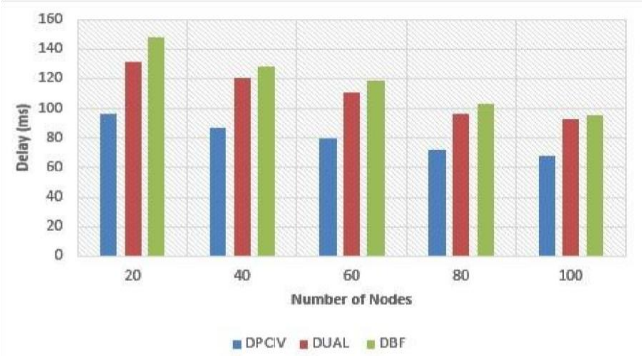

Figure 5. Graph for Delay

V. CONCLUSION AND FUTURE SCOPE

Distance Vector Algorithms (DVA) has advantages over state-of-the-art algorithms, e.g., lower resource requirements and more robustness by maintaining the impact of local change. However, relying on all the areas they make can increase the impact and duration of decisions that are inconsistent across locations. This manifests itself through the temporary hooks and calculation-infinity problem described earlier. Practical ways to overcome these limitations without affecting the intrinsic benefits of vector distance function are important. In this paper, introduce a novel algorithm, Distributed Path Computing with Intermediate Variables (DPCIV), which achieves this by setting rule-overriding existing algorithms and defining an appropriate way to update those rules; both are easy to use. In addition, because DPCIV is not integrated with short traffic integration, it can be used with any routing algorithm. When used with short algorithms, the method has been shown to work better than other current methods, such as the DUAL regression algorithm (and, consequently, DUAL-based principles), both analytically and simulated in various metrics. Another important benefit of DPCIV is that it handles message loss and out-of-sequence delivery, and allows nodes to accept policies that are opposed to the handling of multiple revisions, opening up the possibility of multiple accesses. In the end, the rule set and the evidence for the accuracy of the DPCIV are accurate, which should serve the proper (and correct) implementation. On verifying the results obtained by the proposed DPCIV method, it is very much clear that it outperforms well over the other existing methods. The improvement is of 99.12 %.

REFERENCES

[1] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and K. van der Merwe, "The case for separating routing from routers", in Proceedings ACM SIGCOMM Workshop FDNA, Portland, OR, **pp. 5–12**.Sep. **2004**.

[2] A. Shankar, C. Alaettinoglu, K. Dussa-Zieger, and I. Matta, "Transient and steady-state performance of routing protocols: Distance-vector versus link-state," Internetw.: Res. Exper., vol. **6**, no. 2, **pp. 59–87**, Jun. 1**9**95.

[3] A. Myers, E. Ng, and H. Zhang, "Rethinking the service model: Scaling Ethernet to a million nodes", presented at the ACM SIGCOMM Hot- Nets, San Diego, CA, **2004**.

[4] Y. Ohba, "Issues on loop prevention in MPLS networks", IEEE Communication Magazine, vol. **37**, no. **12**, **pp. 64–68**, Dec. **1999**.

[5] T. Ramesh and S.Sumithra, "A Review on Different Approaches of Mutation Cost Reduction Techniques", International Journal of Innovative Research in Computer and Communication Engineering, Vol.**5**, Issue.**1**, pp.**771-775**, January **2017**.

[6] P. Francois and O. Bonaventure, "Avoiding transient loops during IGP convergence in IP networks", in Proceedings IEEE INFOCOM, Miami, FL, Mar. vol. **1**, pp. **237**–**247**.**2005**.

[7] Gafni and D. Bertsekas, "Distributed algorithms for generating loop-free routes in networks with frequently changing topology", IEEE Transaction Communications, vol. COM-29, no. 1, pp. **11**–**18**, Jan. **1981**.

[8] T. Ramesh and S.Sumithra, OPKNOT – Schemata Method for Mutation Testing Experimented on Event Driven Web Applications", International Journal of Innovative Research in Computer and Communication Engineering, ISSN 2320-9798,Vol **5**, Issue **1**, , **PP 761-770**, January **2017**.

[9] J. M. Jaffe and F. M. Moss, "A responsive routing algorithm for computer networks", IEEE Transaction Communication, vol. **30**, pp. **1768**–**1762**, Jul.**1982**.

[10] J. J. Garcia-Lunes-Aceves, "Loop-free routing using diffusing computations", IEEE/ACMTransaction. Networks, vol. **1**, no. **1**, pp. **130**–**141**, Feb. **1993**.

[11] S. Vutukury and J. J. Garcia-Luna-Aceves, "A simple approximation to minimum-delay routing," in Proceedings ACM SIGCOMM, pp. **227**–**238**.**1999**,

[12] S. Vutukury and J. J. Garcia-Luna-Aceves, "MDVA: A distance-vector multipath routing protocol," in Proceedings IEEE INFOCOM, Anchorage, AK, vol. **1**, pp. **557**–**564**.Apr. **2001**.

[13] K. Elmeleegy, A. L. Cox, and T. S. E. Ng, "On countto-infinity induced forwarding loops in Ethernet networks," in Proceedings IEEE INFOCOM, , pp. **1**– **13**.2**0**06.

[14] S. Dubey1 , R. Jhaggar, R. Verm , D. Gaur, "Encryption and Decryption of Data by Genetic Algorithm", International Journal of Scientific Research in Computer Science and Engineering, Vol.**5**, Issue**.3**, pp.**42-46**, June **2017**.

[15] Amr M. Kishk1, Nagy W. Messiha, Nawal A. El-Fishawy, Abdelrahman A. Alkafs and Ahmed H. Madian, "Proposed Jamming Removal Technique for Wireless Sensor Network", International Journal of Scientific Research in Network Security and Communication, Vol.**3**, Issue. **2**, April **2015**.

AUTHORS PROFILE

K.V. Nishad received Bachelors Degree in Information technology in the year 2018 from Nilgiri College of Arts and Science, Thaloor, The Nilgiri, Tamil Nadu, affiliated to Bharathiar University. He is currently pursuing a Masters Degree in Information Technology from 2018 to 2020, at Bharathiar University, Coimbatore, Tamil Nadu. His area of interest is Web Designing and Cloud computing

R. Vadivel is an Assistant Professor in the Department of Information Technology, Bharathiar University, Tamil Nadu, India. He received his Ph.D degree in Computer Science from Manonmaniam Sundaranar University in the year 2013. He obtained his Diploma in Electronics and Communication Engineering from State Board of Technical Education in the year 1999, B.E., Degree in Computer Science and Engineering from Periyar University in the year 2002, M.E., degree in Computer Science and Engineering from Annamalai University in the year 2007. He had published over 40 journals papers and over 30 conferences papers both at National and International level. His areas of interest include Computer Networks, Network Security, Information Security, etc.