مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (34) العدد (6) 2012
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (34) No. (6) 2012

Multicast Routing Algorithms Used in Path Computation Elements

Dr. Jafar Al-Kheir*

(Received 31 / 5 / 2012. Accepted 9 / 8 / 2012)

□ ABSTRACT □

The scalability and confidentiality constraints are particularly issues for the optimal point-to-point path computation in a multi-domain environment. Path Computation Element (PCE) has been proposed by Internet Engineering Task Force (IETF) to compute a network path or route based on a network graph and applying some constraints during the computation. In this paper, the major issues of supporting the multicast service in a multi-domain environment are discussed. Three investigated routing algorithms, which can be used by PCE, are proposed and investigated to compute multicast distribution trees in a multi-domain topology. These algorithms are Shortest Path in Each Domain (SPED), Shortest Path for All Domains 1 (SPAD1) and Shortest Path for All Domains 2 (SPAD2). At the end, the implementation and the performance evaluation of the investigated algorithms are given. We evaluate the performances of proposed algorithms by comparing different metrics, such as, the link resource usage, the node resource usage and the end-to-end hop count. From the results, we can see that the SPED algorithm is simple and its calculation time is short, but may not find the optimal end-to-end paths in multi-domain topologies.

Keywords: Path Computation Element, Multi-domain, Performance Evaluation, Shortest Path Tree, Multicast.

***Assistant Professor, Department of Computer and Control Engineering, Tishreen University, Lattakia-Syria .**

خوارزميات التوجية لأنظمة الإرسال المتعدد المستخدمة ضمن عناصر حساب المسار

الدكتور جعفر الخير[*]

(تاريخ الإيداع 31 / 5 / 2012. قُبِل للنشر في 9/ 8 / 2012)

□ ملخّص □

تُعَدّ قابلية التوسع (scalability) بالإضافة إلى قيود الوثوقية (confidentiality constraints) من أهم المعوقات التي تواجه طرق حساب المسار الأمثل بين عقدتين ضمن الشبكات المتعددة النطاقات. من أجل ذلك تم اقتراح وحدة حساب المسار (Path Computation Element PCE) من قبل الفريق الهندسي للانترنيت ((Internet Engineering Task Force (IETF) لحساب المسارات ضمن الشبكات اعتماداً على شكل وهيكلية الشبكة مع تطبيق بعض القيود في عملية الحساب. في هذه المقالة تمت مناقشة أهم المعوقات والمشاكل الناتجة عن استخدام خدمات أنظمة الإرسال المتعدد (Multicast) ضمن الشبكات المتعددة النطاقات. حيث تم اقتراح ودراسة ثلاث خوارزميات توجية والتي يمكن استخدامها في وحدة حساب المسار من أجل بناء شجرة التوزيع للأنظمة المتعددة الإرسال ضمن الشبكات المتعددة النطاقات. هذه الخوارزميات هي: خوارزمية حساب أقصر مسار في كل نطاق على حده (SPED)، خوارزمية حساب أقصر مسار في جميع النطاقات 1 (SPAD1) وخوارزمية حساب أقصر مسار في جميع النطاقات 2 (SPAD2). حيث تم محاكاة هذه الخوارزميات لدراسة وتقييم جودة أداء الخوارزميات المدروسة عن طريق مقارنتها باستخدام بارامترات تقييم مختلفة مثل: متوسط استخدام عناصر الشبكة سواء أسلاك الوصل أو العقد (الموجهات) و جودة المسار المحسوب بين نقطتي المصدر– المستقبل. بناءً على النتائج التي تم الحصول عليها نستطيع القول إن الخوارزمية (SPED) تعدّ من أبسط الخوارزميات والتي لا تحتاج إلى زمن طويل للقيام بعملية حساب المسار مقارنة مع الخوارزميات الأخرى المدروسة. هذه البساطة في الخوارزمية تؤدي إلى عدم إمكانية الحصول دائماً على المسار الأمثل بين نقطتي المصدر – المستقبل ضمن شبكات المتعددة النطاقات.

الكلمات المفتاحية: عنصر حساب المسار، الشبكات المتعددة النطاقات، تقييم جودة الأداء، شجرة المسار الأقصر، نظام الإرسال المتعدد.

[*] مدرس – قسم هندسة الحاسبات والتحكم الآلي– كلية الهندسة الميكانيكية والكهربائية– جامعة تشرين–اللاذقية– سورية.

INTRODUCTION :

Since the multicast model for the Internet Protocol (IP) has been developed in 1988 [1], it becomes a very hot area of research, development and testing. Many applications such as multimedia video conferencing, distance education, remote collaboration, data replication and network games are nowadays based on the IP multicast. One of the primary advantages of the IP multicast is that, it delivers a single stream to many receivers .

Multicast Distribution Tree (MDT) is used by multicast-capable routers to build the paths which are used to deliver multicast traffic to all receivers. Therefore, building MDT that spans the multicast source and all destination nodes is a major task of the implementation of an efficient multicast routing protocol [9]. However, the utilization of the multicast technology is basically limited to the local area and small scale networks. In the context of large-scale (wide area networks), which contains multiple domains like Internet, the use of the IP multicast has met some challenges [4], such as: how can services be shared and distributed by different providers? Furthermore, the nodes being responsible for path computation have limited visibility of the inter-network topology when we consider the scalability and confidentiality constraints in a multi-domain environment.

The Path Computation Element (PCE) has been proposed by Internet Engineering Task Force (IETF) to solve this problem. PCE is designed to compute an optimal route between sources and destinations which based on a network graph and some constraints [8][10].

There are several methods proposed in the literature to perform calculation between two nodes using PCE, which means the case for unicast communication. However, these methods are not suitable for multicast communications. This is because the main aim of the unicast communication algorithms is to find the optimal point-to-point path. However, multicast communications aim to find the optimal multicast distribution tree.

The remaining of the paper is organized as follows. An overview of the multi-domain network topology is given in section II. In section III, the path computation element architecture is described. In section IV, three multicast routing algorithms used to find the shortest path tree are discussed. Results and comparisons between the investigated algorithms are shown in section V.

MULTI-DOMAIN NETWORK TOPOLOGY:

From [8] the classical definition of the domain is described as follows :

“A domain is any collection of network elements within a common sphere of address management or path computation responsibility. Examples of domains include IGP areas, Autonomous Systems (ASes), and multiple ASes within a service provider network”.

According to this definition and because of management and maintenance reasons, a large network may be divided into different domains and managed by separate administrative entities. For this structure, nodes within a domain are classified into two kinds: Border Node (BN) and internal node. While the internal node is only embedded within a single domain, the border node is placed at the boundary of a domain and used to connect together two or more domains. Fig. 1 shows the different possible patterns for the BN-BN linkage in a multi-domain (full mesh connection, 1:1 connection and partnered BN) [3]. In this paper the mesh pattern is used.

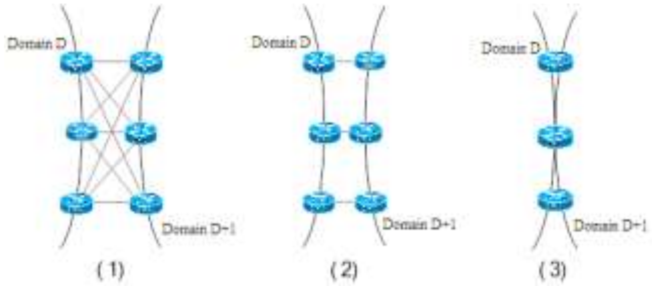


Fig. 1: Inter-domain BN-BN linkage patterns: (1) full mesh; (2) 1:1; (3) partnered BN

As a result of the above node classification, two kinds of network topologies can be defined: Low-Level-Topology (LLT) and Top-Level-Topology (TLT). The LLT shows how the nodes (internal nodes and border nodes) connected within a single domain. On the other hand, the TLT is composed of virtual nodes which represent the domains in the whole network. That is each domain is represented as a virtual node without considering the connectivity of internal nodes in the domain. In other words, TLT presents a virtual shape or structure of multi-domain topology.

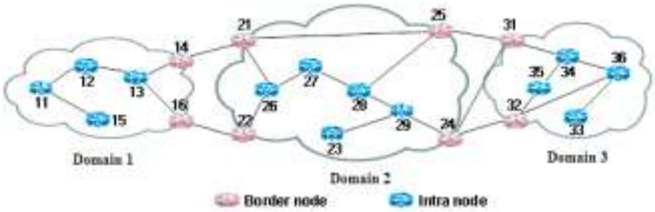


Fig. 2: An example of 3-domains-network topology

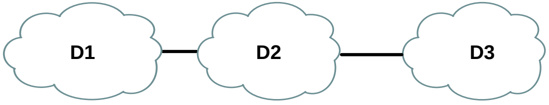


Fig. 3: TLT of Fig. 2 (Linear domain connectivity)

As seen in Fig. 2, the connections of nodes (n11, n12, n13, n14, n15 and n16) are shown as the LLT of domain D1. The network contains three domains: D1, D2 and D3 which form the TLT of the whole network. Note that only a virtual link between two joint domains is used in TLT without taking into account how many link connected between each other. Fig. 2 shows two links between D1 and D2: one link is between n14 and n21, and the other is between n16 and n22. However, only one connection is conceived between D1 and D2 in the TLT, as shown in Fig. 3.

Furthermore, the multi-domain topologies can be distinguished between linear domain connectivity and meshed domain connectivity. In the linear model, all domains connected in a linear order and each domain has maximal two “neighbor” domains. Fig. 2 depicts a simple example of linear domain connectivity. Domains D1, D2 and D3 connect in a linear order. In the mesh model, some of the domains are connected to more than two other domains in the network with a multipoint-to-multipoint link.

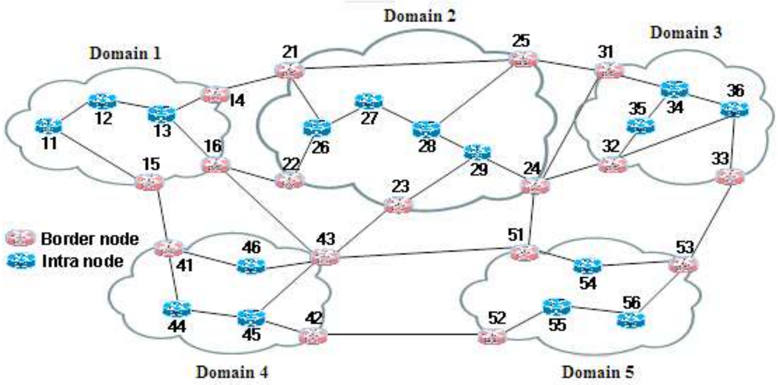


Fig. 4: An example of meshed domain connectivity

An example of meshed domain connectivity is presented in Fig. 4 and Fig. 5.

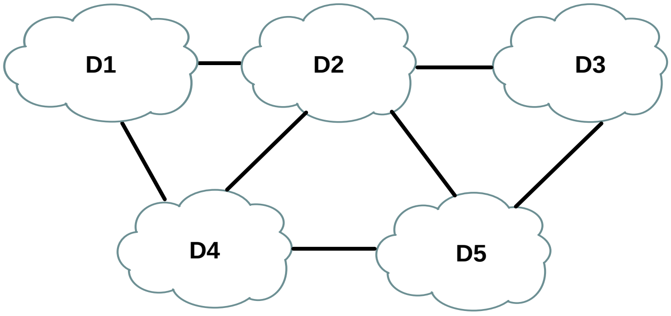


Fig. 5: TLT of Fig. 4

This paper focuses on the meshed domain connectivity scenario, where domain might connect with different “neighbour” domains. In such topologies, an internal node knows only the information of the local domain and the reachable information of BNs in the same domain and a BN has additional information about the connectivity between domains as well. BNs belong to one domain are distinguished between ingress node (IN) and egress node (EN). An ingress node is a BN which is connected with the upstream domain, related to this BN’s domain in the considered sequence of traversed domains. In the contract, the egress node is a BN that connected with the downstream domain.

Going back to the example in Fig. 2, the node n11 is the source (SRC) and its domain D1 is called the source domain. The node n36 is the receiver (RCV) or destination and its domain D3 is called the destination domain. Considering the sequence of traversed domains from D1 to D3, D1 is the upstream domain of D2 and D3 is the downstream domain of D2. In the domain D2, the nodes n21 and n22 are INs and the nodes n24 and n25 are ENs.

PATH COMPUTATION ELEMENT ARCHITECTURE :

The use of multi-domain architecture results in dealing with diversified customers that require different kinds of QoS (quality of service). Traffic engineering (TE) technique is used by the operators in order to provide such requirement, optimize the utilization of network resources and minimize traffic congestion. However, TE is traditionally used inside the domain without exchange with other operators. Consequently, without sharing TE information (including topology) for providing inter-domain services becomes a challenging task. To solve this problem, Internet Engineering Task Force (IETF) have been chartered the Path Computation Element (PCE) Working Group (WG) at the beginning of

2005. The PCE is proposed by this working group to compute the paths across the multi-domain topology. The author from [8] defines the path computation element as follows:

"A Path Computation Element is an entity (component, application, or network node) that is capable of computing a network path or route based on a network graph, and of applying computational constraints during the computation."

The most important reasons to use PCE are referred as:

- Performing path computation is taking constraints, such bandwidth, into consideration.
- Calculating paths between all domains, especially when the node responsible for path computation has limited information about the network topology between the pair source destination.
- Optimizing the calculation of primary and backup paths in multi-domain topology.

A. Terminology

- PCC: Path Computation Client, any client applies requesting a path computation to be performed by the Path Computation Element.
- PCE: Path Computation Element, an entity which performs path computation functions according to a path computation request.
- TED: Traffic Engineering Database. It contains the topology and resource information of the domain.
- PCReq: Path Computation Request, a request for path computation sent by PCC to PCE.
- PCRep: Path Computation Reply, a reply (response) for PCReq sent by PCE to the respective PCC that requested a path computation [8][10].

B. PCE Mechanism Overview

When a path, which satisfies a set of constraints such as QoS parameters, is desired, the source node plays the role as PCC. This PCC sends a path computation request to a PCE, which is designated to response to this PCC. After received a PCReq which contains source and destination addresses and constraints of the path (bandwidth requirement, cost limits, etc), the requested paths are computed by the designated PCE. Path computation requires knowledge of the available network resources like nodes, links and constraints such as connectivity, available bandwidth and link costs etc. This information is stored in the TED, which might be built from the information distributed by a routing protocol like OSPF-TE or IS-IS-TE. PCE uses path computation algorithms such as Constraint Shortest Path First (CSPF) to calculate a path. After finishing the computation, PCE sends a PCRep message as response back to the requested PCC. If the computation of the requested path is successes, a detail of the route will be included in the PCRep. The appropriate TE LSP can be set up by using a signaling protocol like Resource ReSerVation Protocol (RSVP) TE [6]. Otherwise, failure information will be sent back to the PCC.

C. Determination of the Sequence of Domains

The case of an arbitrary set of meshed domains is considered in this investigation. When source and destination are not in the same domain, domains of the path computation need to be decided previously. Domains {Ds…Dn-1, Dn, Dn+1… Dr} that constitute a domain chain between source and receiver represent SoD for PCEs. To avoid loop in meshed domain networks PCEs use SoD when they calculate the request paths. Domains Dn-1, Dn, Dn+1… represent the intermediate domains used in the path computation process.

Two methods are used to calculate the SoD in the proposed algorithms.

1) Shortest Path in TLT (SPTLT)

In the multi-domain topology, TLT indicates the connection structure of the domains. In the first case, the shortest path between the Dr and Ds in TLT is defined as SoD. When there are more than one shortest path exist, we take the path through the domain with lower ID. Taking Fig. 5 for example, we assume that D5 is Dr and D1 is Ds. There are two shortest paths between D5 and D1: {D5, D2, D1} and {D5, D4, D1}. We take the first one due to D2 has a lower number of domain ID. As a result, {D5, D2, D1} is the SoD from D5 to D1.

2) All Possible Paths in TLT (APTLT)

In this case, we take into account all possible paths between the Dr and Ds in TLT. In contrast to SPTLT which only one SoD is used, APTLT uses all SoDs in turn. Going back to the example discussed above, there are two SoDs from D5 to D1: {D5, D2, D1} (SoD1) and {D5, D4, D1} (SoD2).

MULTI-DOMAIN MULTICAST COMMUNICATION BASED ON PATH COMPUTATION ELEMENT:

There are several investigations proposed in the literature discussed the PCE in unicast case which result in computing point-to-point paths. These algorithms are not suitable for multicast communication. In contrast to unicast communication, the multicast communication uses a group address. Thus the source even does not need to know any information about the destinations. For this reason, the computation request is sent from the receiver. When a new receiver desires to join the multicast group, it sends PCReq message to the designated PCEr in the same domain.

How to implement an efficient multicast routing in multi-domain is the major goal of this paper. In the following sections algorithms to compute the MDTs are proposed which based on the shortest path tree principle.

A. Algorithms for Shortest Path Tree

Three algorithms are proposed in this investigation to perform the path computation process which in turn creates the following SPTs: Shortest Path in Each Domain (SPED), Shortest Path for All Domains 1 (SPAD1) and Shortest Path for All Domains 2 (SPAD2).

The first three steps are the same for all SP algorithms.

1. When a PCEr receives a PCReq message, it examines whether the source belongs to Dr ($\mathrm{SRC} \in □_r$), go to step 2. Otherwise, ($\mathrm{SRC} \notin □_r$), go to step 4
2. PCEr calculates the SP between SRC and RCV, then go to step 3.
3. PCEi sends a PCRep message with the calculated path to PCC. Then go to the END.

The remainder steps are not the same, they are introduced separately. However, before we begin to explain the difference between the proposed algorithms, we have to explain how multicast sources and receivers deliver their information.

1) Advertisement of Source Information

When a multicast source starts to send multicast data, it needs to inform all PCEs in the network that an active multicast source becomes active. In this case, multicast source acts as a PCC to send a so-called Source Notice (SN) message to its designated PCEs in Ds. This SN message has not been defined in PCECP. When the algorithms proposed in this paper are used, SN message need to be defined at first. This SN message contains the information about the source, for example, location, multicast group address.

1. PCEi (can be any PCE even PCEs or PCEr) receives an SN message.

2. PCEi examines whether it receives this SN message for the first time. If this is the case, go to step 3. Otherwise, go to step 5.

3. PCEi broadcasts (forwards) this SN message to its entire neighbor PCEs.

4. If at least one receiver has already registered at its PCEr for this MC group ($\mathrm{RCV} \notin \phi$), PCEr immediately performs a path computation process (detailed in the following sections). Otherwise, repeat step 2 until all PCEs in the network received the SN message. After all, the PCEs in the multi-domain topology succeed to perform a “Source Advertisement”. Go to the END.

5. PCEi drops SN message. Then go to the END.

2) Receiver Register

When a receiver wants to join a MC group, it sends a PCReq message to its designated PCEr in Dr. PCEr checks whether there is an active source for the requested multicast group. If there is an active source, PCEr immediately performs a path computation process. Otherwise, PCEr sends a “Path Computation Reply” message to PCC with empty path.

3) Shortest Path in Each Domain (SPED)

The idea of this algorithm comes from unicast path computation technique per-domain path computation (PDPC) proposed in [7]. Each leaf of the SPT is comprised of path segments calculated by PCE in different domains. When PCEs calculate these path segments, if considering the shortest path in each domain and not consider the optimal end-to-end path. The SoD is defined by using SPTLT method, presented by domain chain {Dr…Dn-1, Dn, Dn+1… Ds}.

The following steps present the computation process of this algorithm:

PCEr calculates the SP between ingress border node (IN) in Dr which connected the next domain in the SoD and RCV. If there are more than one BN, it means there are more than one shortest path, because from each IN to RCV has one shortest path. We choose the BN which has the minimum number of hops between BN and RCV from all shortest paths. This BN is the selected IN. If there are more than one path have the minimum number of hops, the lower ID of the BN will be selected, then go to step 5.

PCEr sends a PCReq message with all calculated path segments to its next PCE according to the SoD. Go to step 6.

PCEn examines whether ($\mathrm{SRC} \in □_n$), go to step 7. Otherwise, ($\mathrm{SRC} \notin □_n$), go to step 8.

If ($\mathrm{SRC} \in □_n$), PCEn (is also PCEs) calculate the SP between SRC and egress border node (EN) in Dn, which connected with the ingress node (IN) in Dn-1, then go to step 3.

If ($\mathrm{SRC} \notin □_n$), PCEn calculates the SP between the IN connected to Dn+1 and EN connected with Dn-1 according to the SoD in the direction to Ds, and then sends a PCReq message to PCEn+1.Then, go to step 6.

Let us take the network presented in Fig. 4 for an example. We assume that n11 is SRC. Therefore PCE1 (PCEs) broadcasts an SN message to each PCE in the network. RCV is n56 in D3. Therefore, PCE3 acts as a PCEr and receives a PCReq message from RCV (PCC). SRC and RCV are not in the same domain, thus, SoD is defined as {D3, D2, D1}. There is only one BN (n51) connected to the next domain D2. Hence, according to step 4, the SP from BN to RCV is {n51, n54, n53, n56}. In step 5, PCE3 sends a PCReq message with the calculated path to its next PCE (PCE2). PCE2 examines that ($\mathrm{SRC} \notin □_n$), thus, go to step 8. n21 and n22 are two border nodes connected with Dn-1 (D1). The path from n21 to n24 (connected with n32) only has 4 hops ({n21, n25, n28, n29, n24});

nevertheless, path from n22 to n24 has 5 hops ({n22, n26, n27, n28, n29, n24}). So, n21 is the internal node. PCE2 selects the SP from the internal n21 to egress n24 and adds it to the received path from PCE3. The path {n21, n25, n28, n29, n24, n32, n36} in a PCReq message is sent by PCE2 to PCE1. Again go to step 6, now, PCEn is PCE1. PCE1 examines that SRV is in the same domain, therefore, as detailed in step 7, PCE1 calculates the SP between SRC and EN (n14) in D1. Until now, the end-to-end path from SRC n11 to RCV n56 is {n11, n12, n13, n14, n21, n25, n28, n29, n24, n51, n54, n53, n56}, calculated domain-by-domain. Finally, the PCRep message contains this end-to-end path is sent by PCE1 back to PCE3 through PCE2. If there are other RCVs, uses the same way to get the end-to-end path to SRC n11. As a result, the SPT for this MC group is created (Fig. 6).

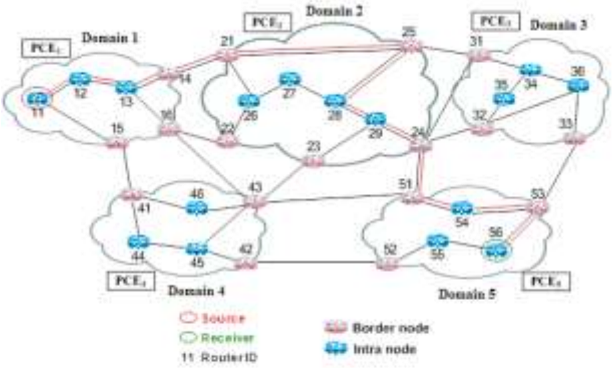


Fig. 6: An Example of SPED algorithm

4) Shortest Path for All Domains 1 (SPAD1)

Shortest Path for All Domains 1 (SPAD1) and Shortest Path for All Domains 2 (SPAD2) algorithms use the path computation procedure based on Backward Recursive Path Computation (BRPC) [4] to calculate each path from receiver to source. In other words, each branch of the SPT is calculated by using the unicast path computation technique. The difference is that in the unicast path computation case, the PCReq is sent by SRC at the beginning, but the calculation process begins from the receiver domain; in the multicast path computation case, the PCReq is sent by RCV, and the calculation process begins also from the receiver domain to the source domain. In SPAD1, each PCEr defines its own SoD by using SPTLT method, i.e., only one SoD is defined for each receiver.

The same example is used to explain SPAD1 algorithm. The first three steps and the SoD {D3, D2, D1} are the same as SPED detailed above. There is only one BN (n51) connected the next domain (D2); hence, the SP from BN to RCV {n51, n54, n53, n56} is the single branch of VSPTr. After PCE2 received the message from PCE5, it finds that ($\text{SRC} \notin \Box_2$), so, PCE2 calculates all potential shortest paths from INs (n21 and n22) to n24 ({n21, n25, n28, n29, n24} and {n22, n26, n21, n25, n28, n29, n24}). New VSPT2 is created and it is sent in a PCReq message to PCE1. As SRC is belong to D1, PCE1 calculates the SPs from SRC to the ENs (n14, n16) ({n11, n12, n13, n14} and {n11, n12, n13, n16}) and selects the shortest path between SRC and RCV. Compare to the blue path, the red one is shorter, thus, {n11, n12, n13, n14, n21, n25, n31, n34, and n36} is the final optimal end-to-end path from SRC to RCV. If there are other receivers, use the same calculation procedure simultaneously until all branches of SPT created (Fig. 7).

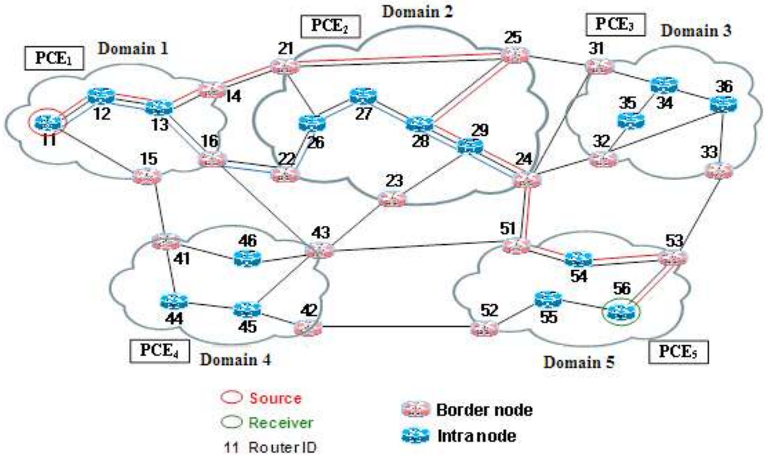


Fig. 7: An Example of SPAD1 algorithm

5) Shortest Path for all domains 2 (SPAD2)

Shortest Path for all domains 2 (SPAD2) algorithm uses the similar idea to SPAD1. The only difference is that in SPAD2 algorithm another method (APTLT) is used to define SoDs. Therefore, each receiver may have more SoDs presented by different domain chains: SoD1, SD2…SoDn, where n is the sum number of SoD for each receiver. Through each SoD, one shortest path is calculated. Finally, PCEs selects the shortest one from n paths. If there are more than one path have the minimum number of hops, choose the path which is in the minimum number of SoD.

Fig. 8 depicts the example used SPAD2. There are two SoDs: SoD1 ({D5, D2, D1}) and SD2 ({D5, D4, D1}). PCEs calculate the shortest path SP1 according to SoD1 (the blue path with 11 hops), SP2 according to SoD2 (the blue path with 8 hops). At the end, the best end-to-end path (the red one) can be selected from the two calculated paths.

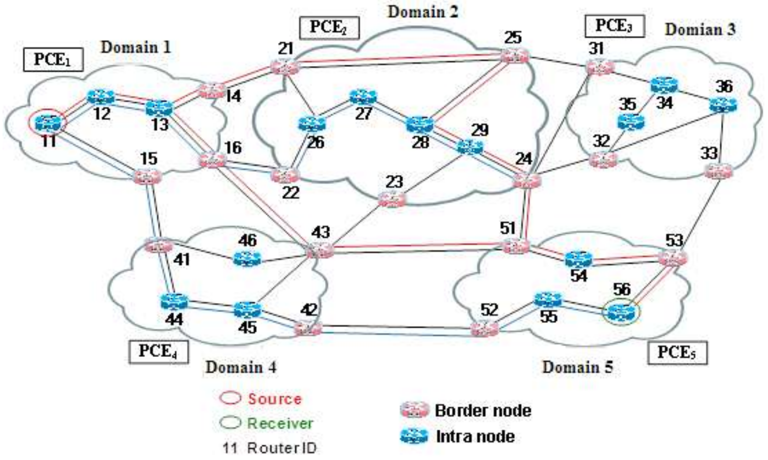


Fig. 8: An Example of SPAD2 algorithm

PERFORMANCE EVALUATION:

A. Calculation Scenarios

To evaluate and compare the investigated algorithms, they are implemented in Matlab [5]. Node Resource Usage (U_n), Link Resource Usage (U_l) and End-to-End Hop Count are used for comparing the performance of these algorithms. For this purpose, a random network topology is generated (T1) with 50 nodes and 100 links using a network generator (BRITE) [2]. Table -1 presents the main parameters of both investigated topologies.

Table -1: Main Parameters of two Investigated Topologies

Parameters	T_1
No. of Nodes (N_n)	50
No. of Links (N_l)	100
Maximum Degree	10
Minimum Degree	2
Mean Degree	4
Network Diameter	5

Three scenarios are defined to build the multi-domain topology in each calculation run:

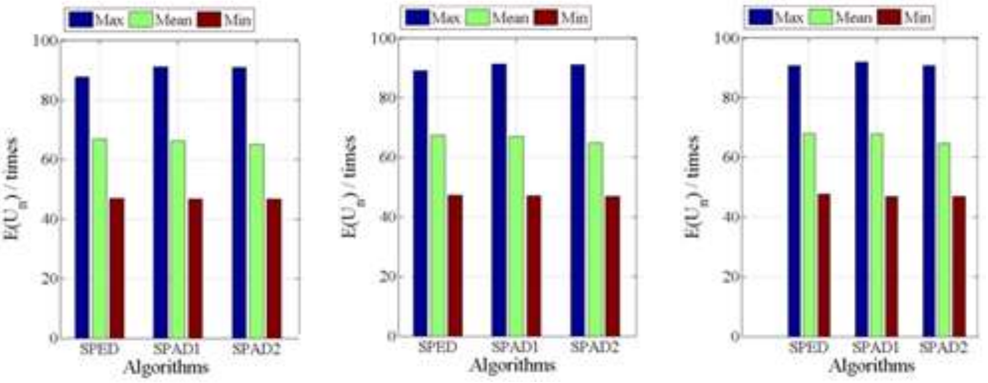
- Scenario 1: T1 is randomly subdivided into 4 domains.
- Scenario 2: T1 is randomly subdivided into 6 domains.
- Scenario 3: T1 is randomly subdivided into 8 domains.

In each Scenario, we run the calculation 100 times for statistical results. 100 different multicast groups are randomly generated in each calculation run. Each scenario uses all proposed algorithms to build the multicast distribution trees.

B. Results Discussion

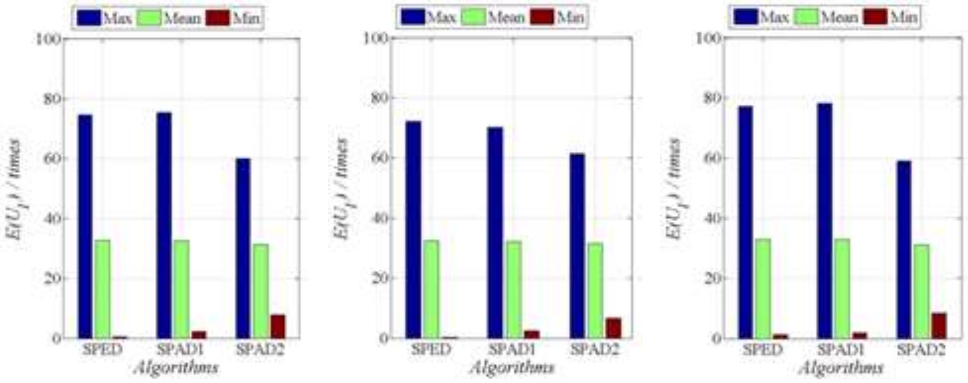
1) Average Node and Link Resource Usages

Fig. 9 and Fig. 10 compare the average, maximum and minimum values of U_n as well as U_l for these three scenarios. We can clearly see that the increase of the number of domains does not play a major role in the performance of the investigated algorithms. This is because the shortest path algorithm tries to distribute the multicast traffic generated by different sources throughout the network. Generally, a network has always more links than nodes which results in finding different paths between nodes. Because of that some links will rarely be used for distributing the multicast traffic. SPAD2 tries to find the shortest path in whole domains by choosing all possible paths, therefore it distribute the multicast traffic throughout the links better than the other algorithms.



(a) (b) (c)

Fig. 9: (a) Average Node Resource Usage for Scenario 1 (b) Average Node Resource Usage for Scenario 2 (c) Average Node Resource Usage for Scenario 3

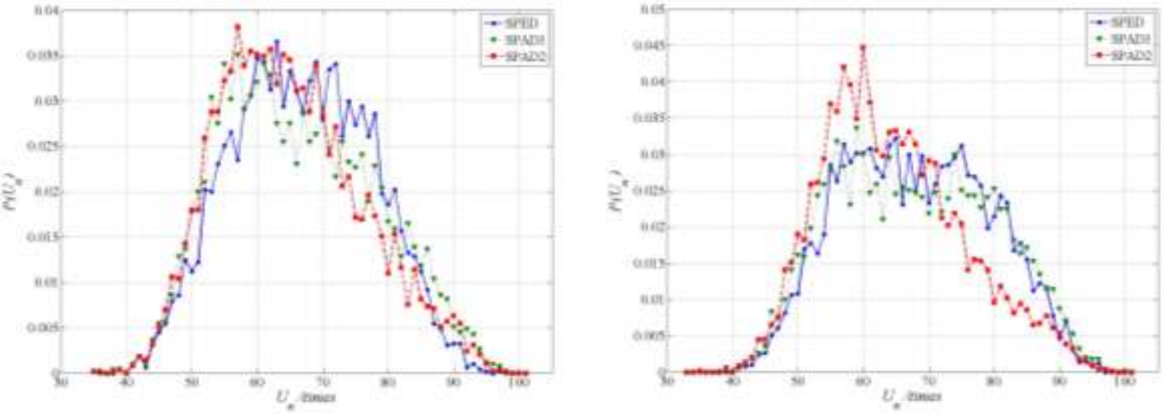


(a) (b) (c)

Fig. 10: (a) Average Link Resource Usage for Scenario 1 (b) Average Link Resource Usage for Scenario 2 (c) Average Link Resource Usage for Scenario 3

2) Probability Function of Node Resource Usages

Fig. 11 displays the probability function of node resource usage in the three scenarios. The results show that the multicast traffic will be distributed in varying degrees within the range [50, 90]. The two algorithms SPED and SPAD1 have roughly the same distribution. The SPAD2 algorithm differs from the other two algorithms in using more nodes with low load (range [50, 60]). This difference increases with increasing the number of domains. This is because in a low number of domains the probability that the multicast source and receiver are placed in the same domain is high compared to the case with large number of domains.

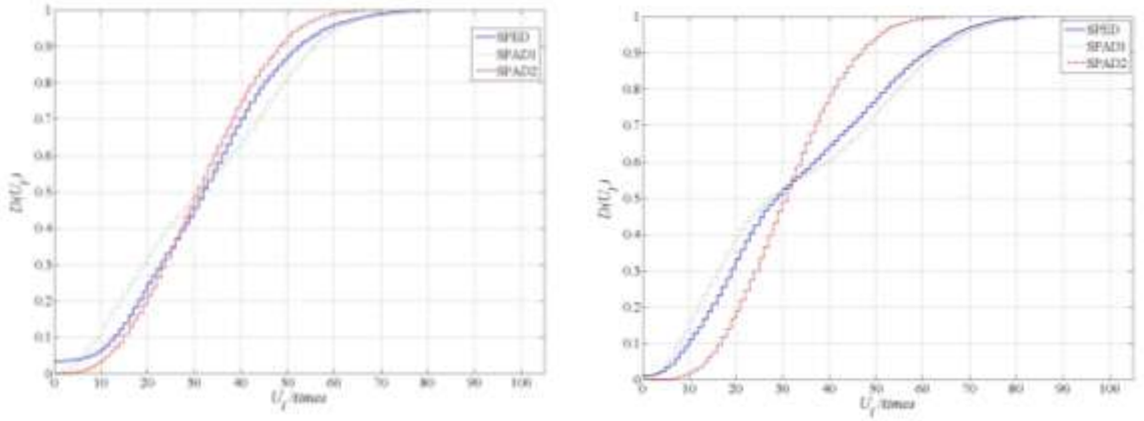


(a) (b)

Fig. 11: (a) Probability Function of Node Resource Usage for Scenario 1 (b) Probability Function of Node

3) Probability Function of Link Resource Usages

Fig. 12 illustrates the cumulative distribution for the investigated scenarios. There are some links in the SPED algorithm which never be used. This is because SPED uses the shortest path trees with root at each BN in each domain. This causes that some links may be not use in these multicast trees. Although both algorithms SPAD1 and SPAD2 use the same principle to find the end-to-end shortest paths, the SPAD2 algorithm performs better than the SPAD1. This is because the SPAD2 tries to find the end-to-end shortest path from all possible paths which results in finding the optimal end-to-end path. This results, in turn, in reducing the usage of links.



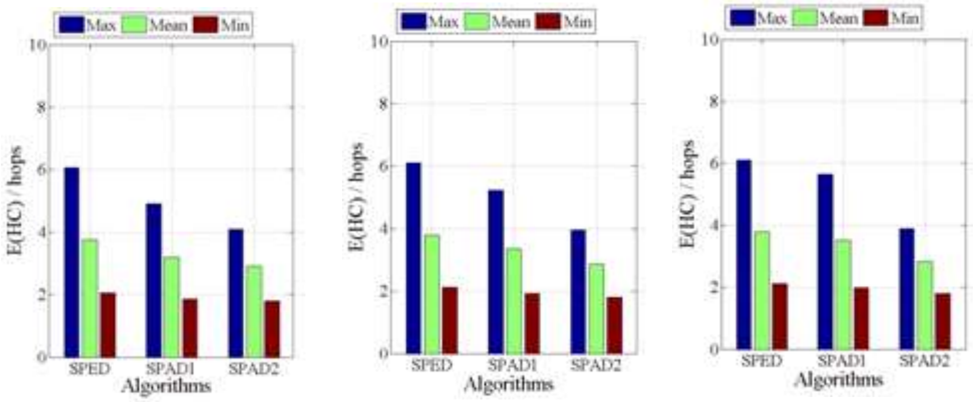
(a)

(b)

Fig. 12: (a) Cumulative Distribution of Link Resource Usage for Scenario 1 (b) Cumulative Distribution of Link Resource Usage for Scenario 3

4) End-to-End Hop Count

Fig. 13 presents the average end-to-end hop counts for each multicast group, as shown in Fig.13 End-to-end hop counts of SPT algorithms are unaffected by changing the number of domains. As mentioned above, these algorithms try to find the shortest path between multicast source and receivers. However, they distinguish from each other in the method that used to find the shortest path. The SPED algorithm finds the shortest path in each domain of the TLT. Then it collects these shortest path segments to obtain the end-to-end path. The SPAD1 algorithm finds the optimal end-to-end shortest path based on the best shortest path in TLT. The last algorithm SPAD2 finds the optimal end-to-end shortest path based on all possible paths in TLT. From the results we can clearly say that SPAD2 find the best end-to-end shortest path because of discussing all possible paths between multicast source and receivers.



(a)

(b)

(c)

Fig. 13: (a) Average End-to-End Hop Count for Scenario 1 (b) Average End-to-End Hop Count for Scenario 2 (c) Average End-to-End Hop Count for Scenario 3

CONCLUSIONS:

The scalability and confidentiality constraints are particularly issues for the optimal point-to-point path computation in a multi-domain environment. Some mechanisms have been investigated. One of the hottest areas is the use of PCE to compute the paths across multi-domain .

The major purpose of this paper is to use PCE to provide inter-domain multicast service. We propose three algorithms based on shortest path algorithm which can be used by PCE to build multicast distribution trees in a multi-domain topology. They are Shortest Path in Each Domain (SPED), Shortest Path for All Domains 1 (SPAD1) and Shortest Path for All Domains 2 (SPAD2).

The investigated algorithms are implemented in MATLAB. A random network topology with 50 nodes is used for the performance evaluation. In order to study the affection of the number of domains on these algorithms, the network topology is randomly subdivided into 4, 6 and 8 domains. We run each calculation scenario for 100 times. Each time, 100 multicast groups are randomly generated .

Through comparing different metrics, such as, the link resource usage, the node resource usage and the end-to-end hop count, we evaluate the performances of the proposed algorithms.

REFERENCE:

[1] S. Deering. "Host Extension for IP Multicasting". IETF (Aug 1989), [RFC 1112]

[2] BRITE, available at http://www.cs.bu.edu/brite/

[3] G. Bertrand, S. Lahoud, G. Texier and M. Moln'ar. "Computation of Multi-Constrained Paths in Multi-Domain MPLS-TE Networks". 978-1-4244-424 (c) 2009 IEEE

[4] J. P. Vasseur, R. Zhang, N. Bitar and JL. Le Roux. "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths". IETF (Apr. 2009), [RFC 5441]

[5] Documentation for MathWorks Products, R2008b, URL: http://www.mathworks.com/access/helpdesk/help/helpdesk.html

[6] J. Polk, S. Dhesikan, Cisco Systems. "A Resource Reservation Protocol (RSVP) Extension for the Reduction of Bandwidth of a Reservation Flow". IETF (May 2006), [RFC 2205]

[7] JP. Vasseur, A. Ayyangar, R. Zhang. „A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)“ IETF (Feb. 2008), [RFC 5152]

[8] A. Farrel, J. P. Vasseur, and J. Ash. "A Path Computation Element (PCE) - Based Architecture". IETF (Aug. 2006), [RFC 4655]

[9] Cisco Corporation. "IP Multicast Technology Overview" (April 2002)

[10] J. P. Vasseur and JL. L. Roux. "Path Computation Element (PCE) Communication Protocol (PCEP)". IETF (Mar. 2009), [RFC 5440]