DACoRM: A Coordinated, Decentralized and Adaptive Network Resource Management Scheme

D.Tuncer, M.Charalambides, G.Pavlou
Department of Electronic & Electrical Engineering
University College London
London, UK

N.Wang
Centre for Communication Systems Research
University of Surrey
Surrey, UK

Abstract—**In order to meet the requirements of emerging demanding services, network resource management functionality that is decentralized, flexible and adaptive to traffic and network dynamics is of paramount importance. In this paper we describe the main mechanisms of DACoRM, a new intra-domain adaptive resource management approach for IP networks. Based on path diversity provided by multi-topology routing, our approach controls the distribution of traffic load in the network in an adaptive manner through periodical re-configurations that uses real-time monitoring information. The re-configuration actions performed are decided in a coordinated fashion between a set of source nodes that form an *in-network overlay*. We evaluate the overall performance of our approach using realistic network topologies. Results show that near-optimal network performance in terms of resource utilization can be achieved in scalable manner.**

Keywords- Adptative Resource Management, Online Traffic Engineering, Decentralized Network Configuration

I. INTRODUCTION

With the evolution of communication technologies and the emergence of new services and applications, developing approaches for the management of network resources with minimum human intervention has become a key challenge. Recent research efforts have been extending the autonomic computing principles [1] by applying them to network management systems. These efforts focus on enabling self-management capabilities, whereby network elements can adapt themselves to contextual changes without any external intervention. According to the autonomic management paradigm, the network is enhanced with self-awareness, self-adaptivity, and self-optimization functionality which is embedded within the network devices.

Today's practices for managing network resources rely mainly on off-line traffic engineering (TE) approaches where the expected demand is calculated from previous usage and a specific routing configuration is produced, aiming to balance the traffic and optimize resource usage for the next provisioning period. Given their static nature, these off-line approaches can be well sub-optimal in the face of changing or unpredicted traffic demand. Furthermore, despite recent proposals for adaptive TE [10][14][15], network resource management normally relies on centralized managers that periodically compute new configurations according to dynamic traffic behaviors. These centralized approaches have limitations especially in terms of scalability (i.e. communication overhead between the central manager and devices at run-time) and lag in the central manager reactions that may result in sub-optimal performance. To meet the requirements of emerging services, network resource management functionality that is decentralized, flexible, reactive and adaptive to traffic and network dynamics is necessary.

This paper describes the main features of DACoRM (Decentralized Adaptive Coordinated Resource Management), a new intra-domain resource management approach for IP networks, in which the traffic distribution is controlled in an adaptive and decentralized manner according to the network conditions. Based on path diversity provided by multi-topology routing (MTR), the traffic between any source-destination (S-D) pair is balanced across several paths according to splitting ratios, which are (re)-computed by the network source nodes themselves. New configurations are not computed by a centralized management entity, but instead, are the result of a real-time adaptation process executed by the source (i.e. ingress) nodes in the network. To decide upon the most appropriate course of action when performing periodic re-configurations, the source nodes coordinate among themselves through an *in-network overlay* (INO) where relevant information about new configurations is exchanged.

We describe in this paper the details of our approach, including an overall performance evaluation that demonstrates its benefits. More precisely, the paper presents the details of the adaptation process and elaborates on the specific algorithm for periodical re-configurations. It also explains the principles of the coordination between the source nodes. The paper further discusses different models to organize the source nodes in the INO and presents a signaling (i.e. in-network management) communication protocol to support interactions between entities in the INO. Results of the evaluation of our solution are encouraging. They indicate that near-optimal performance can be achieved in terms of resource utilization in a scalable and responsive manner.

The remainder of this paper is organized as follows. Section II introduces the background. Section III explains the principles of the coordination process between the different source nodes. Section IV describes the adaptation process by detailing the re-configuration algorithm. Section V presents the communication protocol and model. The performance of

978-1-4673-0269-2/12/$31.00 ©2012 IEEE

the approach is evaluated in section VI while in section VII, we review related work. We finally present a summary and insights for future work.

II. BACKGROUND

Current practices for managing resources in fixed networks rely on off-line approaches, where a centralized management system is responsible for computing routing configurations that optimize the network performance over long timescales, e.g. weekly or monthly. Given their static nature, these approaches can be sub-optimal in the face of unexpected traffic demand. To cope with their limitations, new TE schemes that can adapt to network and traffic dynamics are required.

In order to rapidly respond to traffic dynamics, online TE approaches dynamically adapt the settings in short timescales according to real-time information from the network [11]. There have been some proposals for both online MPLS-based TE, e.g. [12][13] and online IP-based TE, e.g. [10][14][15]. In all these approaches, the volume of traffic (represented by splitting ratio) assigned to several available paths between each S-D pair in the network is dynamically adjusted according to network conditions.

In our approach, the volume of traffic sent across different paths is also dynamically altered according to real-time information from the network. The adjustments are performed by the source nodes themselves which are organized in an INO, where the relevant entities can exchange information about the re-configuration actions to take. Overlay networks have received a lot of attention from the research community over the last decade, especially in the context of peer-to-peer networks [7][8]. An overlay network can be defined as a virtual network of nodes and logical links built on top of an existing physical network. In this paper we investigate different models to connect the nodes in the INO.

To provide a set of multiple routes between each S-D pair in the network, our approach relies on MTR [6] as the underlying network routing protocol. MTR extends the Open Shortest Path First (OSPF) and the Intermediate System to Intermediate System (IS-IS) routing protocols by enabling a virtualization of a single physical network topology into several independent virtual IP planes. The configuration of the different virtual planes is part of an off-line process which computes a set of desired IP virtual topologies given the physical network topology. The derived topologies are such that two objectives are satisfied: a) providing a set of non-completely overlapping paths between S-D pairs in the network, i.e. there is always at least one path which is not overlapping with the others, b) avoid introducing critical links, i.e. given a link *l* that is traversed by some traffic from node S to node D, there always exits an alternative path that can be used for routing the traffic without traversing *l*. The idea of obtaining topologies that satisfy these requirements is the following. Assume that *l* gets congested. We want to be able to move some traffic away from this link towards other parts of the network, i.e. towards other links. By computing topologies which satisfy the above requirements,

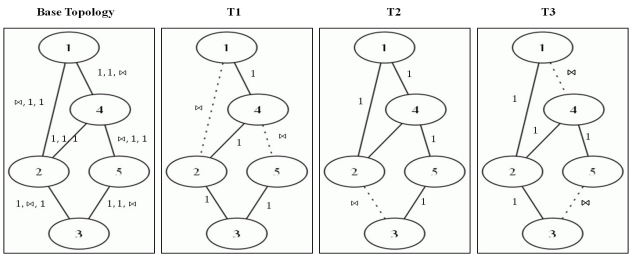


Figure 1. Building multiple topologies

we ensure that for any link *l* in the network, it is always possible to find at least one (S-D) traffic demand that is routed over link l in a set of topologies while it does not traverse *l* in the set of the other topologies.

Fig.1 illustrates a simple example of how virtual topologies that satisfy the aforementioned requirements can be derived from a base physical topology. We consider the S-D pair 1-3 where traffic at source node 1 is forwarded towards destination node 3. In each of the alternative topologies T1, T2 and T3, some links are assigned a MAXIMUM weight (that we represent here with infinity) which prevents these links from being used for routing the traffic demand between node 1 and node 3. With these settings, three non-overlapping paths can be determined between node 1 and node 3: (1;4;2;3), (1;4;5;3) and (1;2;3) and no critical link is created. The configuration of the alternative topologies is represented at the network level by associating a vector of link weights to each link in the network, each component of the vector being related to one topology. We can see in this simple example that only 3 virtual topologies are required to satisfy the objectives. Obtaining the desired virtual topologies in more complex and realistic topologies, where each S-D pair has to be taken into account, is not straightforward. Research work in [5][9][10] where MT principles are used for intra-domain off-line or online TE, shows that good path diversity can be achieved with only a small number of topologies e.g. three.

It should also be noted that balancing the traffic over the different paths provided by MTR may lead to route traffic over paths with longer round trip times. Since we are not targeting quality of service, this is not an issue in this work.

III. COORDINATED ADAPTIVE RESOURCE MANAGEMENT

A. *Overview and Main Features*

DACoRM allows for the traffic between any S-D pair of nodes to be balanced across several paths according to splitting ratios, which are (re-)computed by the source nodes themselves in real-time based on run-time information about the network state. Network information is disseminated to source nodes thanks to TE capabilities of enhanced Interior Gateway Protocols, that can incorporate TE metrics into link state advertisement [19]. Splitting ratios are decided by source nodes only and are not modified by other nodes on the route. To provide a set of possible routes between any of the S-D pairs, DACoRM relies on MTR as described in section II. The

distribution of the traffic load is controlled in an adaptive and decentralized manner through re-configuration actions that dynamically adjust the splitting ratios of some flows, so that the traffic is periodically re-balanced from the most utilized links towards less loaded parts of the network. Note that in this paper we refer to a traffic flow as the volume of traffic between source and destination nodes. The objective of this adaptive control is to permanently minimize the utilization of the most loaded link. Minimizing the maximum utilization in the network is a common objective considered by many load-balancing/TE schemes in the literature (e.g. [2][3][4][5]). In our approach, this objective is achieved through a combination of successive adjustments. More precisely, an adaptation process is periodically triggered. This consists of a sequence of re-configuration actions decided in a coordinated manner between the source nodes in the INO. The INO is formed by all network ingress routers that are equipped with the set of components to support the relevant functionalities. At each sequence/iteration of the adaptation process, the source nodes coordinate through the INO to select one of them that will compute new splitting ratios. The selected node is responsible for executing a re-configuration algorithm over its locally originating traffic flows, with the objective to move traffic away from the most utilized link in the network. It is worth mentioning that the INO is used solely for the signaling, i.e. in-network management, between source nodes for coordination purposes, but not for direct traffic routing/forwarding.

This adaptation process is performed in short-time scales, for instance, in the order of 5-10 minutes, which is in accordance with the common network monitoring interval [10][18].

B. Initiating/Executing a Re-configuration

To prevent inconsistencies between concurrent traffic splitting adjustments, only one source node is permitted to perform a splitting ratio adjustment at a time. The adaptation process is designed so that re-configuration actions are performed sequentially. At each iteration, one source node in the INO (called the Deciding Entity – DE) is selected to initiate re-configuration actions. The DE role can be taken by any node in the INO. To select a unique DE, each source node is therefore equipped with the necessary logic that enables it to determine independently whether or not it can assume the DE role for the new re-configuration interval. This logic relies on a selection rule that uses information about the link, l_{max}, with the maximum utilization in the network.

More precisely, the set of links in the network are statically and logically partitioned into a number of disjoint subsets N, where N is the number of nodes in the INO. The partitioning algorithm works as follows. Initially, each source node is associated with the set its outgoing links. The algorithm then considers one by one the other links in the network (i.e. core links) to determine to which local set it needs to be assigned. A core link is associated to the source node that uses this link the most (in terms of number of paths) to route the local traffic. Due to space limitations, the details of the process of partitioning the set of links are not provided here. The subsets are then distributed among the different nodes in the INO, so that each subset is placed under the responsibility of only one source node, i.e. a potential deciding node. Since the different subsets are disjoint by design, any link in the network belongs to one and only one subset. The subsets are then used by the source nodes to determine whether or not to assume the role of the DE. Upon receiving condition information about the link l_{max}, each source node checks whether l_{max} falls within its associated subset. If so, the relevant source node assumes the DE role for the new re-configuration interval. As explained in section IV, the DE is then responsible for performing the re-configuration.

C. Delegation Process Overview and Principles

While the DE is initially selected to perform re-configuration actions, it may not always be able to determine by itself a configuration with which traffic can be shifted away from l_{max} such that the utilization of l_{max} can be reduced while no other link in the network obtain a new utilization higher than the original utilization of l_{max}. In such a case the DE needs to delegate the re-configuration task to other nodes in the INO.

Upon failure to determine an acceptable configuration, the DE sends a delegation request to some of its neighbors in the INO through the overlay infrastructure. When receiving such a request, neighboring nodes, called Selected Entities (SEs), execute the splitting ratio re-configuration algorithm independently. Their results are communicated back to the DE, which then selects the configuration to apply (among successful ones), and notifies the relevant SE to enforce their new splitting ratios. This selection can be random but it can also follow some selection rules. To limit the number of messages exchanged and the response time, a delegation process can only be initiated by a DE.

Choosing the neighbors to which a delegation request is sent can influence the responsiveness and performance of the algorithm. Sending a request to only a limited number of neighbors can minimize the number of messages exchanged, as well as computation/communication overhead, but can also decrease the probability of discovering a node that can perform a successful re-configuration for further improvement of network performance. This trade-off can be parameterized by varying the number of SEs in the delegation process.

IV. RE-CONFIGURATION ALGORITHM

A. Objective

The overall objective of DACoRM is to balance the load in the network by moving some traffic away from highly utilized links towards less utilized ones in order to reduce the utilization of the hot spots against dynamic traffic behaviors. To achieve this objective, the proposed adaptive resource management scheme successively adjusts the splitting ratios of traffic flows through a sequence of re-configuration actions that constitute the adaptation process.

Each of these re-configuration actions is the result of the execution of a re-configuration algorithm. In fact, at each iteration of this process, the selected DE executes a re-configuration algorithm based on information from the

network concerning the link with the maximum utilization, l_{max}, and the set of other heavily utilized links, S_{HU}. The latter is defined as the set of links in the network with a utilization within α% of the utilization of l_{max}. Based on this information, the re-configuration algorithm tries to modify the splitting ratios of the traffic flows originated from DE, which contribute to the load on l_{max} such that: a) some traffic is moved away from l_{max}, and, b) the diverted traffic is not directed towards links in the set S_{HU} which are potentially vulnerable. A situation that should be avoided is that excessive traffic demands are diverted to a link which is originally not in S_{HU}, so that its new utilization becomes higher than that in S_{HU}.

The adaptation process terminates if a successful configuration cannot be determined or if it reaches the maximum number of permitted iterations (a parameter of the algorithm).

B. Principle/Algorithm

The algorithm consists of three phases.

1) Phase One

In this phase the algorithm determines if a re-configuration can be performed on one of the locally-originated traffic flows. The outcome of the first phase is either positive, which means that part of a local flow can be diverted from l_{max}, or negative if this is not possible.

The algorithm first identifies the local flows *f(S-D)* that can be diverted from l_{max}. A flow qualifies if: a) it is routed over l_{max} in at least one topology, and, b) it is not routed over l_{max} in all topologies, i.e. there exists at least one alternative topology in which the traffic is not routed over *lmax*.
For each *f(S-D)* that satisfies the two conditions, the algorithm defines two sets: S_{lmax}^{S-D} the set of routing topologies that use l_{max} to route *f(S-D)*, and $\overline{S}_{lmax}^{S-D}$ the set of routing topologies that do not use l_{max} to route *f(S-D)*. The set $\overline{S}_{lmax}^{S-D}$ is then itself partitioned into two subsets: the set of topologies in $\overline{S}_{lmax}^{S-D}$ that can avoid using any link from S_{HU} and the set topologies in $\overline{S}_{lmax}^{S-D}$ that use at least one link from S_{HU}. Based on these characteristics, the algorithm then classifies each *f(S-D)* into two categories: Category I - set of flows for which there exists at least one topology in $\overline{S}_{lmax}^{S-D}$ that do not use any link in S_{HU} and Category II - set of flows for which all topologies in $\overline{S}_{lmax}^{S-D}$ are using at least one link in S_{HU}.

The algorithm then considers each of the flows in Category I at a time and tries to adjust the splitting ratios. These are adjusted such that the ratios related to the topologies in S_{lmax}^{S-D} are decreased while the ratios related to the topologies in $\overline{S}_{lmax}^{S-D}$ are increased. The actual algorithm for adjusting the splitting ratios of a flow is presented in section IV.C. The resulting configuration is then analyzed to decide whether it is acceptable or not. A new configuration is said to be acceptable if: a) the utilization of l_{max} is decreased, and, b) no link l in the network attains a utilization higher than the original value of l_{max}. If these conditions are satisfied, the new splitting ratios are accepted. The result of the algorithm is set to positive and the next iteration of the adaptation process (i.e. re-configuration action) is triggered. If none of the local flows can satisfy the requirements, the result of the first phase is set to negative and the algorithm enters the second phase.

2) Phase Two

In case of unsuccessful local adjustments, the DE triggers a delegation process by sending a request to its neighbors in the INO for further attempts at alternative locations. Each neighboring node in the INO is responsible for executing the first phase of the re-configuration algorithm on its local flows, the result of which is communicated back to the DE. The DE is then responsible for selecting one of the proposed new configurations among the positive results and for notifying the corresponding neighbor about the decision. The details of this process are described in section V.

3) Phase Three

If none of the neighbors is able to perform a re-configuration, i.e. all results are negative, the DE can resort to using links from the set S_{HU}. A traffic flow among the ones in Category II is randomly selected and its splitting ratios are adjusted. If no such flows can be identified, the result of the re-configuration algorithm is set to negative.

In order to be implemented, the re-configuration algorithm should be lightweight in terms of computational overhead (i.e. time-complexity) imposed at each source node. The time-complexity of the first and third phases of the algorithm is dominated by the number of locally-originated flows to consider, which depends on the size of the network. In the case of a PoP-level topology with *N* nodes, for instance, where there are traffic demands between any pair of nodes in the network, this is *O(N-1)*. The actual cost of the second phase is related to the communication overhead (see section VI). It has to select one solution among several ones and as such, its complexity depends on the complexity of the selection policies. Since these policies are lightweight in terms of computation (find the maximum or find the first positive solution), it requires negligible CPU computation. As we can see, the overall time-complexity of the algorithm is therefore very low.

C. Adjustment of the Splitting Ratios

The splitting ratios of a traffic flow are modified so that the ratios for the topologies in S_{lmax}^{S-D} are decreased by a factor δ^- and the ratios for the topologies in $\overline{S}_{lmax}^{S-D}$ are increased by a factor δ^+:

$$\forall\, T \in S_{lmax}^{S-D},\ x_{T,new}^{S-D} = x_{T,cur}^{S-D} - \delta^-. \quad (1)$$

$$\forall\, T \in \overline{S}_{lmax}^{S-D},\ x_{T,new}^{S-D} = x_{T,cur}^{S-D} + \delta^+. \quad (2)$$

where $x_{T,cur}^{S-D}$ and $x_{T,new}^{S-D}$ represent the current and newly computed ratios respectively. Parameters δ^- and δ^+ are functions of the volume of traffic shifted away from l_{max} and the number of topologies in each set.

One of the challenges addressed by our re-configuration algorithm is determining the volume of traffic that can be

diverted from l_{max} in each iteration, while at the same time preserving the network stability. If too much traffic is shifted, other links may become overloaded. This may cause oscillations as in the next iteration traffic will need to be removed from these links. The volume of traffic that can be diverted at each iteration is therefore constrained by an upper bound V_{max}. This is determined by the bottleneck capacity in the set $\overline{S}_{lmax}^{S-D}$, by the utilization of l_{max}, and by parameter α of the set S_{HU}. The actual volume of diverted traffic for a selected flow is defined as the total traffic volume from that flow on one topology in S_{lmax}^{S-D} divided by a factor 2^n, where n is an integer that varies between 1 and an upper bound K. The value of n is initially set to 1 and is iteratively incremented by 1 until the diverted traffic volume is less than the upper limit V_{max}. To avoid diverting very little traffic at each iteration, the value of K is also bounded. The volume of traffic shifted from l_{max} is equally distributed across the topologies in S_{lmax}^{S-D} and equally diverted towards the topologies in $\overline{S}_{lmax}^{S-D}$.

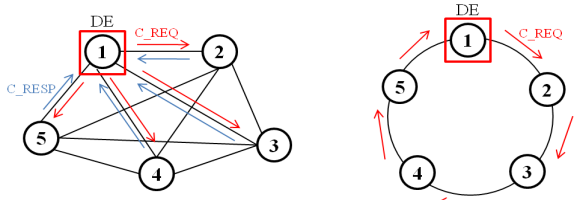
V. SIGNALING COMMUNICATION PROTOCOL

To support the adaptive re-configuration scheme, the source nodes are organized into the INO, where they can exchange information about re-configuration actions to take. The nodes especially interact through the INO in case of delegation where the DE communicates with neighbor nodes to determine a new configuration. In this paper, we consider two different models for the organization of the INO source nodes. In the first model, all source nodes are logically inter-connected forming a full-mesh topology. In the second model, source nodes are connected according to a ring topology, where each node is connected to only two other INO nodes. This section describes the characteristics of a protocol we have developed, which facilitates the communication between the INO nodes and supports the delegation process in each of the two models.

A. *Full-Mesh Model*

In this model, INO nodes are connected in a full-mesh topology, as shown in Fig. 2, where every node can logically communicate with every other node.

To support the delegation process, the developed communication protocol consists of three stages. Upon triggering a delegation process, the DE sends a delegation request - in the form of a *COMPUTE_REQUEST* (*C_REQ*) message - to each of its neighboring nodes (the SEs). The DE then enters a listening period where it waits for replies from all the SEs. Upon receiving a *C_REQ* message, the SEs execute the first phase of the re-configuration algorithm, as explained in section IV, and copy the result into a *COMPUTE_RESPONSE (C_RESP)* message that is sent back to the DE. In addition to compulsory information (such as the success status of any local re-configuration action), the *C_RESP* message can also include optional information that the DE can use when selecting a solution. This information can be for instance the contribution in terms of volume of traffic of the local flow for which ratio adjustments are proposed to the load of l_{max}. Once the listening period expires, the DE considers all the different *C_RESP* messages and selects among the successful



(a) Full-Mesh Model (b) Ring Model

Figure 2. Models to organize source nodes in the INO

TABLE I. STRUCTURE OF A MESSAGE IN THE FULL-MESH MODEL

Field	Description
MESSAGE HEADER	
Length	Length of the packet
Action	COMPUTE / APPLY
Type	REQUEST / RESPONSE
Status	SUCCEED / FAIL
Fill bits	Unused bits
OPTIONAL INFORMATION ELEMENT	
ID	Type of appended information
Value	Value of the appended information

configurations the one to apply. It then notifies the corresponding SE about its choice by sending a *APPLY_REQUEST (A_REQ)* message. Upon receiving this message, the chosen SE is responsible for enforcing the re-configuration it had proposed. Depending on the transport protocol used, the chosen SE may acknowledge the message by sending an *APPLY_RESPONSE* (*A_RESP*) back to the DE. In the rest of this paper, we consider that TCP is used as the transport protocol in order to provide the necessary reliability of the communication protocol.

TABLE.I presents the structure of the messages used in the full-mesh model. Each message consists of a message header and can be extended with optional information elements (IE). Only one IE is typically appended to the messages. According to the action it supports, the message falls into two categories - *COMPUTE* (driving the execution of the re-configuration algorithm at SEs) or *APPLY* (driving the choice of the re-configuration decisions to enforce) - that is indicated in the field *Action*. The type of the message (*REQUEST* or *RESPONSE*) is indicated in the field *Type*. It is important to note that *REQUEST* messages can only be sent by the DE. The result of the re-configuration algorithm is indicated in the field *Status*; the default value is *FAIL* and is updated by the SEs.

B. *Ring Topology Model*

In this model, INO nodes are connected according to a ring topology, as shown in Fig. 2, where each node is connected to only two other nodes. Communication is unidirectional, which means that a node can only pass information to its immediate neighbor in the ring. To communicate with any other nodes, a message needs to be sent over the ring until it reaches its destination. Unlike the mesh model, the set of neighbors of any node is limited to its direct next hop node.

The delegation process in the ring model is supported by a two-stage communication protocol as follows. Upon triggering a delegation process, the DE sends a delegation request to only one of its neighboring node (the direction followed in the ring must be fixed but can be either anticlockwise or clockwise). As in the full-mesh model, the request comes in the form of a *C_REQ* message. The DE then enters a listening period where it waits for the message to travel hop by hop through the ring until it reaches the DE again. Upon receiving the request message, the next hop node analyzes the content of the message to decide whether or not to replace the current re-configuration result with its own result. This is if the contribution in terms of volume of traffic of the corresponding local flow to the load of l_{max} is higher than the one related to the re-configuration currently reported. In that case, the node replaces the current information with the new one and forwards the message to the next hop node. Once the message reaches the DE it is analyzed, and, if a successful re-configuration is reported, the DE sends a *A_REQ* message to the address of the corresponding SE. While this message can be propagated through the ring, it can also be sent directly to the SE in the same manner as in some peer-to-peer file sharing systems where a direct connection is established between peers once the content has been located. Upon receiving the *A_REQ* message, the SE is responsible for enforcing the re-configuration it had proposed. Compared to the full-mesh model, where the final selection of a re-configuration action is left to the DE, each node in this model is responsible for determining whether the local solution is more appropriate than the one currently reported. The DE is not responsible for applying any selection rule. The structure of the messages used in the ring model is similar to the one used in the full-mesh model (see TABLE.I).

It can be inferred that the waiting time for the DE to obtain the best re-configuration proposal is relatively long, as the message needs to traverse all the nodes attached to the INO. In addition, due to the nature of the model, the actual waiting time increases with the number of nodes. For the delay not to be an issue in practice, the time required to perform re-configurations needs to be kept small (maximum few seconds) compared to the frequency at which adaptation is invoked (order of tens of minutes). Section VI investigates how the ring model behaves with the regards to the total re-configuration delay.

VI. EXPERIMENTAL RESULTS

In order to determine the overall efficiency of the proposed scheme, we have evaluated the performance of the different mechanisms used in DACoRM. We first quantify the gain that our adaptive scheme can achieve in terms of resource utilization. We then analyze the behavior of our approach according to the communication protocol described in the previous section.

A. *Performance of the DACoRM Adaptive Scheme*

We have evaluated the gain that our adaptive resource management scheme can achieve in terms of resource utilization using two real PoP-level topologies, namely the GEANT network [21] and the Abilene network [20], for which real traffic measurements datasets are available.

TABLE II. EXPERIMENT PARAMETERS

	GEANT	**Abilene**
Number of PoP	23	12
Number of unidirectional links	74	30
Number of topologies	5	4
Number of traffic matrices	672	
Frequency of adaptation	Every 15 min	
α	10 %	
Max number of iterations	50	

To quantify this gain we analyze the deviation of the maximum utilization in the network (max-u) from the optimum in different schemes:

- **Original scheme**: the original link weight settings are used in the original topology and no adaptation is performed.
- **DACoRM scheme**: virtual topologies are used to provide path diversity and periodic adaptation of the splitting ratios is performed.
- **The optimum:** we use the TOTEM toolbox to compute the optimal maximum utilization for each traffic matrix.

The incentives for this methodology rely on the fact that existing online TE approaches can achieve close to optimum performance (e.g. [13][14][10]). As such, instead of choosing an existing algorithm to compare against, we believe that directly comparing to the optimum provides the most relevant evaluation factor.

The settings of the different parameters used to perform the experiments are summarized in TABLE.II. The virtual topologies are computed according to the requirements described in section II. In order to represent a wide range of traffic conditions we consider traffic matrices over a period of 7 days. Although measurements for the Abilene network are available at shorter timescales (5 minute intervals) than the ones for GEANT (15 minute intervals), adaptation is performed at a frequency of 15 minutes in both topologies for consistency.

The average deviation of max-u from the optimum over a period of one week for the Original scheme and DACoRM is presented in TABLE III. The results show that near-optimal performances can be achieved by DACoRM in both the GEANT and the Abilene networks, with an average deviation of less than 10% from the optimal and for 98% and 96% of the traffic matrices considered respectively. DACoRM outperforms the Original scheme, with a gain of more than 100%. To observe the dynamics of the traffic traces used for the experiments in the GEANT network, the evolution of max-u at 15 minute intervals for a) DACoRM and b) the Original scheme is presented in Fig.3. As we can see, DACoRM can achieve a significant gain in terms of resource utilization in the GEANT network. The max-u obtained in our scheme is permanently much lower than the max-u obtained in the Original scheme.

TABLE III. DEVIATION OF THE MAXIMUM UTILIZATION FROM THE OPTIMAL

	GEANT	**Abilene**
Original scheme	89.88%	54.34%
DACoRM	9.07%	7.53%

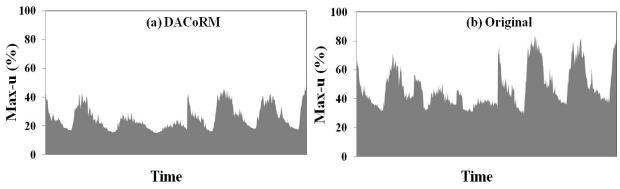


Figure 3. Evolution of max-u at 15 minute intervals using (a) DACoRM, and (b) Original scheme, for the GEANT network

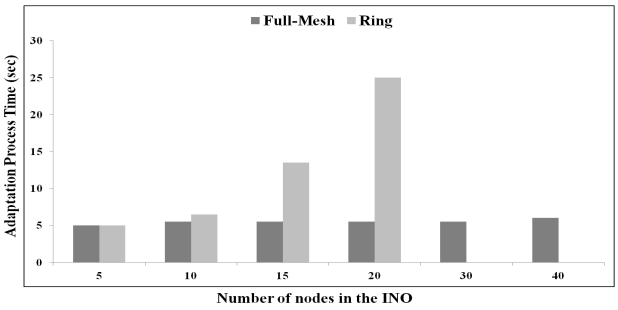


Figure 4. Evolution of the total execution time

Due to space limitation we only present the evolution for this network. Similar results are shown in the Abilene network.

B. Evaluation of the Communication Protocol

In addition to the performance in terms of resource utilization gain, the overall performance of DACoRM also relies on the convergence time and cost (in terms of management overhead) of the scheme. Different factors may influence the time required to complete the adaptation process, such as the physical characteristics of the network and the execution of the delegation process at different iterations of the adaptation cycle. The actual time to execute one iteration depends on the execution time of the re-configuration algorithm described in section IV. In particular, in the best case where no delegation is required, the execution time of the algorithm is given by its first phase. In this case, it takes only 7ms on average for a source node to determine new splitting ratios for the topologies considered. In case of delegation, however, the total execution time of the algorithm is driven by the second phase of the algorithm. Since this phase requires interaction between physically distant entities, its execution time may be significantly longer than the first phase (this involving only local actions). Several factors may affect the actual time requires for the second phase, such as the structure of the INO, the number of neighbors in the INO, the physical distance between INO nodes, but also, the characteristics of the communication protocol to support the interactions. In this section we analyze how the two models proposed in section V to organize the source nodes in the INO may affect the performance of DACoRM, both in terms of convergence time and in terms of overhead associated with coordination among the nodes.

In order to evaluate these factors, we consider a set of nodes that we connect according to the two models described previously, i.e. in full-mesh or in a ring. We perform several sets of experiments by varying the number of nodes in the INO and the connectivity model of the nodes. An experimental set involves the emulation of the adaptation process. A node in the INO is randomly selected to be the DE. The adaptation is run over 50 re-configuration iterations and at each iteration, the delegation process is triggered by the DE. This initiates a communication with its neighbors according to the communication protocol described in section V. The parameters used to perform the experiments are consistent with those considered in section VI.A.

It is also worth noting that although delegation may not be triggered at each iteration in a realistic scenario, our evaluation considers the worst case scenario. For each set of experiments we investigate the total time required to complete a cycle of the adaptation process ($T_{adaptation}$), i.e. to find and enforce new configurations, and we determine the volume of coordination messages required during the adaptation.

Fig. 4 shows the evolution of $T_{adaptation}$ according to the number of nodes in the INO for the two models. We can observe that the total time is not affected by the number of nodes in the full-mesh model, whereas this substantially grows as the number of nodes increases in the ring model. The results also show that the full-mesh model performs better than the ring model in terms of execution time. In fact, the ring model performs as well as the full-mesh for a small number of nodes (up to 10) but shows poor performance with a large number of nodes. Given the poor scalability performance achieved from only 20 nodes in this model, we do not extend the experiments to a larger number of nodes. Even if the actual time required for enabling communication between the different entities may be affected by the physical distance between source nodes, as reported in [17], the results show that the total time required for the adaptation can be kept to an insignificant level (few seconds) compared to the frequency at which the adaptive resource management scheme is invoked, i.e. every 15 minutes.

The evolution of the total number of coordination messages exchanged during the adaptation process is presented in Fig 5. As explained previously, we use the worst case scenario for our experiment where delegation is triggered

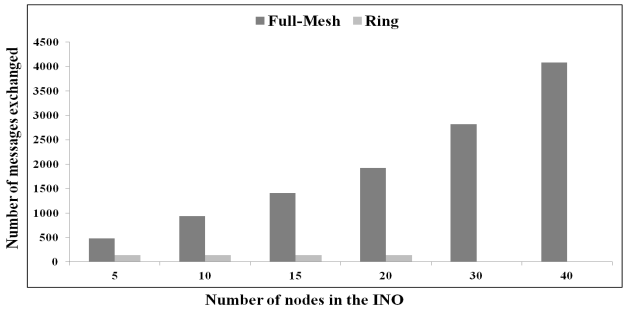


Figure 5. Evolution of the total number of coordination messages exchanged during the Adaptation Process

at each iteration of the adaptation process. We can observe that the actual gap between the number of exchanged messages in the two models increases significantly as the number of nodes in the INO increases. These results show that the ring model scales better than the full-mesh model in terms of communication overhead. As explained previously, we do not perform experiments with more than 20 nodes in the ring model given the poor scalability performance achieved in terms of delay. To analyze the scalability of the two models, we can theoretically compute the number of coordination messages required for each re-configuration interval in case of delegation. Assuming an INO of *N* source nodes, the actual number of messages exchanged in the full-mesh approach is the sum of *(N-1)* C_REQ, *(N-1)* C_RESP, 1 A_REQ and 1 A_RESP, i.e. a total of *2N* messages. In the ring model, the number of messages would be the sum of 1 C_REQ, 1 A_REQ and 1 A_RESP, i.e. a total of 3 messages. Although the number of messages is independent of the number of INO nodes in the ring model, it linearly increases with the number of INO nodes in the mesh approach. To minimize the number of signaling messages exchanged, compute requests can be sent only to a limited number of neighbors, but this is at the risk of decreasing the probability of discovering a node that can perform a successful re-configuration. Given the small size of coordination messages (typically less than 10 bytes), the overhead incurred by the delegation process is not significant given today's network capacities.

VII. RELATED WORK

Online TE approaches have been investigated both in the context of MPLS-based networks, e.g. [12][13], and IP-based networks, e.g. [10][14][15]. [12] and [13] propose to dynamically adjust the splitting ratios of network traffic flows over a set of pre-computed LSPs according to network conditions in order to optimize some objective functions. While the work in [12] aims at minimizing the sum of delays in the network, the authors in [13] are interested in minimizing the maximum utilization in the network. To support adaptation decisions taken at network edges, core nodes in [13] implement a control mechanism. Compared to these approaches, the authors in [14] propose a distributed solution where all nodes in the network are allowed to take adaptation decisions. These are responsible for dynamically splitting the traffic between different available next hops, based on real-time information received from upstream nodes. The main issue of this distributed approach is that a significant signaling overhead may be incurred since all nodes need to communicate to exchange information about the current state of the network. In [10], the authors propose a centralized adaptive TE approach that relies on two components: an off-line link weights computation algorithm to configure different virtual topologies in order to support path diversity, and an online adaptation algorithm to dynamically adjust the splitting ratios. Unlike previous approaches, the adjustments are not performed by the network nodes themselves but they are instead determined by a central manager that has a global knowledge of the network state. Although the consistency between re-configuration decisions is guaranteed due to the centralized nature of the approach, a significant communication overhead is incurred given that at each re-configuration period the central controller needs to gather information from all the links and nodes in the network. In addition lag in the central manager reactions may result in sub-optimal performance.

In DACoRM, new configurations are not computed by a centralized management entity that has a global view of the network. However, unlike the decentralized approaches described above, only source nodes are involved in the adaptation process. These coordinate among themselves through an INO to decide on the course of re-configuration actions to perform. Overlay networks have been widely used in the context of peer-to-peer systems [7][8], where research efforts have focused on developing scalable systems through optimized logical topologies and overlay routing protocols. Although an INO is used in DACoRM to support interactions between the source nodes, the purpose of this work is not to investigate features and techniques to support overlay systems.

To avoid flooding the network with signaling messages, the authors in [15] propose a scheme by which nodes use only local information from their direct outgoing links to decide whether or not to use them to route traffic. Due to the local scope of information regarding network conditions, this approach does not target optimality but robustness. However, the main drawback of approaches focusing on robustness is that they often have poor performance in terms of resource utilization in case of lightly loaded conditions in the network.

VIII. SUMMARY AND FUTURE WORK

This paper describes DACoRM, a new intra-domain resource management approach for IP networks, where traffic distribution is controlled in an adaptive and decentralized manner according to network conditions. Unlike off-line TE schemes, which rely on static configurations, DACoRM can efficiently deal with network and traffic dynamics by performing adaptations of routing configurations in short timescales. The analysis and experimental evaluation of the different mechanisms of DACoRM indicate that our approach can achieve near-optimal performance in terms of resource utilization in only few seconds and this, without overloading the network with excessive coordination messages. In future extensions of this work we plan to investigate other organizational models of the source nodes in the INO and also to investigate the influence of different factors on the overall performance of our approach such as the number of deciding entities in the network. Future work will also further evaluate the scalability of our approach using larger scale network topologies. We are finally interested in identifying generic patterns and infrastructure that can be used in different in-network self-management applications.

ACKNOWLEDGMENT

This work was partially supported by the European Union UniverSELF and COMET projects of the 7th Framework Program.

REFERENCES

[1] J.O. Kephart and D. M. Chess, "The vision of autonomic computing," IEEE Computer, vol. 36, no. 1, pp. 41-50, January 2003.

[2] B. Fortz et al., "Traffic engineering with traditional IP routing protocols," IEEE Commun. Mag., vol. 40, no. 10, pp. 118–24, October 2002.

[3] A. Sridharan, R. Guerin, C. Diot, "Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks," IEEE/ACM Transactions on Networking, vol. 13, no. 2, pp. 234-247, April 2005.

[4] Dahai Xu, Mung Chiang, and Jennifer Rexford, "Link-state routing with hop-by-hop forwarding can achieve optimal traffic engineering," IEEE/ACM Transactions on Networking, vol. 19, no. 6, pp. 1717-1730, December 2011.

[5] J. Wang, Y. Yang, L. Xiao, K. Nahrstedt, "Edge-based traffic engineering for OSPF networks," Comput. Netw., vol. 48, pp. 605-625, July 2005.

[6] P. Psenak et al., "Multi-Topology (MT) Routing in OSPF," IETF RFC 4915, June 2007.

[7] E.K. Lua, J.Crowcroft, M.Pias, R. Sharma, S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," IEEE Communications Surveys and Tutorials, vol. 7, no. 2, pp. 72-93, 2005.

[8] J. Risson, T. Moors, "Survey of research towards robust peer-to-peer networks: search methods," Comput. Netw., vol. 50, pp. 3485-3521, December 2006.

[9] A. Kvalbein, O. Lysne, "How can multi-topology routing be used for intradomain traffic engineering," in the Proc. of the 2007 SIGCOMM workshop on Internet network management, Japan, 2007.

[10] N. Wang, K.-H. Ho, and G. Pavlou, "Adaptive multi-topology IGP based traffic engineering with near-optimal network performance," in the Proc. of the 7th IFIP-TC6 Networking conference (Networking'08), Singapore, May 2008.

[11] Wang et al. , "An overview of routing optimization for Internet traffic engineering," IEEE Communications Surveys Tutorials, vol. 10, no. 1, pp. 36-56, 2008.

[12] A. Elwalid, C. Jin, S. Low, and I. Widjaja, "MATE: MPLS adaptive traffic engineering," in the Proc. of the 2001 IEEE INFOCOM Conference, 2001.

[13] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the tightrope: responsive yet stable traffic engineering," in the Proc. of the 2005 ACM SIGCOMM conference (SIGCOMM '05), USA, 2005.

[14] S. Fischer, N. Kammenhuber, and A. Feldmann, "Replex: dynamic traffic engineering based on wardrop routing policies," in the Proc. of the 2006 ACM CoNEXT conference (CoNEXT '06), Portugal, 2006.

[15] A. Kvalbein, C. Dovrolis and C. Muthu, "Multipath load-adaptive routing: putting the emphasis on robustness and simplicity," in the Proc. of the 17th IEEE International Conference on Network Protocols (ICNP 2009), October 2009.

[16] S.Sundaresan, C.Lumezanu, N.Feamster, P.Francois, "Autonomous traffic engineering with self-configuring topologies," Short Paper, in the Proc. of the 2010 ACM SIGCOMM conference (SIGCOMM '10), India, 2010.

[17] Y. Zhu, C. Dovrolis, M. Ammar, "Combing multihoming with overlay routing (or how to be a better ISP without owning a network)," in the Proc. of the 2007 IEEE INFOCOM, Anchorage, Alaska, 2007.

[18] S. Uhlig et al, "Providing public intradomain traffic matrices to the research community," ACM SIGCOMM Computer Communication Review, vol. 36(1), pp. 83-86, January 2006.

[19] D.Katz, K.Kompella, D.Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2," IETF RFC 3630, September 2003.

[20] The Abilene topology and traffic matrices dataset: http://www.cs.utexas.edu/~yzhang/research/AbileneTM/

[21] The GEANT topology : http://www.dante.net/server/show/nav.007009007 (2004)