A Distributed and Autonomic Virtual Network Mapping Framework

Ines HOUIDI, Wajdi LOUATI, Djamal ZEGHLACHE Groupe des Ecoles des Telecommunications Institut National des Telecommunications 9 rue Charles Fourier 91011 EVRY Cedex, France {ines.houidi, wajdi.louati, djamal.zeghlache}@int-edu.eu

Abstract

This paper addresses the challenge of assigning virtual networks, through network virtualisation, to the underlying physical network in a distributed and efficient manner. The paper reports on an implementation and evaluation of a distributed and autonomic framework with algorithms responsible for mapping virtual nodes and links to the physical resources. A Virtual Network (VN) Mapping Protocol is proposed for exchange of messages between autonomic substrate nodes. Results of the implementation and a performance evaluation of the distributed and autonomic VN mapping using a Multi-agent approach are provided.

1. Introduction

Network Virtualization has recently emerged as an important enabler for designing the Future Internet architecture [2, 3]. The Network Virtualization concept has been proposed as a promising way for diversifying the Future Internet architecture into separate logical networking environments called Virtual Networks (VN). Multiple network architectures, experiments and services can be simultaneously supported by the VNs over a shared substrate network [10, 11, 17].

As depicted in the figure 1, a Virtual Network consists of a group of *Virtual Nodes* (e.g. virtual routers) interconnected via dedicated *Virtual Links* over a shared substrate network. Multiple virtual networks may share the same underlying physical network. The virtual nodes and virtual links which constitute the VN should be mapped to a specific set of substrate nodes and substrate paths, respectively. A substrate path is a logical path between two substrate nodes which may be a single substrate link or a sequence of substrate links.

The creation and set up of a VN is achieved by selecting the best and optimal network topology from the physical substrate. Mapping multiple virtual networks into a



Figure 1. Mapping of Virtual Networks to a shared substrate network

shared physical infrastructure represents a significant challenge that has been addressed in many research studies. Finding the optimal VN mapping solution that achieves the best load balancing among the substrate resources and satisfies multiple constraints can be formulated as an NP-hard problem. Past research has proposed a number of heuristic, customized and/or greedy algorithms to ensure efficient mapping of virtual networks to specific set of nodes and links in the substrate network [8, 13, 14, 15, 16]. In these proposals, the VN mapping algorithm is carried out in a centralized manner. A central entity is responsible for receiving VN requests from users and for selecting and assigning a set of virtual nodes to a set of substrate nodes. However, the centralized approach may present serious limitations on scalability, efficiency and resiliency of VN provisioning and management. This becomes a challenge especially when the substrate network is a highly dynamic and changing environment (e.g. node/link failures, node mobility, etc).

To deal with the dynamic aspect of a substrate network, there is a real need to decentralize the substrate control and management to efficiently map VNs onto the substrate while reducing complexity and improving scalability. The objective of this work is to decentralize the decision-making for VN mapping among all substrate nodes. To achieve this purpose, the substrate nodes should be *autonomic* to decide which mapping, i.e. decisions/actions, to undertake automatically and without human intervention. This is exactly the motivation of this paper whose goal is to design a *Distributed* and *Autonomic* Mapping Framework responsible for self-organizing and self-managing the substrate nodes supporting the VNs. Consequently, the VN mapping algorithm is carried out in a distributed way among the autonomic substrate nodes.

Furthermore, the current proposals [8, 13, 14, 15, 16] do not address the problem of how to carry out their proposed VN mapping algorithms with respect to the design and implementation of the mapping framework, mechanisms and protocols used to exchange substrate messages, etc. This paper provides the design and implementation of the Distributed and Autonomic VN Mapping Framework based on the Multi-Agent based approach [9] to ensure distributed negotiation and synchronization between the substrate nodes. These nodes handle autonomous and intelligent agents which exchange messages and cooperate to carry out the distributed VN mapping algorithms. A Virtual Network Mapping Protocol is proposed to enable communications between the agent-based substrate nodes. Performance and scalability results of the distributed and autonomic VN mapping framework developed using the Multiagent based approach are reported.

Section II of this paper summarises related work. Section III presents the distributed VN mapping algorithm. Section IV describes the design of an autonomic and distributed mapping framework for the substrate. The implementation and evaluation of the proposed framework and algorithm based on multi-agent based approach is reported in Section V.

2. Related Work

VN mapping algorithms: A number of heuristic greedy algorithms [8, 15], customized algorithms [16] and iterative mapping process [14] have been put forward to efficiently assign VN to substrate resources. Y. Zhu and M. Ammar [15] proposed heuristic greedy algorithms to maintain low and balanced stress among all substrate nodes and links during VN assignment process. The aim of greedy algorithms is to assign virtual nodes to substrate nodes with maximum available resources. The overall objective is to achieve near optimal VN mapping solutions. In [14, 16], authors propose to use customized mapping algorithms to map special VN topologies (e.g. hub-and-spoke and backbone-star based VN topologies) in an optimal manner. J. Lu and J. Turner [14] proposed an iterative refinement procedure to map constraint-based virtual networks onto a common

physical substrate. Their objective is to find the least-cost virtual network topology that can handle any traffic pattern allowed by a set of traffic constraints.

Discussion and motivation: In these proposals, the VN mapping algorithms are carried out in a centralized manner. A central entity is responsible for selecting clusters from the VN topology (e.g. star cluster [15]) and for mapping the selected clusters to the substrate. The central entity should maintain up to date information about the substrate network (e.g. available resources) to make the appropriate VN mapping decisions (i.e. centralized decision making). However, maintaining up to date information about the substrate network in a centralized way suffers from scalability limitation, high latency and serious delays in making decisions especially when the underlying physical network is a highly dynamic and changing environment (e.g. node joining/leaving). Hence, the first objective of this work is to distribute the substrate control and management over the substrate nodes. These nodes should exchange messages, also called *knowledge*, and cooperate in a structured way to ensure decentralized decision-making for VN mapping. To achieve self-organization in distributed environments, the substrate nodes should control, manage and organize themselves to reach the VN mapping objectives. This paper provides the design requirements of autonomic substrate nodes to handle decentralized decision-making algorithms for VN mapping automatically and without human intervention.

3. Distributed Localized Virtual Network Mapping Algorithm

3.1. VN decomposition

Since VN topologies can become quite large, mapping the requested VN to the entire substrate at once is not feasible for latency and complexity reasons. One well known and viable solution, when conducted properly, is to subdivide the entire VN topology into a set of elementary clusters (e.g. path, tree and star clusters). This decomposition may reduce the complexity of mapping the entire VN topology since less virtual nodes and links are considered in each cluster mapping process. The VN decomposition provides also an efficient solution to deal with the dynamic aspect of a substrate. Indeed, when a change occurs in the substrate/virtual networks (e.g. node/link failures, congestion), it is more suitable to localize the concerned cluster and reassign it instead of reassigning the whole VN topology.

For instance, Y. Zhu and M. Ammar [15] proposed advanced algorithms to break up the VN topology into a number of connected sub-VN such as hub-and-spoke clusters. The hub-and-spoke (or star) topology is composed of a central node (i.e. *hub*) to which multiple adjacent nodes (i.e. *spokes*) are connected. In a hub-and-spoke cluster, some spoke nodes may act as the hubs of other clusters. This paper focuses on the star based VN decomposition and addresses how to decompose and map the star topologies in a *collaborative* and *decentralized* manner rather than centralized as in the case of [15].

3.2. Collaborative and Distributed Decision Making for VN Mapping

To deal with the dynamic aspect of a substrate, the idea of this work is to decentralize the decision-making process among all substrate nodes to decompose and map virtual networks onto the substrate. This is achieved by allowing substrate nodes to subdivide the VN into hub-andspokes (i.e. star) clusters and plan which mapping decisions/actions to undertake. The distributed mapping of huband-spokes clusters to the substrate is recursively executed as follows (Algorithm 1):

Algorithm 1 : Hub-and-Spokes Clusters Mapping

1. Find the substrate node with the maximum available resources (referred to as the *root* node)

2. The root node selects the Hub-and-Spokes Cluster from the VN topology (see Algorithm 2 for VN decomposition)

3. Assign the hub node to the root node

4. The root node determines the set of substrate nodes able to support the spoke nodes based on shortest path algorithm.

5. Remove the substrate nodes and paths already assigned to the hub-and-spoke cluster.

6. The root node selects the next root node: Go to (1)

Each substrate node designated as root will be responsible for selecting a hub-and-spokes cluster from the VN topology and assigning it to the substrate. The selection of the hub-and-spokes clusters from the entire VN topology is recursively performed as follows (Algorithm 2):

Algorithm 2 : Hub-and-Spokes Cluster Selection

1. Find the virtual node with the highest capacity to make it the hub of the cluster.

2. Determine the neighboring virtual nodes directly connected to the hub node to represent the spoke nodes.

3. Remove the hub and spoke nodes as well as their corresponding links from the VN topology

The mapping of hub-and-spoke clusters to the substrate is carried out in a distributed way among the substrate nodes. The distributed VN mapping algorithm is considered as a cooperative task executed jointly by all substrate nodes via messages exchange. Indeed, each substrate node can make local decisions individually (i.e. virtual resources allocation) but should also communicate, collaborate and interact with others to plan collective mapping decisions to efficiently assign VNs to the substrate. This enables *distributed localized mapping* of VNs. To self-organize and self-manage the interaction, cooperation and negotiation between all substrate nodes, a *Distributed and Autonomic Mapping Framework* is proposed. The objective is to ensure distributed decision making (for VN mapping) among the substrate nodes based on global knowledge/view of the entire substrate and virtual network topologies.

4. Distributed and Autonomic Mapping Framework

In this section, a Distributed and Autonomic Mapping Framework is defined to self-provision and self-manage VNs according to high-level goals and policies (representing business-level objectives). This framework is used to map, in a distributed way, multiple VN topologies to the substrate network on behalf of the substrate provider.

As shown in the figure 2, the proposed framework is composed of *Autonomic* Substrate Nodes (section 4.1) interconnected to each other using a *VN Mapping Protocol* (section 4.3). The autonomic substrate nodes should exchange control/signaling information needed to carry out the distributed VN mapping algorithm (section 4.4).



Figure 2. Autonomic and Distributed Management Framework Architecture

4.1. Autonomic Substrate Nodes

A substrate node designated as a root node is responsible for selecting the hub-and-spokes cluster from the VN topology and assigning it to the substrate. In a dynamic substrate environment, the root nodes should control and manage their virtual resources without human intervention. This section provides the design requirements of substrate nodes to allocate virtual resources to VNs in a dynamic and autonomic manner. The substrate nodes should be open, flexible, virtualized and *autonomic* to support dynamic VN mapping, provisioning and management. The *Autonomic Substrate Node* capabilities are [6]:

- Self-configuration : The Autonomic Substrate Node should be able to automatically configure and reconfigure itself and the supported cluster following unpredictable VN deployment and provisioning requests and external conditions. Since VN requests may arrive dynamically, the autonomic substrate node should re-configure itself at run time in accordance with arrival/departure of VN requests (e.g. virtual resources (re)allocation, etc).
- Self-optimisation : The Autonomic Substrate Node should seek ways to optimise itself to improve its execution and to ensure that the node is running at optimum levels. The autonomic substrate node should share its virtual resources between multiple virtual nodes in efficient manner. In fact, the substrate node should not allocate large substrate resource to assign virtual nodes which do not really require important resources (e.g. spoke nodes which do not support important traffic volume).
- Self-healing : To discover and repair localized problems and failures automatically in a substrate node (e.g. node and link failures), there is a definite need for an effective self-healing system. The self-healing system should monitor, analyze, diagnose and identify errors and failures without human intervention.
- Self-protection : The Autonomic Substrate Node should be capable of identifying, detecting and protecting its resources from malicious attacks and inadvertent cascading failures. The Autonomic Substrate Node should maintain overall system security and integrity.
- Self-awareness : The Autonomic Substrate Node needs detailed knowledge of its state (e.g. available capacity), components and behaviors and should be aware of its logical and physical resources that can be shared, isolated, allocated or borrowed. The Autonomic Substrate Node should monitor, maintain and adjust its operation, resources and components during VN provisioning.

As depicted in the figure 3, each Autonomic Substrate Node integrates an *Autonomic Manager* responsible for:

• Analyzing policies specified by the substrate provider. The policy rules maintain the VN topology as well as



Figure 3. Design of an autonomic substrate node

performance metrics associated to virtual nodes and links.

- Supervising and monitoring (M) physical and virtual resources within the substrate node. The monitoring entity generates metrics that express the node state, capabilities and performance (i.e. availability of resources). These metrics are stored in a knowledge (K) data base.
- Making decisions (*D*) in accordance with the local knowledge base. In each autonomic substrate node, the Decision entity is responsible for maintaining and executing the collaborative and distributed VN mapping algorithm introduced in the section 3.2.
- Executing actions (E) in the substrate nodes (i.e. managed elements). For instance, if the autonomic substrate node is selected as a root node, it should allocate virtual resources to support the hub node, on the one hand, and determine the set of substrate nodes able to support the spoke nodes based on shortest path algorithm, on the other hand.

These four stages described above are achieved under a closed-loop control.

4.2. Multi-Agent based Autonomic and Distributed Mapping Framework

Since the Multi-Agent System approach [9] represents a good tool to build modular and autonomous systems capable of operating in dynamic and distributed environments, we propose to implement the Autonomic and Distributed Mapping Framework based on autonomous and intelligent Agents [7]. The Multi-Agent based framework ensures distributed negotiation and synchronization between the autonomic substrate nodes. These nodes handle autonomous and intelligent agents which exchange messages and cooperate to perform the distributed VN mapping framework and algorithm.



Figure 4. Design of an autonomic substrate node using agent based approach

As shown in the figure 4, an Agent based Autonomic Substrate Node is composed of three main components: a Knowledge Base, a Rule Base, and an Engine. The Engine is the active component in the Agent based Autonomic Substrate Node responsible for executing three phases: Perception (i.e. Monitoring), Making Decisions and Action (i.e. Execution). The autonomic manager explored in section 4.1 is performed by the Engine. The Knowledge Base maintains the state of the substrate node (e.g. available capacity of virtualized resources) and its environment (the state of the links directly connected to the node as well as the state of adjacent nodes). The Rule Base maintains a set of possible actions that an agent based substrate node should execute whenever rule conditions are satisfied. The decision making process consists in selecting the appropriate action, based on the knowledge and rule base, to be executed and applied in the substrate node.

4.3. VN mapping protocol

A communication protocol, called *Virtual Network Mapping Protocol*, is defined in this section to exchange messages and information between all substrate nodes. The VN mapping protocol is based on five types of messages: *START, NOTIFY, MSG, NEXT* and *STOP*. These messages are sent and received asynchronously between substrate nodes to exchange information and organize the distributed algorithm iterations. The protocol messages are described as follows:

- **START** (*VN Request*): This message is sent from a synchronizer (e.g. substrate provider) to all substrate nodes to trigger and start the distributed mapping algorithm. The START message handles the VN request information (e.g. list of virtual nodes and links, capacities, etc).
- NOTIFY (n_v-id, n_s-id): After assigning a virtual node n_v to a substrate node n_s, the substrate node

should notify the other substrate nodes about this mapping. The n_v -*id* and n_s -*id* represent the unique identifiers for the virtual node n_v and the substrate node n_s , respectively.

- MSG (n_s-id, C(n_s)): This message is used to exchange node capacities among all substrate nodes. Each node broadcasts its capacity to all other nodes.
- NEXT: Once the hub-and-spoke cluster has been assigned, the root node supporting the hub should send a NEXT message to designate the successor root node to support the hub of a new cluster.
- **STOP**: This message is used to stop the execution of the distributed algorithm once the entire VN topology is mapped successfully onto the substrate.

4.4. Multi-Agent based Distributed VN Mapping Algorithm

Figure 5 depicts a step by step scenario describing the implementation of the distributed VN mapping algorithm (*Algorithm 1 and 2*) using the Multi-Agent framework.



Figure 5. Multi-Agent based Distributed VN Mapping Algorithm

In each substrate node, the VN mapping algorithm (i.e. Algorithm 2) is started upon receiving a START message handling the VN request from a synchronizer (e.g. substrate provider). The substrate nodes should communicate with each other to determine the root node which corresponds to the node with maximum available resources in the substrate. For this purpose, the substrate nodes exchange their node capacities via the MSG message. Based on this exchange of capacities, each node compares its capacity to that of the other nodes. As depicted in the figure 5, the node i has the highest capacity compared to the all other capacities (i.e. node i is the root).

Next, the root node should select the hub-and-spoke cluster in the VN topology to be assigned to the substrate (using algorithm 1). The hub node is determined by searching for the highest virtual node capacity.

Once the hub-and-spoke cluster is selected, the hub node is assigned to the root node in the substrate. Next, the root node determines the appropriate set of substrate nodes capable of supporting the spoke nodes efficiently. To achieve this goal, the root node determines the substrate nodes that have the shortest path to the root node and, at the same time, have the maximum node capacity. The spoke nodes with higher required capacity are mapped to the substrate nodes with higher available capacity.

Once the hub and spoke nodes are assigned to the selected substrate nodes, the root node removes these nodes from a node list and notifies all substrate nodes, via the NO-TIFY message, to inform them about this mapping. Upon receiving a NOTIFY message, the substrate nodes remove the virtual nodes and substrate nodes already assigned from their node lists and store the mapping information in their knowledge databases (k).

Once the first hub-and-spoke cluster is mapped, the node i designates the next root (or root successor) in the substrate (with the maximum available capacity) to support the next hub-and-spoke cluster of the VN topology. Upon receiving a NEXT message, the new root node (e.g. node k) repeats the same mapping procedure described above while taking into account the substrate nodes already assigned to previous clusters.

Once the entire VN topology is mapped successfully onto the substrate, a STOP message is sent by the last root node to stop the execution of the distributed algorithm.

5. Implementation and Performance Evaluation

The Autonomic and Distributed Mapping Framework has been implemented and tested over the GRID5000 platform [4]. This platform can emulate a real substrate network where different topologies can be generated. An agent development framework [5] is used to implement the autonomous agents responsible for carrying out the distributed VN mapping algorithm. These agents are deployed in the GRID5000 machines to emulate the autonomic substrate nodes and to handle the VN mapping framework. A declarative Agent Communication Languages (ACLs) [1] is used to define and specify the interactions and messages between the autonomic substrate nodes. Our objective is to evaluate the distributed VN mapping algorithm performance and scalability in terms of time delay and number of messages required to map one VN request to a substrate network. The aim is to study the impact of the size and topology, of both substrate and virtual networks, on the distributed algorithm performance.

An experiment has been conceived to evaluate the number of messages and time delay taken by the distributed algorithm to map different VN topologies (composed of 1, 2, 4 and 6 clusters) to different sizes of substrate network (i.e. 25, 50, 75 and 100 substrate nodes). In this experiment, each VN cluster has one hub and three spoke nodes.



Figure 6. Time delay taken by the Distributed Algorithm to map different VN topologies with different number of clusters

As depicted in figure 6, the time delay required to map a VN topology increases quasi-linearly with the increasing number of substrate nodes. For instance, the time delay needed to map a VN topology composed of one hub-andspoke cluster increases from 3.11 s to 11.48 s when the number of substrate nodes increases from 25 to 100. This delay is more noticeable in the case of mapping a VN topology composed of an important number of hub-and-spoke clusters (e.g. six hub-and-spoke clusters).

Figure 6 shows also that the time delay required to map a VN topology in a substrate grows quasi-exponentially when the number of VN clusters increases. For example, in the case of using a substrate with 25 nodes, the time delay needed to map a VN increases from 3.11 s to 18.27 s when the number of VN clusters increases from 1 to 6. This delay becomes quite considerable when the number of substrate nodes increases (e.g. 75 or 100 nodes).

The increasing time delay required to map a VN topology is induced by the increasing number of messages (e.g. MSG, NOTIFY) exchanged between the autonomic substrate nodes. Figure 7 depicts the number of messages exchanged between nodes that grows with increasing size of the substrate topology. The important number of messages illustrated in figure 7 and used to map a virtual network corroborates the delay results shown in figure 6.



Figure 7. Number of messages used by the Distributed Algorithm to map different VN topologies with different number of clusters

Our major observation is that the time delay grows with increasing number of VN clusters. This result shows that the time delay needed to map a VN topology depends on the number of clusters that compose the topology, on the one hand, and the number of nodes constituting the substrate, on the other hand.

As there are currently no known distributed VN provisioning algorithms available in the literature, at least to our knowledge, this paper is not in a position to provide any comparison with existing distributed VN mapping algorithms.

6. Conclusion

This paper presented the design, implementation and evaluation of a *distributed* and *autonomic* virtual network mapping framework. The substrate nodes integrates autonomous and intelligent agents which exchange messages and cooperate to carry out the proposed VN mapping algorithm. A VN Mapping Protocol is used to communicate and exchange messages between the autonomic substrate nodes. Results from performance evaluation of the VN mapping framework suggest that the challenge of constructing virtual networks from a shared physical network can find viable solutions.

Future work will consist of implementing and evaluating more thoroughly the autonomic substrate node. Reasoning and learning mechanisms to ensure and provide intelligence to agent based substrate nodes will be added. Indeed, the intelligence of an agent element is very limited to its rule base and can not deal with unpredictable situations. Mechanisms like *Collaborative Reinforcement Learning* [12] should be used to provide intelligence and autonomic behaviors to agent based systems. The distributed and autonomic VN mapping framework presented in this paper can be a potential starting point for self-provisioning and self-management of virtual networks in a shared substrate and certainly a framework for further investigation.

References

- [1] Fipa (foundation for intelligent physical agents), http://www.fipa.org/.
- [2] Future internet network design (find).http://find.isi.edu/.
- [3] Global environment for network innovations (geni).http://www.geni.net/.
- [4] Grid 5000, https://www.grid5000.fr/.
- [5] Jade (java agent development framework), http://jade.tilab.com/.
- [6] P. Horn, Autonomic computing: Ibm's perspective on the state of information technology. *IBM Corporation, Online reference, www.research.ibm.com/autonomic/*, 2001.
- [7] S. Bandini, et al. Mechanisms to support situated agent systems. 7th IEEE Symposyum on Computers and Communication, 2002.
- [8] R. ricci, et al. A solver for the network testbed mapping problem. ACM Computer Communication Review, 33(2):65–81, January 2003.
- [9] G. Tesauro, et al. A multi-agent systems approach to autonomic computing. *Third International Joint Conference on Autonomous Agents and Multi Agent Systems*, pages 464– 471, 2004.
- [10] T. Anderson, et al., Overcoming the internet impasse through virtualization. *IEEE Computer Magazine*, 38(4):34–41, 2005.
- [11] A. Bavier, et al. In vini veritas: Realistic and controlled network experimentation. ACM SIGCOMM, pages 3–14, September 2006.
- [12] J. Dowling et al. Building autonomic systems using collaborative reinforcement learning. *The Knowledge Engineering Review*, 21(3):231–238, September 2006.
- [13] J. Fan, et al. Dynamic topology configuration in service overlay networks: A study of reconfiguration policies. *IEEE INFOCOM*, 2006.
- [14] J. Lu, et al. Efficient mapping of virtual networks onto a shared substrate. Washington University, Technical Report WUCSE-2006-35, 2006.
- [15] Y. Zhu, et al. Algorithms for assigning substrate network resources to virtual network components. *IEEE INFOCOM*, 2006.
- [16] M. Yu, et al. Rethinking virtual network embedding: Substrate support for path splitting and migration. *Princeton University, Technical Report TR-788-07*, 2007.
- [17] N. Feamster, et al. How to lease the internet in your spare time. *SIGCOMM Computer Communication*, 37(1):61–64, January 2007.