

Iterative Discovery of Multiple Disjoint Paths in Switched Networks with Multicast Frames

Diego Lopez-Pajares, Joaquin Alvarez-Horcajo, Elisa Rojas, Juan A. Carral, Guillermo Ibanez

Departamento de Automatica, University of Alcalá, Alcalá de Henares (Madrid), Spain.

Email: diego.lopezp@uah.es, j.alvarez@uah.es, elisa.rojas@uah.es, juanantonio.carral@uah.es and guillermo.ibanez@uah.es

Abstract—Routing via multiple paths offers many advantages such as traffic separation, congestion reduction and quality of service policies in enterprise, campus and data center networks. Disjointness of the multiple paths may provide extra benefits like increased reliability and improved traffic separation. We propose DMDP, a layer two protocol that discovers multiple disjoint paths in a switched network by exploration of the network with multicast frames. DMDP provides fast discovery of node-disjoint or link-disjoint paths, selecting the fastest disjoint paths available without any route computation over the network graph. Paths obtained by DMDP are similar to the multiple paths computed with a k-shortest path algorithm, but much simpler to find. Multiple disjoint path search may be implemented as a proactive, reactive on-demand or hybrid service at edge switches.

Index Terms—Multiple disjoint path routing, Exploring tree, Spanning Tree, Shortest Path Bridges, Ethernet, Routing bridges

I. INTRODUCTION

Routing only via the shortest paths does not provide maximum performance in networks because these paths might become overloaded. To prevent this, multipath routing uses multiple alternative paths through a network. Besides reducing congestion, multipath routing can provide fault tolerance, increased throughput and improved security. Multiple paths may be node-disjoint, edge-disjoint or partially overlapped. Among the available proposals, we focus on disjoint multipath protocols in layers two and three, as they are the most relevant for data center and enterprise networks. Wireless networks are set aside as they represent other casuistry [1].

ECMP and multiple VLANs are the basis for randomized load balancing currently used in data centers in layer three (OSPF) and layer two (SPB [2], TRILL [3]) protocols. But randomized load balancing is usually unfair [4] and often creates hot-spots surrounded by low loaded links. Additionally, most protocols neither guarantee the edge or node disjointness of the paths, nor take into account the actual traffic load on network links to select the paths.

This paper describes Iterative Discovery of Multiple Disjoint Paths (DMDP), a layer-two protocol that uses network exploration mechanisms to find node-disjoint or edge-disjoint paths between edge switch pairs of a switched network. Section II describes the protocol, Section III the evaluation, Section IV the related work, and Section V the conclusion.

II. DMDP PROTOCOL

A. The Disjoint-Paths Problem and DMDP

The definition of the disjoint paths problem is as follows: Given a graph $G = (\mathcal{N}, \mathcal{L})$ of a set $\mathcal{N}$ of N nodes, a set $\mathcal{L}$ of L links, and two nodes $s, t \in \mathcal{N}$. For $k > 0$, find k paths $P_1, P_2, \dots, P_k$ from s to t , which do not share any common nodes (or links). The problem is NP-complete in both cases [5]. DMDP is designed as a distributed protocol that finds multiple paths between edge switches, having as a metric the network latency. The environment to be deployed are switched networks, and its possible uses cases are varied, as for example multipath routing with MPTCP, QoS routing or resilience. Nevertheless, these scenarios are out of the scope of this paper. In short, DMDP finds a specified number (or less) of multiple paths with the desired restrictions of node-disjointness or edge-disjointness between two edge bridges. Forwarding between edge bridges can be performed via encapsulation/decapsulation of the user frames at the source/destination edge nodes.

B. First path set-up

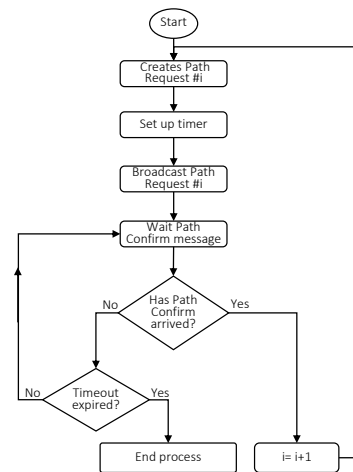


Fig. 1: Basic control of disjoint paths set up

The start protocol sequence is shown in Figure 1. The edge switch (A), when requested to find multiple path towards switch (H), multicasts a Path Request packet, containing the type of disjoint paths requested, through all its output links (except the link receiving the request) and waits to receive a unicast Path Confirm from destination switch. The process is

repeated until the number of paths found reaches its maximum limit or until timed out.

Figure 2 shows the pseudocode with the protocol logic that process the received control frames. For each incoming Ethernet frame, the EtherType value (type/length field) is checked. If the EtherType value corresponds to DMDP, the packet is processed according to the packet type: *Path Request*, *Path Confirm* or *Path Delete*. The sub consequent actions are, depending on the packet type, origin and destination address of frame and their presence or not in the routing table, associated to a path: confirmation of a path record in table, activate timer for path confirmation, refresh path timer, delete path record from table, send Path Confirm or Path Request.

```

function packet_analisys
  if ethertype==DMDP_ethertype
    case Path_request packet
      Path_request_analisys
    case Path_confirm packet
      Path_confirm_analisys
    case Path_delete packet
      Path_delete_analisys
  else
    if path is known
      resend packet
    else
      standard retransmission
  return

function Path_req_analisys
  if SRC and DST MAC addresses pair is in table
    case node mode
      discard packet
      return
    case link mode
      if path id is the same
        discard packet
        return
      else
        create new table entry: associates SRC and DST MAC
          addresses with packet input port and path id
        refresh timer
        broadcast the packet through all ports less the
          input port
        return
  else
    create new table entry: associates SRC and DST MAC
      addresses with packet input port and path id
    refresh timer
    broadcast the packet through all ports less the input
      port
    return
  return

function Path_confirm_analisys
  complete table entry, by adding packet input port.
  case non-edge sw
    forward the packet to the edge-sw
  case edge-sw
    start the next exploration sequence
  return

function Packet_delete_analisys
  case path delete concrete
    remove concrete table entry
  case all path delete
    Remove all entries
  return

```

Fig. 2: DMDP protocol logic

Figure 3a illustrates the first flooding of the Path Request packet. Switch A sends through all links the multicast frame.

Every switch receiving it (B,C,G) associates the source address to the input port and forwards it through all links afterwards. The mechanism used at switches to prevent loops and broadcast storms is based on locking the association of the source address to the first port receiving the frame for a short time, as in All-Path switches [6].

C. Set-up of additional disjoint paths

1) *Edge disjoint paths*: Figure 3 shows the complete path setup process. Once the setup of the first path is completed, switch A proceeds to create the remaining disjoint paths, sending a Path Request with Path ID #2. If the type of the requested path is edge-disjoint, switch A sends the Path request indicating it in the path type field through all the ports except the port associated to Path #1. This prevents that the requested path uses links already assigned to Path #1. Every switch receiving the Path Request for Path #2 proceeds in the same way as for the first path, with the only difference of not forwarding the Path Request via the link assigned to Path #1. When the Path Request of Path #2 reaches destination switch H, it is also confirmed with a Path Confirm packet. The process is analogous for the rest of paths.

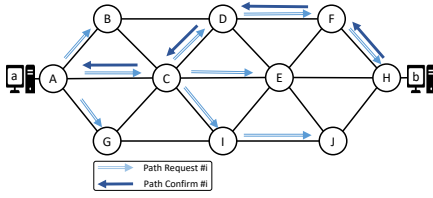
2) *Node disjoint paths*: Figure 4 displays the whole path setup process. If the requested path is node-disjoint, the procedure is similar to the previous subsection. Switch A sends the Path Request packet with the Path Type field set to node-disjoint value. The behavior of the first path is equal to the previous example. From Path #2 and successive, the switches discard the packet if they have a previous confirmed path, otherwise, they forward the Path Request packet through all ports except the input port. The path confirmation process is the same as for edge-disjoint.

3) *Maximum number of disjoint paths*: The maximum number of edge-disjoint paths is determined by the maximum number of source destination pairs (s_i, t_i) that can be routed in graph G via mutually edge-disjoint s_i-t_i paths. The edge-connectivity version of Menger's theorem [7] states that maximum number of pairwise edge-independent paths from s_i to t_i is equal to the size of the minimum edge cut for t and s (i.e. the minimum number of edges whose removal disconnects t from s). The node-connectivity version, for the node-disjoint paths, states that the maximum number of node disjoint paths in a network is equal to the minimum number of nodes whose removal disconnects s_i from t_i .

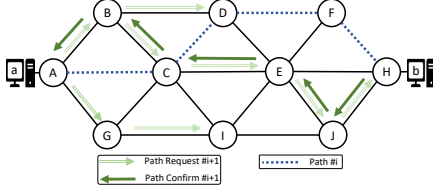
DMDP protocol is topology agnostic, the limit for the maximum number of disjoint paths to be requested is set just with the degree of the edge node having lowest degree of the two. Both edge switches know this limit because in the Path Request and Path Confirm packets a field contains the maximum number of links (i.e. possible paths) that connect the sending edge switch to the network.

D. Path deletion

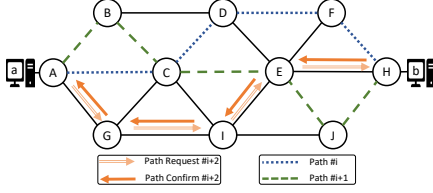
Established paths between two edge switches can be deleted selectively or all at once sending respectively Delete Path #ID or Delete All Path packets. The Delete Path packet is sent on



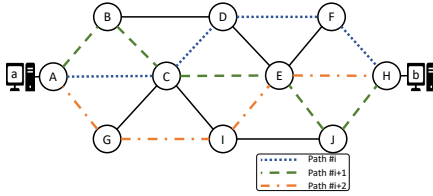
(a) Discovery and confirmation of Path #1.



(b) Discovery and confirmation of Path #2.

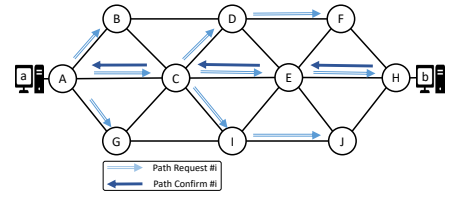


(c) Discovery and confirmation of Path #3.

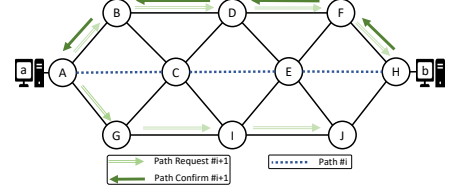


(d) The three edge disjoint paths confirmed

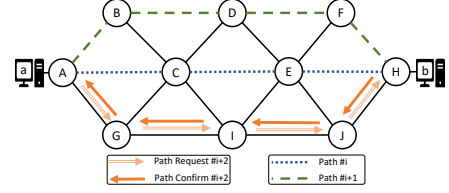
Fig. 3: Discovery of edge disjoint paths.



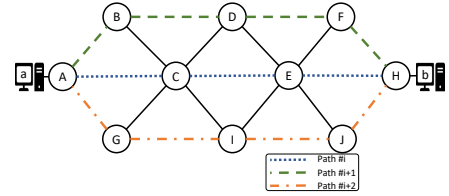
(a) Discovery and confirmation of Path #1.



(b) Discovery and confirmation of Path #2.



(c) Discovery and confirmation of Path #3.



(d) The three edge disjoint paths confirmed

Fig. 4: Discovery of node disjoint paths.

a unicast frame via the path to be deleted. The path is deleted as the packet progresses toward the destination. The Delete All Paths packet is sent as multicast frame through all links.

E. End to end routing

To be able to reach the hosts, edge switches need to know the edge switch address of every destination host. Similarly to TRILL, edge switches can build and maintain a complete host-edge switch list via a periodic multicast announcement packet sent from every edge bridge, containing the list of the MAC addresses of its directly connected hosts.

III. EVALUATION

In this section, we compare DMDP with other disjoint-path routing solutions in terms of the number of paths found.

A. Paths set-up time

The minimum time to set up the paths with DMDP is the product of the number of paths to be found and the RTT (measured between edge switches, not between hosts) of the network. The maximum time spent on search is determined by the search timer value. A conservative value for RTT in a data

center is 1 ms [8]; so if we assume 2 ms per path, the setup time for 4 disjoint paths will be 8 ms. Path setup time increases linearly with the network diameter (6-8 hops typically [9]), so it does not grow significantly with network size. Latency of the routing service is null if implemented either as a proactive or as a reactive on-demand routing service.

B. Experiments

We aim to check if our protocol can find disjoint paths in network topologies. For that, we construct a testbed in two different environments. The first one consists of the implementation of a modified Naive LBA [10] algorithm in Matlab, as it is a simplified version of Bhandari's Algorithm to find disjoint paths [11]. The second one is the node-disjoint DMDP protocol deployed using the Mininet SDN platform, due to its convenience for fast development and testing time. The protocol logic is executed at the OpenFlow controller (Ryu), which provides results to validate the protocol proof-of-concept. This centralized operation does not provide realistic results regarding path setup times because path selection is delayed by the need of every switch to access the controller at every multicast frame hop. Also, we generate different

topologies with BRITE topology generator [12] to compare the same topology in both scenarios.

For each topology we obtain the node-disjoint paths between 10 pairs of nodes randomly chosen. Figure 5 shows the paths discovered in a random topology between the nodes 3 and 9. Both, the Matlab algorithm and the DMDP protocol, obtain the same paths. Nevertheless, the discover order differs as the DMDP protocol bases its search on minimum latency paths while the Matlab algorithm computes over the k-shortest paths.

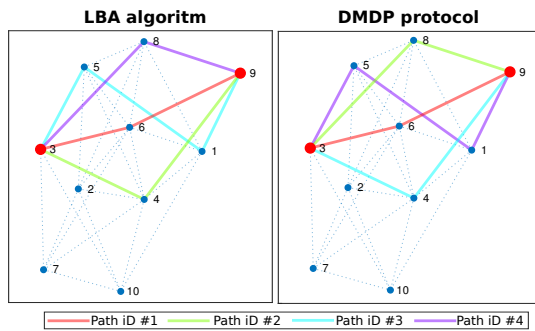


Fig. 5: Node-disjoint paths discovered in a random topology

The same happens in the other 9 pair of nodes, i.e. the paths discovered are equal in both alternatives and the discover order differs as in the previous example. We repeat the experiment with large topologies (up to 30 nodes with connection degree per node from 3 to 7 neighbors). The DMDP protocol achieves an effectiveness ratio between 60 to 97 percent compared with the Naive LBA algorithm. This variability is provoked by the exploration process (in larger topologies is probable that the fastest path is not the shortest path, thus, this path crosses more nodes, preventing the creation of new possible node-disjoint paths). This ratio will be better in link-mode operation because paths might share the same node (but not the same link).

IV. RELATED WORK

The disjoint paths problem was studied by Suurballe, first in [13] and then in [14], with an algorithm that finds two disjoint paths in a directed graph. Bhandaris algorithm [11] simplifies the implementation of Surballe's algorithm. Equal Cost Multipath (ECMP) [15] is the reference protocol for multiple paths at IETF and IEEE. ECMP is insensitive to link load and may create hot spots, although MPTCP improves its performance. In layer two, IEEE standard Shortest Path Bridges [2] implement equal cost multipath routing over symmetrical paths via precise identification of the chain of traversed nodes. The need to apply selection priorities identically in opposite path directions multiplies by N the computational complexity to $O(N^3)$. TRILL [3] implements a similar approach using also an adapted IS-IS protocol. AntNet [16] is based on swarm routing, and focuses on adaptive routing.

A procedure for discovery of multiple paths in wired networks is described in [17]. The response packets to the sender report back with data to assist in the selection of the shortest

paths. However, the protocol does not guarantee disjointness of any kind.

V. CONCLUSION

DMDP is a layer-two multiple disjoint path protocol based on network exploration that provides simple mechanisms for routing without path computation, then suitable for large data center networks. Initial results show effectiveness with low complexity. As future work, our priority is the link-mode operation and the detailed evaluation in loaded data center scenarios, with regular and irregular topologies, where the load adaptivity and scalability may be particularly important.

ACKNOWLEDGMENT

This work was partially funded by a grant from Comunidad de Madrid through Project TIGRE5-CM (S2013/ICE-2919) and the "Formacion del Profesorado Universitario (FPU)" program from the University of Alcala. Thanks to Luis Felix Rodriguez-Cano, Alvaro Molinero and Alberto Garcia-Merino for their contributions to the evaluation.

REFERENCES

- [1] B. Ismail, F. Hacene, Z. Ouadoudi, and A. Driss, "Carrier sense aware multipath geographic routing protocol," *Wireless Communications and Mobile Computing*, vol. 16, no. 9, pp. 1109–1123.
- [2] IEEE, "802.1aq - Shortest Path Bridging," 2006.
- [3] R. I. L. Perlman and D. H. T. Eastlake, "Introduction to TRILL," *The Internet Protocol Journal*, vol. 14, no. 3, pp. 2–19, 2011.
- [4] M. Al-Fares, S. Radhakrishnan, and B. Raghavan, "Hedera: Dynamic Flow Scheduling for Data Center Networks," *Nsdi*, p. 19, 2010.
- [5] M. Middendorf and F. Pfeiffer, "On the complexity of the disjoint paths problem," *Combinatorica*, vol. 13, no. 1, pp. 97–107, 1993.
- [6] E. Rojas, G. Ibanez, J. M. Gimenez-Guzman, J. A. Carral, A. Garcia-Martinez, I. Martinez-Yelmo, and J. M. Arco, "All-Path bridging: Path exploration protocols for data center and campus networks," *Computer Networks*, vol. 79, no. 0, pp. 120 – 132, 2015.
- [7] R. Aharoni, "Menger's Theorem for Graphs Containing no Infinite Paths," *European Journal of Combinatorics*, vol. 4, no. 3, pp. 201–204, 1983.
- [8] V. Vasudevan, A. Phanishayee, H. Shah, E. Krevat, D. G. Andersen, G. R. Ganger, G. A. Gibson, and B. Mueller, "Safe and effective fine-grained tcp retransmissions for datacenter communication," in *ACM SIGCOMM computer communication review*, vol. 39, no. 4. ACM, 2009, pp. 303–314.
- [9] C. Guo, G. Lu, H. J. Wang, S. Yang, C. Kong, P. Sun, W. Wu, and Y. Zhang, "Secondnet: a data center network virtualization architecture with bandwidth guarantees," in *Proceedings of the 6th International Conference*. ACM, 2010, p. 15.
- [10] Y. Guo, F. Kuipers, and P. Van Mieghem, "Link-disjoint paths for reliable QoS routing," *International Journal of Communication Systems*, vol. 16, no. 9, pp. 779–798, 2003.
- [11] R. Bhandari, "Optimal diverse routing in telecommunication fiber networks," *Proceedings of INFOCOM '94 Conference on Computer Communications*, no. 908, pp. 1498–1508, 1994.
- [12] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: Universal topology generation from a user's perspective," Boston University Computer Science Department, Tech. Rep., 2001.
- [13] J. W. Suurballe, "Disjoint paths in a network," *Networks*, vol. 4, no. 2, pp. 125–145, 1974.
- [14] J. W. Suurballe and R. Tarjan, *A Quick Method for Finding Shortest Pairs of Disjoint Paths*, 6 1984, vol. 14.
- [15] Network Working Group, "Analysis of an Equal-Cost Multi-Path Algorithm RFC 2992," pp. 1–8, 2000.
- [16] M. Blesa and C. Blum, "Ant Colony Optimization for the Maximum Edge-Disjoint Paths Problem," in *Applications of Evolutionary Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 160–169.
- [17] C. Minkenberg and M. R. Gusat, "Multipath discovery in switched ethernet networks," Jan. 31 2012, uS Patent 8,107,482.