International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2021/209188 Volume 8, Issue 3, May – June (2021)

RESEARCH ARTICLE

A Hybrid Distance Vector Link State Algorithm: Distributed Sequence Number

Hussein Khayou
Department of Computing Machines Systems and Networks, National Research University "MPEI", Krasnokazarmennaya, Moscow, Russia.
hussein.khayou@gmail.com

Margarita A. Orlova
Department of Computing Machines Systems and Networks, National Research University "MPEI", Krasnokazarmennaya, Moscow, Russia.
OrlovaMA@mpei.ru

Leonid I. Abrosimov
Department of Computing Machines Systems and Networks, National Research University "MPEI", Krasnokazarmennaya, Moscow, Russia.
AbrosimovLI@mpei.ru

Received: 03 April 2021 / Revised: 07 May 2021 / Accepted: 18 May 2021 / Published: 28 June 2021

Abstract – Requirements in data centers to meet the increasing demands on traffic have evolved. There is a need for a simple, scalable routing protocol, which has the flexibility and ease of management to support large networks. Distance vector routing protocols are very simple and easy to implement but they suffer from routing loops. Link state protocols, on the other hand, have the advantages of fast convergence, area division of the routing domain, at the expense of the added complexity of implementation, configuration, and troubleshooting. A new loop free protocol is proposed in this paper that combines the simplicity of the distance vector protocols, loop freedom, and the ability to be used in large scale mesh networks as in link state protocols. The protocol uses a hybrid distance vector link state algorithm. It employs techniques from Enhanced Interior Gateway Routing Protocol (EIGRP), Babel, and Open Shortest Path First Protocol (OSPF). Simplicity, ease of implementation, and scalability make the proposed solution appropriate for large scale networks. Additionally, it can be used to perform the underlay routing in SDN (Software Defined Networks) overlay networks in place of IS-IS (Intermediate-System to Intermediate-System) protocol, which is usually used in these solutions. The combination of distance vector and link state helps to reduce the size of information in the database that is needed to be maintained by each node. It also helps to reduce the overhead and computing load after topology changes.

Index Terms – Hybrid Routing Protocol, Loop Free Routing, Distance Vector, Link State, Sequence Number, Babel, DUAL, EIGRP, OSPF.

1. INTRODUCTION

IP packet-based networks have traditionally been designed to support data transmission. However, these networks currently support a much wider range of services, such as voice, video, data, and all sorts of media. The static architecture of traditional networks is decentralized and difficult to manage, while modern networks under increased traffic demands require more flexibility and ease of management and troubleshooting [1]. To meet these requirements, new network and computing architectures have been developed, such as software-defined network (SDN), cloud computing, and network virtualization. SDN was commonly associated with the OpenFlow protocol. However, OpenFlow is now no longer the only solution as companies developed their proprietary solutions [2]. Many solutions rely on SDN overlay networks. Overlay networks provide a flexible foundation by tunneling network traffic through secure, authenticated end-to-end overlay links. The emergence of virtualization has raised the interest in introducing new tunneling protocols, for example, IEEE802.1aq (shortest path bridging), MPLS (Multiprotocol Label Switching), VXLAN (Virtual eXtensible Local Area Network) [3], Geneve (Generic Network Virtualization Encapsulation) [4], LISP (Locator ID Separation Protocol) [5], etc.

Data center networks have evolved significantly over the past decade, as have their traffic patterns. Traffic in data centers was primarily north-south, however, in current ultra large data centers, most traffic has become east-west. Thus, data centers' requirements have also evolved in terms of the topology and the routing protocol. The routing protocol should be simple with ease of implementation, in terms of programming code complexity and ease of operational support. Also, the set of

RESEARCH ARTICLE

features and protocols is limited so that they are supported by several vendors. The period of failure of the protocol or equipment is contained. Traffic engineering can be used with the routing protocol [6]. This is why, especially in terms of scalability, large enterprises have resorted to using BGP (Border Gateway Protocol) for routing in ultra large data centers [7].

In this paper, a simple hybrid distance vector link state protocol is proposed, which has the scalability and the simplicity needed in big data centers. Additionally, the protocol can perform the underlay routing in SDN overlay networks which eases the automation of managing the network. To simplify protocol operation, the distance vector algorithm is preserved and loop freedom operations are passed on to metric computation.

EIGRP was previously advertised as a hybrid routing protocol, however, Cisco Systems, Inc. now classifies it as an advanced distance vector protocol [8][9]. EIGRP is a loop free routing protocol that employs DUAL algorithm (Diffusing Update Algorithm) [10]. In DUAL a node can only change its successor if the feasibility condition is met. Feasible condition guarantees that when a node selects a feasible successor to a certain destination, it will not form a loop in routing to that destination. When a node cannot find a feasible successor, it will become active for this route and starts a diffusing computation. In diffusing computations, each active node has a finite state machine (FSM) and the route calculations are based on requests and replies. The replies are sent in a certain order enforced by the FSM. This ensures that an active node changes its successor only after the whole upstream tree of nodes have updated their distances to proper values so that no loop will be formed. Although Cisco Systems, Inc. opened EIGRP and published it with informational status [8], however, to our knowledge, it has not been adopted by other vendors. In addition, EIGRP does not support area segmentation of the routing domain.

It is shown in [11] that if routing matrix "decreases" in a single iteration, it will continue to decrease to convergence if the topology remains static afterward. This idea applies in the babel routing protocol [12], a distance vector protocol, where a non-decreasing sequence number is added to the metric. Cost now is in the form of *(s; d)* where *s* is the sequence number and *d* is the distance. When a node starves i.e., it cannot find any feasible successor to a certain destination, then it will send a request to increase the sequence number for that destination. The increase in the sequence number will cause the routing to decrease to convergence.

This technique of adding a sequence number is first used in Destination Sequenced Distance-Vector (DSDV) routing protocol [13], and it is also used in AODV (Ad-hoc on-demand protocol) [14]. However, in DSDV the sequence number is increased by the route's owner in every update. Therefore, hop count is the only feasible metric in DSDV, because updates with fewer hop counts will be always preferable, as their sequence number will be bigger. The technique of using sequence number is also used in link state protocols, where every router adds a sequence number to its link state advertisements (LSAs), and increases it when a change occurs to indicate more recent LSA information. Adding the sequence number to the metric can be modelled using the lexical product.

In Babel when a node starves (it has no feasible successor), then it will send a unicast request to increase the sequence number to the route's owner [12]. The request is sent over the non-feasible network. When the request reaches the route owner it will increase the sequence number. This has the effect of starting the calculation for this route from the beginning because the cost with a bigger sequence number is considered cheaper (smaller in accordance with the order relation). The routing will be decreasing and loop free. However, Babel is not suitable for large stable networks, because Babel relies on periodic routing table updates rather than using a reliable transport; hence, in large, stable networks it generates more traffic than protocols that only send updates when the network topology changes [12].

In this paper, a hybrid distance vector and link state algorithm that is suitable for large-scale networks is presented. The algorithm has both the advantages of simplicity and ease of implementation of distance vector protocols and the ability to be used in large-scale networks as in link state protocols. It also supports mesh routing and can be used for Layer 2 "routing" (shortest path bridging) and Layer 3 routing for both IPv4 and IPv6, with the ability of extension by adding new types of TLVs (Type-Length-Values).

The distance vector algorithm is used to compute the routes between the nodes inside the area. A sequence number is added to the metric. Starvation is resolved by using the distributed sequence number algorithm, which distributes the request to increase the sequence number until it reaches the route's owner. After that, the route's owner issues an update with the increased sequence number. Link state algorithm is used to distribute the networks (IP prefixes) within the same area and between the different areas. The combination of distance vector and link state helps to reduce the amount of link state information in the database which is needed to be maintained by each node, since prefixes instead of the links are advertised in link state. Prefixes can also be further summarized using route summarization. The routing domain can be divided into areas as in link state protocols in order to increase the scalability of the protocol, which was not available in pure distance vector protocols. The distance vector algorithm is used to compute the routes to the nodes instead of the networks connected to each node, thus, no dependency on IP connectivity is required. This also limits the

RESEARCH ARTICLE

number of updates after topology changes. Unequal load balancing can be supported by the protocol as in EIGRP, because each node maintains a list of feasible successors.

The paper is organized as follows. The next section scans related work. Definitions, terminology, and assumptions are presented in section 3. Section 4 introduces the distance vector part of the algorithm. The hybrid distance vector and link state algorithm is discussed in section 5. Section 6 explains how triggered updates, and split horizon concepts are applied in the algorithm. The message format is described in section 7. Section 8 introduces the hello protocol. The performance of the presented algorithm is compared to DUAL in section 9. Section 10 concludes the paper.

2. RELATED WORK

Composite cost consists of multiple metric components. It can be modeled using the lexicographic product [15], when route selection is based on lexicographic comparison, i.e., the most important component is considered first, then the next component, and so on. Adding the sequence number to the metric like in Babel and our proposed protocol is attributed to this type of cost. Non lexical costs such as the metric of EIGRP can be modeled using the functional product [16]. In this type of composite metric, all components are involved at the same time in the computation of the metric. Sufficient algebraic properties for convergence and convergence to optimal solutions is developed by Sobrinho in [17, 18, 19]. The inflationary property (we adopted the name from [20], and it has also been called increasing [21]) guarantees the existence of shortest paths and convergence of the shortest path algorithm. It can be explained in words, as the cost of a path cannot decrease, in respect to an order relationship, when it is extended [18]. This property is sufficient for convergence [19]. Another important property is monotonicity, which means that the order relationship between the costs of any two paths with the same origin is preserved when both are extended to the same node [18]. The later property is needed for convergence to global optimal solutions [20]. Inflationary and monotonicity combined assures that Bellman-Ford algorithm considers only loop free paths in static topologies [11], however, in dynamic topologies routing loops can happen, leading to the counting to convergence, or counting to infinity problem. It is worth mentioning that, in many works in the literature, the inflationary property is called monotonicity, while what we call monotonicity is called isotonicity [17][18][19]. In [22], sufficient algebraic properties are studied for different path calculation algorithms, and packet forwarding schemes used in wireless routing protocols.

EIGRP is a Cisco Systems, Inc. proprietary loop free protocol. It was opened in 2013 and published with informational status [8]. EIGRP is an advanced distance vector protocol based on DUAL algorithm [10]. Diffusing Computation concept was first proposed by Dijkastra and Scholten [23]. Feasible conditions for loop free routing were presented in [10]. Feasible conditions, when always met, ensure that when a node changes its successor, no loop will be formed as a result. DIC (Distance Increase Condition) was discussed in the literature prior to the work of Jaffe and Moss, however, they were first to prove that DIC is sufficient for loop freedom [24]. DIC states that a node can only change its successor if the distance using the new successor is less than the feasible distance used by the node. CSC (Current Successor Condition) and SNC (Source Node Condition) were proposed and proved correct by Garcia [10]. In CSC, a node can change its successor only if the reported distance of the new successor is less than the reported distance of the current successor, while in SNC, a node can change its successor only if the reported distance of the new successor is strictly less than the current feasible distance of the node. In [11], loop free routing in dynamic routing with a generalized metric is studied algebraically, whereas in [10] only a simple metric was considered.

EIGRP has the advantage of fast convergence and scalability in small and medium networks. EIGRP also supports unequal-cost multipath routing using the concept of feasible successors. However, EIGRP was not adopted to our knowledge by other vendors. In addition, EIGRP does not provide a mechanism to support the division of the routing domain into areas or subdomains.

Babel is a loop-avoiding distance-vector routing protocol, that uses sequence number to prevent routing loops [12]. Babel has the advantage of simplicity and small size. It is suitable for routing in highly dynamic wireless mesh networks. However, it is not well suited for large and relatively stable networks, as it depends on periodic updates. Furthermore, in Babel, the network during the reconfiguration phase can be unstable, because upstream nodes which did not receive updates that their current routes are invalid will continue to forward traffic over the invalid routes.

In [6] routing design in hyper-scale data centers is described using BGP as the only routing protocol. The use of BGP in place where commonly an IGP (Interior Gateway Protocol) is envisioned to meet the increasing requirements of growing traffic in data centers. Advantages include: less complexity in protocol design and less overhead of information flooding compared to link state IGPs, and ease of management and troubleshoot. BGP is also known for scalability as it scales for the whole internet.

However, BGP paths may have sub-optimal characteristics in terms of QoS (Quality of service). For example, alternative paths can be found that can improve end-to-end delay [25]. BGP is also prone to persistent route oscillations [26]. Because full mesh iBGP network is not scalable, alternative methods have been standardized: route reflectors and AS

RESEARCH ARTICLE

(Autonomous System) confederations. The loss of complete visibility may result in persistent route oscillations [26].

SDN OpenFlow solutions, on the other hand, can effectively improve link utilization, as shown experimentally, however, the increasing burden on the controller will lead to increasing computation time of topology updates. Hence, the centralized control mode of SDN OpenFlow creates challenges on scalability [1][27]. Recent research focuses on improving the flow scheduling algorithms for improving load balancing and link utilization. Authors in [28] has developed a server cluster based on OpenFlow. Floodlight an open-source controller has been chosen. The default Floodlight's load balancing strategy, which is round-robin, has been replaced by a dynamic weighted random selection DWRS strategy proposed by the authors of [28]. In DWRS statistics about servers' load are collected in real-time to dynamically update the server weights. The servers with higher weights have higher chance of being selected as target servers. In [29] a flow scheduling algorithm based on residual neural networks algorithm were used. Results showed that the algorithm can reduce transmission time by approximately 50%, reduce the packet loss rate from 0.05% to 0.02%, and improve bandwidth utilization by 30%, in comparison with the round-robin and weighted round-robin scheduling algorithms. Table 1 summarizes advantages and limitations of the above solutions.

Solution	Advantages	Limitations
EIGRP	Fast convergence and scalability in small and medium networks. Unequal load balancing.	Cisco Systems, Inc. only. Inability to partition the routing domain.
Babel	Simplicity and small size.	Not suitable for large relatively stable networks. Periodic updates. Network unstable during reconvergence phase.
Link State: OSPF IS-IS	Complete view of topology. IS-IS can work on layer 2 no dependence on IP.	Complexity in management and troubleshooting. Overhead during reconvergence.
BGP	Scalability. Less complexity in protocol design and less overhead of information flooding compared to link state, and ease of management and troubleshoot.	Non-optimal computed paths. Persistent route oscillations.
SDN Openflow	Ease of managnet and automation. Traffic engineering, and support of QoS.	Scalability limitations on the control plane in large networks. Single point of failure. Security attacks on the controller.

Table 1 Comparison of Existing Solutions

3. TERMINOLOGY

Connections in the network are bidirectional. A link can connect more than two routers. Every router has a unique router-id. Every router maintains a table of costs for every other router. The cost is in the form of *(s; d)*, where *s* is the sequence number which is a positive integer assigned by the route owner, and *d* is the distance. *d* can be a composite metric, and this will not affect the protocol operation, however, in this description, *d* is assumed to be a positive number. The cost (*s1; d1*) is considered cheaper (smaller or more preferable) than the cost (*s2; d2*) if *s1 > s2* or *s1 = s2^d1 ≤ d2*. Every router also assigns a feasible cost to all other routers. The feasible cost for a router is the cheapest cost learned for that router. Every router also stores the reported costs for its directly connected neighbors to all the routers in the network. We say that a neighbor *n* is one of *s*'es feasible successors to a router *d* if the reported cost of *n* for *d* is strictly cheaper than *s*'es feasible cost for *d*. We say that a feasible successor *n* is one of *s*'es successor to a router *d* if the cost from *s* to *d* is cheapest through *n*. Updates are sent over reliable connections, this is accomplished by acknowledging the receipt of an update.

4. DISTRIBUTED SEQUENCE NUMBER ALGORITHM

The main drawback of the approach used in Babel is that, when a node no longer has a feasible successor for a route, it responds by sending a *unicast* request to the route's owner to increase the sequence number, instead of sending updates to all its neighbors. Consequently, some of the upstream nodes, which did not receive updates that their current routes are not valid will still forward traffic according to their invalid routing information, and the traffic will be dropped until the new information comes from the route's owner.

To address this issue, we propose the following mechanism: the node which had a starvation (no feasible successor) for a route *d*, will send *an update* to *all* its directly connected neighbors containing *infinite distance* for *d* (route poisoning), and the *current sequence number* (only route's owner is authorized to increase the sequence number). The update

RESEARCH ARTICLE

should be marked with a *request to increase the current sequence number* of this route. The node which sent the request will not change its feasible cost (the cost used to check the feasibility condition for received updates), but it will change its distance and (reported distance) to infinity. We allow changes on the feasible cost only if the feasible cost decreases. This leads that the distance component in the feasible cost might be changed to a larger value if and only if the sequence number in the received update is bigger because costs with larger sequence numbers are considered preferable (cheaper) in the order relation. When the request to increase the sequence number is set for a router *d* with infinite distance, a timeout timer will be used for this router. If the router does not receive an update with a higher sequence number during an appropriate hold down time, the router *d* along with all its routes will be timed out and deleted from the database. This helps to solve the problem if *d* (the route's owner) is no longer connected to the network then it will be timed out and erased with all its routes. Setting the request to increase the sequence number for a router *d* is only cleared when a new update is received with a higher sequence number for *d*. After a node sets its request to increase the sequence number for *d*, all the subsequent updates for *d* will be marked with the request to increase the sequence number, until an update with a higher sequence number is received, or the router is timed out and deleted.

A node receiving this request/update will check first the router-id. If the node is the route's owner and the sequence numbers in the update/request and in the owner are equal, then the owner will *increase* the sequence number and send an update with the new sequence number to all its directly connected neighbors (split horizon should not be applied in this case). The request to increase the sequence number should be cleared in the update. Else, if the sequence number in the update/request is less than the sequence number used by the owner, the update/request should be ignored. Routes' owners never set the request to increase the sequence number for themselves.

If the node is not the route's owner, it will check the sequence number. If the sequence number in the update/request is strictly less than the sequence number in use by the current node, then the update/request will be ignored as it contains old information. If the sequence number is the same, the node will check if it has responded to a request to increase the sequence number for this route. If the node has already set the request to increase the sequence number to this route previously, then the update/request will be treated as a regular update. This means, that if the feasibility condition is met, and the cost using this update is cheaper than the cost currently used by the node, then the node will change its cost and send an update with the new cost. The feasible cost can also be changed here if the new cost is cheaper than the feasible cost. The request to increase the sequence number will be still set in the update because the node did not receive an update with a higher sequence number. This helps if some nodes have feasible routes, then they can still use them, and advertise them while waiting for the new sequence number. If the distance in the node does not change after receiving the update, then the reception of the request will not trigger updates in the node.

Else if the request to increase the sequence number was not set in the node receiving the request/update, and the sequence number in the request/update is the same, the node will set the request to increase the sequence number for this route, the node will also check if it has a feasible successor or not. If the node has a feasible successor. The node changes its distance and reported distance if needed, however, the feasible cost should not be changed if the new cost is not cheaper than the current feasible cost. The node will also send a triggered update/request to all its directly connected neighbors (respecting split horizon rules when used), the request/update will contain the new distance if the distance is changed (or the old distance if not), and the request to increase the sequence number will be set. If the node receiving the update/request does not have a feasible successor, it will change its distance and reported distance to infinity, the feasible cost is left unchanged, and it should send an update/request to all its directly connected neighbors containing the infinite distance, and the request to increase the sequence number.

If the node's request to increase the sequence number is set for a route, and it receives an update with a higher sequence number, then the node clears the request to increase the sequence number (if it is cleared in the update received, if not it will keep the request to increase the sequence number set for this route). The node in this case changes its cost, reported cost, and feasible cost, then sends a triggered update with the new cost (sequence number and distance) respecting split horizon rules if used. The feasible cost will be changed here because the new cost through the neighbor who sent the update is cheaper as it contains a bigger sequence number.

To optimize the performance of the algorithm, when a node has to send an update to its neighbors marked with a request to increase the sequence number for a node *d* (the owner). The node should choose to send the updates in ascending order *by the neighbors' reported costs* for *d* so that the neighbor with the *least reported cost* will receive the update first. This helps in propagating the request to increase the sequence number faster towards the route's owner so that the new calculations with the higher sequence number happens quicker. As a further optimization of the algorithm. If a node *i* has already received a request to increase the sequence number from a neighbor *k*, and there is no change in the sequence number nor in the distance of node *i* since the last update from *i* to *k* i.e., the current cost in node *i* is the same reported to *k*, then *i* will not include *k* in its update/request which is triggered after

RESEARCH ARTICLE

receiving a request from another neighbor say m, because *i* has no new information to be reported to *k* in this case as the cost did not change and *k* has already set the request to increase the sequence number.

The correctness of the algorithm is easily deducted from the proof of correctness of feasible conditions in [10] and its generalization in [11]. Since we only allow the feasible cost to decrease, then it will always be valid. We also do not change the successor, unless the feasible condition is respected, so the algorithm will always be loop free.

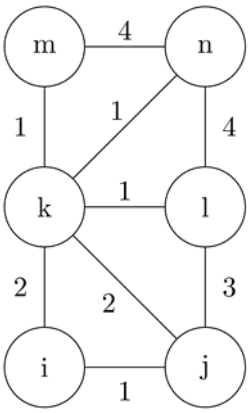

Figure 1 Simple Graph

In Figure 1, let *i* be the destination and suppose that the link between *i* and *k* is disconnected, node *k* will change its successor to j, because *j* is a feasible successor, since its distance is 1 < 2 (2 is node's *k* feasible distance), and send updates to its neighbors *l*, *n* and *m* reporting the new distance 3. Node *l* will change its successor to *j*, because *j* is also a feasible successor for *l* as 1 < 3 (3 is *l*'s feasible distance). Node *l* will send updates reporting its new distances 4 to nodes *n* and *k*. Neither *m* nor *n* has a feasible successor because each of them has a feasible distance of 3 so both of them will send a request. The request/update sent will contain the current sequence number and infinite distance. The updates will be sent to all their neighbors. When node *m* receives the update/request from *n*, this will not cause any change because node *m* has already sent a request for this route so the request is ignored. The same happens when node *n* receives the request from n. When node *k* receives the request from either *m* or *n* it will send a request containing its distance and current sequence number, as node *k* has a feasible successor. The update/request is sent to *l*, *j*, and *n*. Here, as an optimization of the algorithm, node *k* can opt out not to send a request/update to *n* if node *k* has already received the request from n. Node *l* will respond to the request received from *n* by sending update/request to node *k* and j. Here also node *l* may not send the request to *k* if it has already received a request from *k*. Suppose that *j* receives the request from *k* first. Then node *j* will send update/request to *i* and *l*. Here, also, node *j* will not send a request to *l* if it has already received the request from *l*. Node *i* is the owner, so it will increase the sequence number and send an ordinary update containing the distance 0 and the new sequence number.

5. HYBRID DISTANCE VECTOR LINK STATE ALGORITHM

If a network is connected to more than one router, then who will be the owner of this network? To address this question, we propose the following approach. The distance vectors algorithm described above will not be used for calculating distances to networks, but to the routers (or bridges if it is used in shortest path bridging). Networks are distributed in a link state manner. We do not need a separate link state protocol, because we need to flood only the information of the connected networks to each router. These pieces of information could be distributed incrementally using the same updates of the distance vector protocols. Higher sequence number means more recent information as in link state protocol. The link state information and the link sate databases will be a lot smaller than the case when a pure link state protocol is used because we distribute here only networks. Networks can also be summarized, to further reduce the link state information that should be distributed.

Areas can be formed also like any link state protocol. Routers use an initial distance of 0 for internal directly connected intra-area networks, and the distance to the closest external router (inter-area networks) owning the network for external networks. We also use different types of TLVs (Type-Length-Value) to distinguish internal routes and external routes where internal routes are always preferable.

For multiple areas, our approach is similar to OSPFv2 [30], that area 0 is the backbone area. However, each router should belong to only one area. Routers belonging to areas different than area 0 can have neighborhood relation to routers with the same area or with routers from area 0. When the routers on the link are from different areas. They do not exchange distance vector information. Each router summarizes all the known networks in its area and sends an update containing the external summary prefixes (networks). The other edge router distributes these link state pieces of information in its updates inside its area after adding the distance between the two edge routers to the distance received in the external TLVs. Edge routers from areas other than area 0 build summary external TLVs for networks residing in their area only. Edge routers in area 0 build summary external TLVs for their area and distribute received summary external TLV from other areas after modifying the distance field to reflect their distance to the external network, however, they do not send summary external TLVs of an area A to a router from the same area A. Our approach is similar to OSPFv2 multiple areas, but there is a major difference. In our approach, edge routers belong to only one area. This will decrease the load on edge routers, so

RESEARCH ARTICLE

they do not participate in the distance vector updates for multiple areas.

To calculate, the distance for networks. For intra-area networks: We first identify the routers that own the networks, then we compare the distances to these routers. Next hops will be routers with the smallest distances. For inter area networks, we identify the edge routers that advertised the external networks in the area and add the external advertised distance to the distance to the edge routers. We then identify the edge routers with the smallest aggregate distances. Next hops will be the next hops used to reach these routers (recursive routing).

It is true, that the distance vector algorithm presented in the previous section is loop free. However, the combination of link state and distance vector could cause short lived loops, which is inherent to all link state protocols. To break loops before they form, we adopt the following mechanism when a change occurs, that results in a change in the routing table to a network e.g., new link state information is received (linking a network to a router or withdrawal of a network), or change in the cost, etc., the router first invalidates its route to this network. Then, it sends its triggered updates to its neighbors. Only after it receives acknowledgments from neighbors it can install the changed routes, in its routing table.

6. TRIGGERED UPDATES, SPLIT HORIZON, AND TABLES

Updates in the protocol are triggered by change. Change can be in the cost, setting the request to increase the sequence number for a certain route, and change in the link state information. Updates should be incremental i.e., only changed components should be sent and only to the nodes that are unaware of those changes. When the links are symmetrical, an optimization technique known as split horizon should be used.

In this technique, node *i* does not send updates of changes in the link state database to node *j* if *i* has learned about the changes from *j*. Also, node *i* does not send updates about cost changes to node *j* if *j* is the successor for *i*. However, when node *i* has to send an update to *j* because of changes in the link states (setting the request to increase the sequence number, higher sequence number, etc.), then it uses the techniques of poison reverse, i.e., it reports infinite distance to its successor.

Each DSN speaker must maintain a neighbor table. The tables contain the directly connected neighbors, their router ids, sequence numbers, area ids, hello intervals, dead intervals, and area ids. Besides, the router should maintain the database table, which contains all learned routers, their router ids, sequence numbers, distance, area ids, advertised TLVs. Another important table is the topology table. This table contains, the feasible cost, which are composed of two parts (sequence number, distance) for every router in the area, the reported costs of neighbors which are also composed of (sequence number, distance). Feasible successors are the neighbors whose reported costs for a route is smaller than the feasible cost for that route.

7. MESSAGE FORMAT

The protocol message can be sent directly over link layer information using multicast link local MAC addresses. This makes it appropriate to be used for shortest path bridging and IP routing as well. Router IPv4 and IPv6 interfaces' addresses are learned using the TLVs in the update using 32 prefixes or 128 respectively if needed. Routers also may optionally advertise their addresses and MAC addresses in the Capabilities field in the Hello message. Every update message contains the router-id of the sending router as well a list of router-ids of routers that should receive this message.

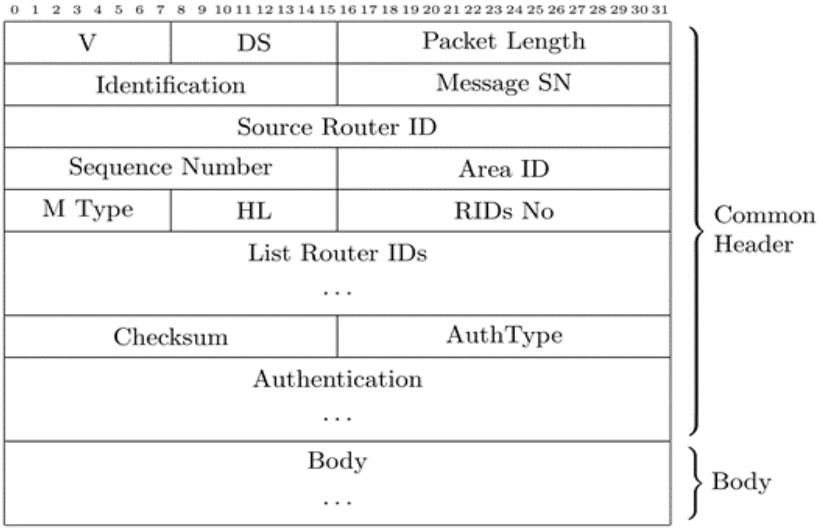

Figure 2 Common Header

RESEARCH ARTICLE

Figure 2 depicts the common header of protocol messages. The protocol should be reliable. So, message receipt should be acknowledged except for hello messages. Message identification and Message SN (Sequence Number) are echoed in the Ack Identification and Ack Message SN fields in the Acknowledgment message, which is shown in Figure 3. Acknowledgment messages are sent to one router only to avoid confusion.

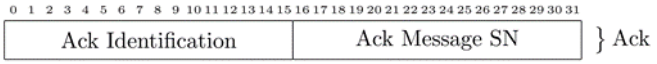

Figure 3 Acknowledgement

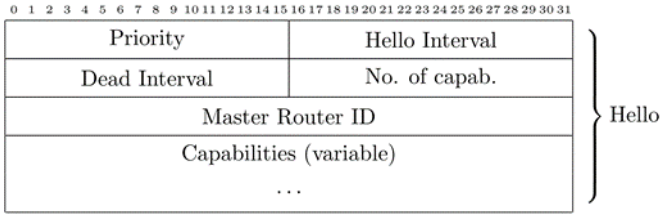

Figure 4 Hello Message

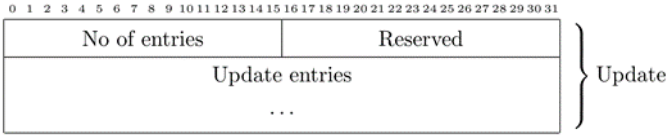

Figure 5 Update Message

The hello message, which is shown in Figure 4, should be sent each hello interval to keep the link between the routers alive. Another function is to elect a master router for non-point-to-point links. The master will distribute the link state information (networks) for new routers that join the link. Other routers also should send their distances and sequence numbers to other routers in the area. Capabilities can include router physical and logical addresses and protocol extensions. Update messages (Figure 7 consists of update entries (Figure 7). Each update entry contains a router-id owning the entry and its sequence number. The distance from the current router to the owner. R flags should be set when a request to the owner to increase the sequence number.

IPv4 addresses and IPv4 prefixes are distributed using the IPv4 TLV (Figure 7). The R means retraction, this should be set when the router has lost connection to a network and wants to remove it from the database. The O flag means overwrite, this flag should be set to overwrite information, which is needed in some type of TLVs. External IPv4 TLVs (Figure 8) contain information about the edge routers that created them. Edge Router-ID, Edge Area ID, and Edge Sequence number belong to the first edge router that created the TLV. Other edge routers change the Distance field when diffusing the external TLV in their area.

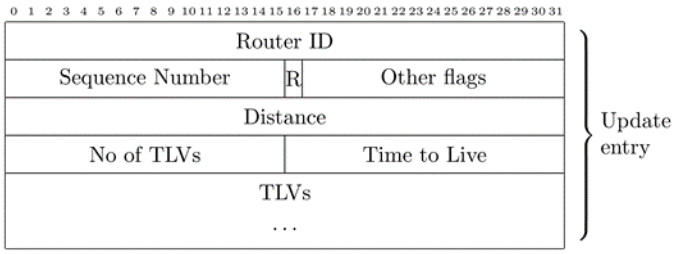

Figure 6 Update Entry

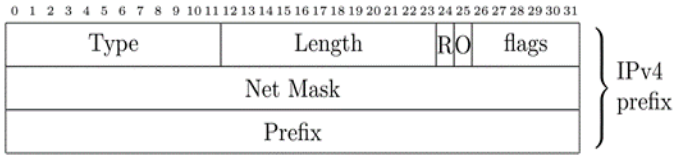

Figure 7 IPv4 Prefix TLV

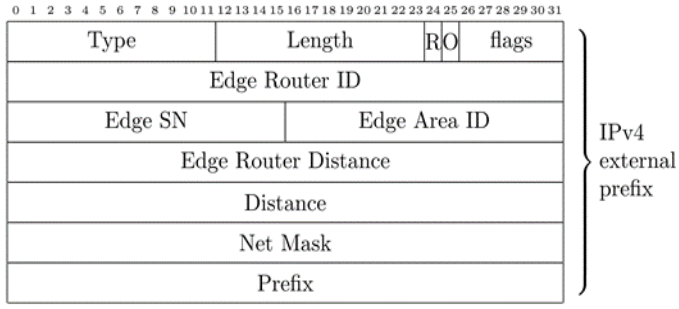

Figure 8 IPv4 External Prefix TLV

8. HELLO PROTOCOL AND NEIGHBOR ACQUISITION

When a DSN speaker starts the DSN protocol. It will first send a hello message containing its router-id and sequence number (which is randomly generated). If there is a master router on the link. Routers will send a hello message containing the list of router IDs connected to the link. The master will also send a triggered update containing the list of routers in the area, their sequence numbers, distance, and link state information (connected networks). Other routers connected to the link will send also their distance vector updates containing the routers in the area (IDs, sequence numbers, distances).

If the database already contains the router-id of the new router, then the router checks the records of its router-id in the database. It can then send an update with an increased sequence number confirming the connected networks. It also retracts networks, which are no longer directly connected to the router, and adds the new connected networks. The update should also contain the distance vector information, routers-ids in the area, and the router distance to these routers.

If the above mechanism is not utilized. The new router will send then an update with the new chosen sequence number. The old information of the router (distance vector and link

RESEARCH ARTICLE

state) if present will be timed out because a request to increase the sequence number must have been propagated when the router had failed in the network, and no update is received containing the increased sequence number. As the new sequence will be different from the old sequence number because it is chosen randomly. Routers will check the difference between the sequence numbers. If the difference between the sequence numbers is larger than a configured value, the sequence number will be considered new.

When a router receives a hello message for the first time, it will respond by sending a hello containing the router-id of the other router. Router with the highest priority, or when there is a tie, the router with the highest router-id will become the master of the link, and its router-id will be included in the Master id in the hello message. The election happens once, a new election occurs only if the master disconnects. Masters will be responsible for distributing link state information for new routers joining the link. When a router receives new information, it will send it to the master and the master will send it to the other nodes on the link.

9. PERFORMANCE COMPARISON OF THE DISTANCE VECTOR ALGORITHM WITH DUAL

The convergence time of the suggested distance vector algorithm DSN is comparable with DUAL. In Dual, when a node becomes active, updates are propagated mainly in the unfeasible network back and forth, after the whole upstream nodes updates their distances, the active node can become passive again. In contrast, in our algorithm, the request to increase the sequence number is propagated towards the route's owner in the feasible network, then updates go back towards the node with the new sequence number.

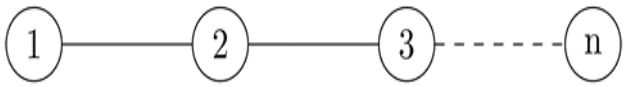

Figure 9 Simple Graph 2

Let us consider the simple topology depicted in figure 9. Let us assume that an increase occurs in the link cost between node 1 and node 2 so that node 3 has no feasible successor for routing towards node 1. This corresponds to one of the worst-case scenarios in DUAL, as a diffusing computation will be started in node 3, this diffusing computation will be propagated in the upstream tree until we reach node *n* then the replies go back until it reaches back to node 3, so that node 3 is the last node to go passive.

However, in DSN, node 3 send a request to increase the sequence number, this request is propagated downstream so it reaches node 1, and it is propagated upstream so that all nodes with no feasible routes invalidate their routes (as there is no feasible successor). After the request reaches node 1, the updates with the higher sequence number are propagated upstream. So, the convergence time in this case of DSN is better than DUAL, however, the number of messages exchanged in DSN is bigger.

A second case where the increase in distance occurs in the link between node 2 and 3, so that node 1 has no feasible successor in routing towards node n. This case corresponds to DUAL's best-case scenario, as node 1 will go passive again directly after the reply is received from node 2 because there are no upstream nodes. However, in DSN, this corresponds to the worst-case scenario, because the request to increase the sequence number will be propagated from node 1 towards node n. After that, the updates with the higher sequence number will be propagated backward until it reaches node 1. Here, the convergence time in DUAL is better than DSN, however, as in DUAL only node 1 is affected by the change. The number of messages exchanged in DSN is also bigger than the one with DUAL.

From the above discussion, convergence time for a node in DSN is twice the time required to exchange messages over the shortest path from the node to the destination, while in DUAL it is twice the time required to exchange messages over the longest path in the upstream tree. Hence, the convergence time of DSN is comparable to DUAL, however, the number of messages exchanged in DSN is larger, because in DSN messages are exchanged in the feasible and unfeasible networks, while in DUAL messages are exchanged primarily in the unfeasible network.

The number of exchanged messages in DSN when a starvation occurs is twice the number of exchanged messages in the original distributed Bellman-Ford algorithm. However, DUAL applies to networks while DSN applies to the nodes themselves. This will limit the number of exchanged messages in DSN. Furthermore, the routing domain can be divided in DSN into areas, which helps to reduce the domain where the distance vector messages are exchanged.

10. CONCLUSION

The algorithm in the presented protocol is a hybrid distance vector link state algorithm. This way, the protocol will have a much smaller database to maintain than other link state protocols. This is because we only advertise the networks of the connected links rather than the links themselves, and networks can also be further summarized using route summarization. The protocol is also a distance vector for routing nodes (not the networks themselves), so this will limit the number of updates when the topology changes, and large topologies can be divided into areas, which was not possible in existing distance vector protocols. When multiple areas are used, all routers (including edge routers) belong to only one area. Thus, edge routers will be less loaded because they only participate in distance vector updates for their areas.

RESEARCH ARTICLE

Another advantage is that the algorithm is simpler compared to the EIGRP protocol, since we do not need a finite state machine, as in EIGRP, instead, all updates are sent according to the principles of the distance vector algorithm. Since the algorithm calculates the distances to the routers themselves and advertises the networks in a link state manner, the protocol has built-in support for mesh routing. However, the presence of the hello protocol, and the requirement of reliable transport for updates, make it difficult for the protocol to scale in wireless ad-hoc network routing. The protocol is more suitable for large-scale networks, where network maintenance requires simplicity and ease of management.

REFERENCES

[1] Benzekki, Kamal, et al. "Software-Defined Networking (SDN): A Survey." Security and Communication Networks, vol. 9, no. 18, 2016, pp. 5803–33. Wiley Online Library, doi:https://doi.org/10.1002/sec.1737

[2] Raza MH, Sivakumar SC, Nafarieh A, Robertson B. "A comparison of software defined network (SDN) implementation strategies". Procedia Computer Science 2014;32:1050{5. doi:10.1016/j.procs.2014.05.532

[3] Sridhar, T., et al. Virtual EXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks. Tech. Rep., Aug. 2014. [Online]. Available: https://doi.org/10.17487/rfc7348

[4] Gross, Jesse, et al. Geneve: Generic Network Virtualization Encapsulation. Tech. Rep., Nov. 2020. [Online]. Available: https://doi.org/10.17487/rfc8926

[5] Lewis, Darrel, et al. The Locator/ID Separation Protocol (LISP). Tech. Rep., Jan. 2013. [Online]. Available: https://doi.org/10.17487/rfc6830

[6] Premji, Ariff, et al. "Use of BGP for Routing in Large-Scale Data Centers". Tech. Rep., Aug. 2016. [Online]. Available: https://doi.org/10.17487/rfc7938

[7] Medhi, Deepankar, and Karthik Ramasamy. Network Routing: Algorithms, Protocols, and Architectures. 2nd edition, Elsevier, Morgan Kaufmann Publishers, an imprint of Elsevier, 2018.

[8] D. Savage, J. Ng, S. Moore, D. Slice, P. Paluch, and R. White, "Cisco's enhanced interior gateway routing protocol (EIGRP)," Tech. Rep., May 2016. [Online]. Available: https://doi.org/10.17487/rfc7868

[9] A. Bruno, CCIE Routing and Switching Exam Certification Guide, ser. Certification and training series. Cisco Press, 2002. [Online]. Available: https://books.google.ru/books?id=NzYb1pPZTB0C

[10] J. Garcia-Lunes-Aceves, "Loop-free routing using diffusing computations," IEEE/ACM Transactions on Networking, vol. 1, no. 1, pp. 130–141, 1993. [Online]. Available: https://doi.org/10.1109/90.222913

[11] H. Khayou, M. A. Rudenkova, and L. I. Abrosimov, "On the algebraic theory of loop free routing," in Distributed Computer and Communication Networks. Springer International Publishing, 2020, pp. 161–175. [Online]. Available: https://doi.org/10.1007/978-3-030-66471-8_14

[12] J. Chroboczek and D. Schinazi, "The babel routing protocol," Tech. Rep., Jan. 2021. [Online]. Available: https://doi.org/10.17487/rfc8966

[13] C. E. Perkins and P. Bhagwat, "Highly dynamic destinationsequenced distance-vector routing (DSDV) for mobile computers," ACM SIGCOMM Computer Communication Review, vol. 24, no. 4, pp. 234–244, Oct. 1994. [Online]. Available: https://doi.org/10.1145/190809.190336

[14] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," Tech. Rep., Jul. 2003. [Online]. Available: https://doi.org/10.17487/rfc3561

[15] A. J. T. Gurney and T. G. Griffin, "Lexicographic products in metarouting," in 2007 IEEE International Conference on Network Protocols. IEEE, Oct. 2007. [Online]. Available: https://doi.org/10.1109/icnp.2007.4375842

[16] H. Khayou and B. Sarakbi, "A validation model for non-lexical routing protocols," Journal of Network and Computer Applications, vol. 98, pp. 58–64, Nov. 2017. [Online]. Available: https://doi.org/10.1016/j.jnca.2017.09.006

[17] J. Sobrinho, "Algebra and algorithms for QoS path computation and hop-by-hop routing in the internet," in Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213), vol. 2. IEEE, 2001, pp. 727–735 vol.2. [Online]. Available: https://doi.org/10.1109/infcom.2001.916261

[18] J. L. Sobrinho, "Network routing with path vector protocols," in Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications – SIGCOMM '03. ACM Press, 2003. [Online]. Available: https://doi.org/10.1145/863955.863963

[19] J. Sobrinho, "An algebraic theory of dynamic network routing," IEEE/ACM Transactions on Networking, vol. 13, no. 5, pp. 1160–1173, Oct. 2005. [Online]. Available: https://doi.org/10.1109/tnet.2005.857111

[20] T. G. Griffin, "Lecture notes in an algebraic approach to internet routing," 2010. [Online]. Available: https://www.cl.cam.ac.uk/teaching/1011/L11/

[21] T. G. Griffin and A. J. T. Gurney, "Increasing bisemigroups and algebraic routing," in Relations and Kleene Algebra in Computer Science. Springer Berlin Heidelberg, 2008, pp. 123–137. [Online]. Available: https://doi.org/10.1007/978-3-540-78913-0_11

[22] Y. Yang and J. Wang, "Design guidelines for routing metrics in multihop wireless networks," in IEEE INFOCOM 2008 - The 27th Conference on Computer Communications. IEEE, Apr. 2008. [Online]. Available: https://doi.org/10.1109/infocom.2008.222

[23] E. W. Dijkstra and C. Scholten, "Termination detection for diffusing computations," Information Processing Letters, vol. 11, no. 1, pp. 1–4, Aug. 1980. [Online]. Available: https://doi.org/10.1016/0020-0190(80)90021-6

[24] J. Jaffe and F. Moss, "A responsive distributed routing algorithm for computer networks," IEEE Transactions on Communications, vol. 30, no. 7, pp. 1758–1762, Jul. 1982. [Online]. Available: https://doi.org/10.1109/tcom.1982.1095632

[25] Brugnoli, Ignacio, et al. "Tunnelless SDN Overlay Architecture for Flow Based QoS Management." 2021 24th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), IEEE, 2021, pp. 62–69. DOI.org (Crossref), doi:10.1109/ICIN51074.2021.9385539

[26] McPherson, D., et al. Border Gateway Protocol (BGP) Persistent Route Oscillation Condition. Tech. Rep., Aug. 2002. [Online]. Available: https://doi.org/10.17487/rfc3345

[27] Yu, Chen, et al. "Intelligent Optimizing Scheme for Load Balancing in Software Defined Networks." 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), IEEE, 2017, pp. 1–5. DOI.org (Crossref), doi:10.1109/VTCSpring.2017.8108541

[28] Chiang, Mei-Ling, et al. "SDN-Based Server Clusters with Dynamic Load Balancing and Performance Improvement." Cluster Computing, vol. 24, no. 1, Mar. 2021, pp. 537–58. Springer Link, doi:10.1007/s10586-020-03135-w

[29] Liu, Yazhi, et al. "Load Balancing Oriented Predictive Routing Algorithm for Data Center Networks." Future Internet, vol. 13, no. 2, Feb. 2021, p. 54. www.mdpi.com, doi:10.3390/fi13020054

[30] J. Moy, "OSPF version 2," Tech. Rep., Apr. 1998. [Online]. Available: https://doi.org/10.17487/rfc2328.

RESEARCH ARTICLE

Authors

Khayou Hussein - Postgraduate Student Dept. of Computing Machines, Systems and Networks National Research University "Moscow Power Engineering Institute" (111250, Russia, Moscow, Krasnokazarmennaya 14, email: hussein.khayou@gmail.com, ORCID: https://orcid.org/0000-0002-9790-5871).

Orlova Margarita A. – Assistant Dept. of Computing Machines, Systems and Networks, National Research University "Moscow Power Engineering Institute" (111250, Russia, Moscow, Krasnokazarmennaya 14, email: OrlovaMA@mpei.ru, ORCID: https://orcid.org/0000-0002-8214-4117).

Abrosimov Leonid I. - Dr.Sc.Eng., Professor Dept. of Computing Machines, Systems and Networks, National Research University "Moscow Power Engineering Institute" (111250, Russia, Moscow, Krasnokazarmennaya 14, email: AbrosimovLI@mpei.ru, ORCID: https://orcid.org/0000-0001-6171-8559)

How to cite this article:

Hussein Khayou, Margarita A. Orlova, Leonid I. Abrosimov, "A Hybrid Distance Vector Link State Algorithm: Distributed Sequence Number", International Journal of Computer Networks and Applications (IJCNA), 8(3), PP: 203-213, 2021, DOI: 10.22247/ijcna/2021/209188.