Scalable Path Computation Flooding Approach for PCE-Based Multi-domain Networks

Jordi Perelló, Guillem Hernández-Sola, Fernando Agraz, Salvatore Spadaro, and Jaume Comellas

In this letter, we assess the scalability of a path computation flooding (PCF) approach to compute optimal end-to-end inter-domain paths in a path computation element-based multi-domain network. PCF yields a drastically reduced network blocking probability compared to a blind per-domain path computation but introduces significant network control overhead and path computation complexity. In view of this, we introduce and compare an alternative low overhead PCF (LoPCF) solution. From the obtained results, LoPCF leads to similar blocking probabilities to PCF while exhibiting around 50% path computation complexity and network control overhead reduction.

Keywords: Multi-domain, PCE, BRPC, domain sequence.

I. Introduction

As current transport network infrastructures grow, they are commonly segmented into multiple domains due to administrative, technological, reliability, or scalability reasons. In the framework of multi-protocol label switching (MPLS) and generalized MPLS networks [1], the Internet Engineering Task Force (IETF) has standardized the path computation element (PCE) to be responsible for computing both intra-domain and inter-domain paths. In a PCE-based multi-domain network [2], each domain can maintain one or even multiple PCEs, for example, for load sharing or enhanced resilience. These PCEs can compute intra-domain paths easily since they have full internal domain visibility. However, they usually lack the necessary visibility to compute whole end-to-end inter-domain paths.

In order to allow this computation, PCEs must be provided with the sequence of domains to be traversed from source to destination. On this sequence, a per-domain path computation can be performed. In this approach, the end-to-end inter-domain path is computed and provisioned on a per-domain basis so that every domain computes the path segment from its ingress node to the egress node connected to the following domain. Nonetheless, this concatenation of locally optimal-path segments likely leads to sub-optimal inter-domain paths.

Facing these limitations, the IETF proposes a backward recursive PCE-based computation (BRPC) procedure in [3] where the peering PCEs in the pre-defined domain sequence cooperate for computing an optimal end-to-end path. Besides this, by applying path-key techniques, BRPC can preserve the confidentiality across domains [3].

In standard BRPC, the responsible PCEs in every domain in the pre-defined domain sequence cooperate to generate a virtual shortest path tree (VSPT) with all the potential end-to-end paths crossing that sequence. Once the source domain PCE receives this VSPT, it selects the optimal candidate path. Therefore, the optimality of the computed path strongly depends on how this domain sequence is selected. In this context, the PCE standardization has not come up with any definitive solution to this goal, being still a work in progress within the IETF. As mentioned in [4], a path computation flooding (PCF) approach would allow BRPC to obtain optimal end-to-end inter-domain paths automatically without requiring any pre-defined domain sequence. However, due to its poor scalability, PCF is initially discarded since it is envisaged as unpractical for large multi-domain networks.

In this letter, we evaluate the performance of the PCF approach in terms of connection blocking probability (BP),

Manuscript received Feb. 25, 2010; revised Apr. 24, 2010; accepted May 13, 2010.

This work has been supported by the Spanish Science Ministry through the project ENGINE (TEC2008-02634).

Jordi Perelló (phone: +34 93 4054065, email: perello@ac.upc.edu), Guillem Hernández-Sola (email: guillem.hernandez@tsc.upc.edu), Fernando Agraz (email: agraz@tsc.upc.edu), Salvatore Spadaro (email: spadaro@tsc.upc.edu), and Jaume Comellas (email: comellas@tsc.upc.edu) are with the Advanced Broadband Communications Center (CCABA), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain.

doi:10.4218/etrij.10.0210.0063

control network overhead, and path computation complexity. As will be shown, PCF drastically reduces the connection BP of the per-domain approach but faces increased path computation complexity and control overhead. This motivates the proposal of a low overhead PCF (LoPCF) mechanism, whose performance is compared to PCF and BRPC solutions. Finally, in order to completely assess LoPCF, its feasibility on top of current PCE standardization is analyzed.

II. Proposed mechanisms

Working on the guidelines in [4], the next paragraphs introduce a PCF mechanism allowing BRPC to directly determine the optimal domain sequence in the path computation. To this end, we model the network as a graph G=(V, E), where V and E represent the sets of nodes and links, respectively. This global graph joins D sub-graphs, $G^i=(V^i, E^i)$, $1\le i\le D$, as D independent domains. In particular, $V^i=\{v_1^i,\cdots, v_{N_i}^i\}$ is the set of N_i intra-domain nodes in G^i, so that $V^i \cap V^j=\emptyset$, $i\neq j$, $1\le i, j\le D$, E^i is the set of intra-domain links, $E^i=\{e(v_w^i, v_k^i)\in E: v_w^i, v_k^i\in V^i, w\neq k\}$, and $\delta_{mn}^{ij}=\{e(v_m^i, v_n^j)\in E: v_m^i\in V^i, v_n^j\in V^j; 1\le m\le N_i, 1\le n\le N_j\}$ is the set of inter-domain links. Each domain G^i has a PCE^i responsible for the computation of paths inside it.

In our PCF mechanism, if a source node v_s^i needs an inter-domain path to a destination node v_d^j, it requests the end-to-end route to PCE^i using a *path computation request* (PCReq) message [5]. PCE^i then forwards this PCReq message directly to PCE^j (the destination domain PCE). Upon receiving it, PCE^j looks for every adjacent domain through which the source one can be reached, sending a *path computation reply* (PCRep) message [5] to the PCEs in these domains.

These PCRep messages contain the identifier of the current path computation request (Request ID) in addition to the computed VSPT to the specific neighboring domain (the path segments representing the best routes between the destination node and the border nodes connected to it). Each path segment in the VSPT is identified as (BN_j, v_d^j, C_j, K_j), where BN_j is the border node in G^j, C_j the cost metric associated to the path segment, and K_j is the path-key [3] stored in BN_j together with the computed intra-domain route from BN_j to v_d^j, which ensures the confidentiality between domains. Finally, although not considered in [4], we improve PCF operation with a domain list in the PCRep messages containing the identifier of the PCEs that have processed this PCRep. In this way, a loop-free flooding mechanism is obtained.

A PCE receiving a PCRep message updates the included VSPT, introduces its PCE ID to the domain list, and replicates the message to the PCEs of all potential upstream domains, except those appearing in the list. This is repeated at every transit domain. At the end, PCE^i will receive multiple PCRep messages with different VSPTs. These VSPTs will be used to create an N-ary path tree over which the optimal end-to-end route will be found and returned to v_s^i in a PCRep message.

This reverse flooding allows PCF to consider every possible VSPT from source to destination but at expenses of increased overhead and path computation complexity as demonstrated in section III. Aiming to improve the scalability of PCF, we also propose LoPCF as a low overhead PCF mechanism. In LoPCF, the source domain PCE (PCE^i) sends a direct PCReq message to the destination domain PCE (PCE^j). Then, PCE^j finds all possible upstream domains and sends a PCRep message to their PCEs. Such PCRep messages contain the Request ID and the VSPT to the specific upstream domain.

Algorithm 1. LoPCF: PCRep message processing at PCE^u.

```
If (PCRep→SourceNode∈G^u)
    Update (PCRep→VSPT);
    AddVSPTInformationToNaryTree();
Else If (ProcessedRequestID (PCRep→RequestId) == true)
    Discard (PCRep);
Else
    For (each upstream domain G^k)
        PCRepCopy = Copy (PCRep);
        Update (PCRepCopy→VSPT);
        Send (PCRepCopy, PCE^k);
    StoreProcessedRequestID (PCRep→RequestId);
    Discard (PCRep);
End;
```

When a neighboring PCE^u receives a PCRep message, it processes it as depicted in algorithm 1. Firstly, it checks if the source node is contained in its domain. If so, it updates the contained VSPT and includes this information in the N-ary tree for that Request ID. Otherwise, it checks if any PCRep for that Request ID was already processed. If this is true, the message is directly discarded. In contrast to PCF, LoPCF only allows a given domain to be included in the VSPT of the first PCRep message received, which may lead to sub-optimal inter-domain paths. Nonetheless, LoPCF allows the network to be sufficiently explored while introducing lower control overhead. If that Request ID was not previously processed, a PCRep message is forwarded to every upstream domain PCE with the VSPT accordingly updated. Finally, the PCRep message Request ID is stored to permit discarding any further PCRep for the same path computation request. This operation itself assures loop free VSPTs, making the domain list used in PCF not strictly necessary in LoPCF. Nonetheless, it demands the Request ID values to be globally unique and constant along the path computation, which is not a requisite in [3].

An important issue in both PCF and LoPCF is how the PCRep messages are collected at the source node to update the N-ary path tree. In this work, a timer is set to ensure that all the undiscarded PCRep messages are collected, which allows us to

quantify the scalability of PCF and LoPCF. Further work could address more complex strategies yielding accurate route selection and low path computation times.

III. Performance Evaluation and Discussion

The performance of PCF and LoPCF has been compared to standard BRPC by means of a self-developed C++ simulator describing the 9-domain transport network in Fig. 1, where each link carries 8 bidirectional wavelengths. In this network, intra-domain nodes are all-optical (without conversion), while border nodes employ optical-electrical-optical (OEO) conversion. Poisson connection requests are generated following a 70%/30% intra/inter-domain ratio. For inter-domain connections, source and destination domains are uniformly selected as well as the source/destination nodes in the respective domains. For intra-domain connections, source and destination nodes are randomly chosen in the same domain. Different loads are obtained by fixing the connection holding time to 200 s and varying the inter-arrival time accordingly. Connection requests demand a full wavelength. In the simulator, PCE and domain reachability are configured manually, although the latter could be alternatively provided by border gateway protocol. Note that PCE protocol (PCEP) messages are sent through the shortest route in terms of PCE hops.

Figure 2(a) depicts the connection BP achieved by standard BRPC, PCF, and LoPCF. In addition, the per-domain approach is also shown as a benchmark. In BRPC, the shortest domain sequence in terms of traversed domains is used. This leads to significantly increased BP compared to PCF and LoPCF. For instance, by fixing BP ≈ 0.5%, the offered load to the network can be doubled if PCF or LoPCF is implemented (from 60 to 130-150). As seen, PCF yields the best BP figures, closely followed by LoPCF.

Figure 2(b) compares the scalability of PCF and LoPCF in terms of the path computation complexity at source nodes. To this end, the size of the N-ary tree to compute each requested inter-domain path has been quantified during a 1,000 s time interval for 150 Erlangs (BP ≈ 0.5%). As seen, the high number of VSPTs gathered by PCF even lead to 30 N-ary tree branches in some cases with very far off source/destination domains (averaging 8.88 in the whole time interval). In contrast, this number is reduced by 53% when LoPCF is applied, averaging 4.15 in the considered time interval. Note that LoPCF also results in much reduced control overhead in terms of PCEP messages to serve a certain set of intra-domain and inter-domain connections. For instance, for 150 Erlangs, we measured that LoPCF reduces PCF control overhead by 51%, ultimately resulting in only 130% of the overhead in BRPC.

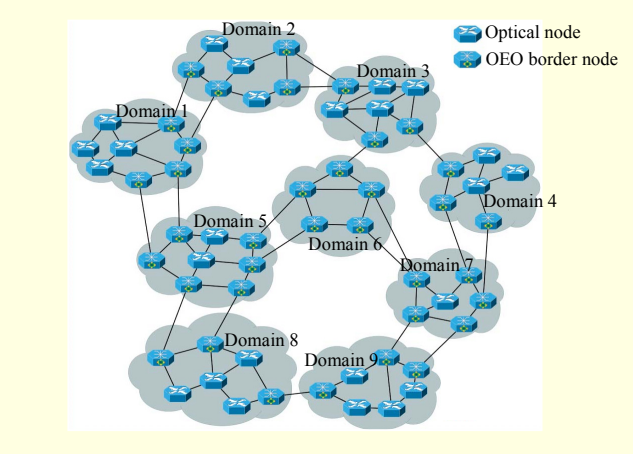

Fig. 1. Nine-domain network composed of 61 nodes and 95 links. 19 of 61 nodes are inter-domain nodes.

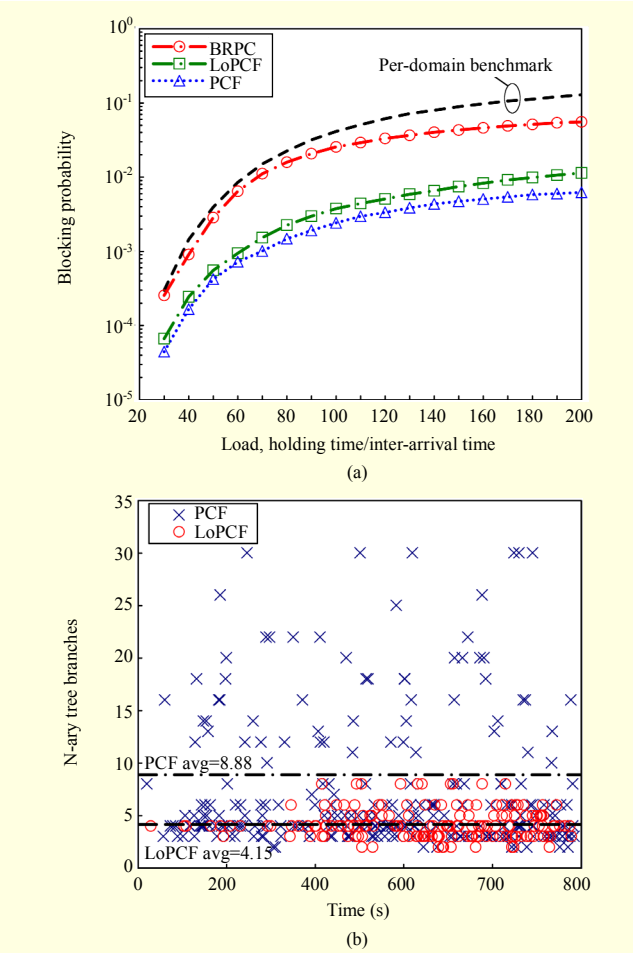

Fig. 2. Comparison of standard BRPC, PCF, and LoPCF: (a) network blocking probability and (b) path computation complexity represented as the number of branches in the N-ary tree.

The obtained low BP figures and scalable behavior allows us to propose LoPCF for PCE-based multi-domain networks.

Moreover, it would also fit the current PCE framework well. Looking at the PCEP standardization, no message format modification is required for LoPCF as the Request ID value can be carried in the *Request-id-number* sub-object [5]. In fact, the most notable differences to standard BRPC and PCEP are found in the LoPCF behavior, although they should not entail critical implementation issues: firstly, the PCReq is sent directly to the destination instead of hop-by-hop through the domain sequence; secondly, the backward flooding (already introduced in PCF [4]) breaks the initial PCEP client-server model as PCRep messages are not sent in response of a PCReq; finally, the Request ID values must be globally unique, which is not an initial requisite in [3].

References

[1] E. Mannie, "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," *IETF RFC 3945*, Oct. 2004.

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

[3] J.P. Vasseur et al., "A Backward-Recursive PCE-Based Computation Procedure (BRPC) to Compute Shortest Constrained Inter-domain Traffic Engineering LSPs," *IETF RFC 5441*, Apr. 2009.

[4] D. King and A. Farrel, "The Application of the PCE Architecture to the Determination of a Sequence of Domains in MPLS & GMPLS," *IETF draft draft-king-pce-hierarchy-fwk-03.txt*, Dec. 2009.

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