SFR: Scalable Forwarding with RINA for Distributed Clouds

Fatma Hrizi, Anis Laouiti, Hakima Chaouchi
Telecom SudParis, CNRS Samovar, UMR 5117, France
Email: fatma.hrizi,anis.laouiti,hakima.chaouchi@telecom-sudparis.eu

Abstract—Distributed clouds have been regarded as the key enabling technology to provide new trends of services. The contribution of this paper is twofold. First, we discuss current distributed clouds architecture and highlight the issues involved, specifically scalability. Second, we investigate the benefits of the use of the Recursive InterNetwork Architecture (RINA) as a networking solution for the distributed clouds. Our proposal, called Scalable Forwarding with RINA (SFR), has been evaluated via simulations and showed to reduce the routing table size by around 75%. The paper concludes by highlighting the advantages of the RINA-based approach over current distributed clouds networking solutions in terms of scalability, simplicity and manageability.

Keywords—Distributed Clouds, RINA, network management, routing, topological addressing, performance evaluation, simulation.

I. INTRODUCTION

In the last few years, cloud computing has gained considerable interest from researchers, industries and standardization bodies [1], [2], [3]. This promising technology has enabled the deployment of a large set of use case scenarios that were not economically feasible in traditional infrastructure settings (e.g., big data analytics, mobile clouds and High Performance Computing applications). Cloud computing technology has introduced a new computing model in which resources (i.e., storage and CPU) are made ubiquitously available as general utilities that can be used in an on-demand style and at very low costs. When the computing resources are distributed in different geographical regions, we call this scattered deployment "Distributed Clouds". Distributed Clouds, can directly reach users due to their distributed infrastructure which makes large-scale applications possible to deploy. Basically, distributed clouds rely on resources where dedicated facilities are deployed in traditional datacenters. Nevertheless, another kind of distributed clouds has emerged. It consists mainly of resources scattered in offices, customers’ homes and/or data centres participating to the cloud services in a volunteer fashion. This new concept of decentralized clouds paves the way to the development of more scalable, resilient and flexible clouds. Accordingly, robust and resilient networks in terms of availability, routing and security are therefore needed to cope with the evolution of the cloud systems. VIFIB [4], [5] is an example of these decentralized "Volunteer" clouds. Mainly, it has been proposed to protect critical corporate data against possible downtime or destruction. Moreover, it is designed to enable the deployment and configuration of applications in a heterogeneous environment. By hosting computers in many different locations and copying each associated database in at least three different distant sites, the probability of mass destruction of the whole infrastructure becomes extremely low. In case of a disaster or downtime of a server, the data is replicated and the cloud continues to operate. The VIFIB system is based on master and slave design. The master controls the different computers running slaves. In terms of networking, the master and the slaves at different locations are interconnected through multiple IPv6 providers. In order to guarantee a high reliability, VIFIB uses an overlay called re6st [6], [7], which creates a mesh network of OpenVPN tunnels on top of several IPv6 providers and uses the Babel protocol [8] for computing the routes between nodes. Despite its effectiveness, the re6st has certain limitations related mainly to scalability and security. The re6st creates a flat topology that does not scale in case of large networks, which becomes an important concern for the future of the distributed clouds.

In this paper, we study mainly the issues related to scalability in distributed clouds, specifically, we consider the case of the VIFIB system. We, first, highlight the issues and limitations related to the networking architecture of the VIFIB system. Then, we propose a solution based on a new promising architecture, namely, the Recursive InterNetwork Architecture (RINA) [9]. RINA, by its design, is better suited to handle large scale networks and provide interesting benefits towards the current over IP solutions, like enhanced security or extended programmability. The objective of this paper is to demonstrate how our RINA-based architecture, namely Scalable Forwarding with RINA (SFR), outperforms the re6st overlay and gives better results. Using the RINASim [10] platform, we illustrate how SFR showed to be significantly more adapted to distributed clouds, improving the forwarding table size by around 75%. This paper is organized as follows. Section II gives some background on the VIFIB distributed clouds. In Section III the recursive RINA architecture is described. Section IV addresses the possible application of RINA to efficiently manage the VIFIB system. Section V details some evaluation studies that have been conducted in the scope of PRISTINE project [10] and using RINASim simulator. Finally, section VI reports the conclusions and provides directions for further research.

II. DISTRIBUTED CLOUDS: THE re6ST (RESILIENT OVERLAY NETWORKING SYSTEM)

In this Section, we give some background on the VIFIB system. Furthermore, we discuss the issues and challenges related to its networking system; the re6st, VIFIB is a decentralized cloud system, also known as resilient computing [4]. Resources are scattered in computers that are located in customers’ homes and in offices. The VIFIB system is based on
In this section, we introduce RINA and highlight some interesting features related to networking, routing and addressing in this architecture. RINA [9] is a clean slate architecture that introduced a new science of networking based on the "Inter-Process Communication" (IPC) model [11]. The theory of "networking is IPC" is not new as it was first introduced in the ARPANET and XNS/Ethernet [9]. In RINA, the idea is to leverage this approach and develop it with the main aim to revisit the current Internet architecture and provide a more structured model. Networking is no more a layered set of different tasks (as in TCP/IP model) but it is considered instead as a single layer of distributed IPC that could be recursively repeated in different scopes. Each layer of distributed IPC implements the same mechanisms, as illustrated in Figure 2 but policies are tuned to operate over different scopes of performance (bandwidth, range and scale). This provides a clean separation between mechanism and policy which gives considerable freedom in expressing a variety of policies. Another feature of RINA is the fact that it adopts and extends the Saltzer’s model [12] in the context of a recursive and scalable model. As depicted in Figure 2, data transfer, control and management are separated in order to isolate short, medium and long time-scales. Reiterating the IPC model over multiple levels provides a comprehensive architecture to support well-organized and well-scoped services. This "divide and conquer" concept enables scalability over large networks and avoids issues related to growing routing tables due to the limited scope of each IPC layer.

RINA with its IPC model captures the common elements of distributed applications, called DAFs (Distributed Application Facilities) [9]. A DAF is defined as the collection of "Distributed Application Processes" (DAPs), which collaborate to perform a given task. They communicate using a specific application protocol which enables DAPs to exchange structured data in the form of objects. These DAPs uses an underlying facility in order to communicate which is called "Distributed IPC Facility" (DIF). A DIF is a collection of DAPs cooperating to provide Inter-process communication (IPC). The DAPs that are members of a DIF are called IPC Processes or IPCPs. Basically, the DIF facilitates flow allocation between DAPs over different scopes. Figure 2 shows an example of two levels of DIFs (each at a given scope) used by three different DAPs. RINA by its recursive model gives more flexibility to define scalable design. DIFs are independent and isolated which assists scalability in terms of both network operation and management.

Forwarding and Addressing in RINA

Forwarding in RINA is addressed basically from two perspectives, i.e. routing inside the DIF and "routing" or mapping between DIFs. The former concerns "local" routing in the scope of the DIF itself. It is a task in the relaying and multiplexing mechanism of RINA architecture. In a given DIF, the choice of a particular forwarding algorithm is a matter of policy 2. The latter has a wider scope and it is better called mapping. It provides the mapping between the DIFs. It is managed by the DIF allocator [13] component of RINA. The DIF allocator is a function of the IPC Management that determines the set of DIFs to cross in order to reach a given destination. If the desired application is not on one of the available DIFs, then a new DIF can be created to enable the communication. In order to take full advantage of RINA environment, an intelligent mapping should be implemented in the DIF allocator to manage the DIFs creation and the inter-DIFs routing. Especially, in dynamic large-scale scenarios, dynamic creation and suppression of DIFs should be considered. The idea is to use the DIF allocator to create the needed DIFs ensuring the connectivity between communicating IPCPs. The creation of the DIFs will be done dynamically and on demand following RINA logic. This will support scalability as we optimize the use as well as the management of the DIFs.
As mentioned above, RINA adopts the Saltzer’s naming and addressing schema. It has a complete naming and addressing model, providing names for the basic entities of the network (applications, nodes, and Point of Attachments). As a consequence, RINA supports mobility and multi-homing inherently. Moreover, it is worth mentioning that in RINA there is no a global address space. Addresses are private and managed in the limited scope of the DIF which will solve the issues related to address scalability.

IV. APPLYING RINA TO THE VIFIB SYSTEM

In this section, we investigate the application of RINA to the distributed clouds, and more specifically, the VIFIB system. In the scope of this paper, we will focus on the system architecture from routing and addressing point of view. In the following, we will address the global architecture of DIFs ensuring the exchange of data between the DAPs and we introduce our proposal for routing and addressing: Scalable Forwarding with RINA (SFR).

SFR: Scalable Forwarding with RINA

Naturally, in distributed cloud systems, an increasing number of users would affect the performance of the cloud services, as resources are very limited while the requirements from the applications are growing. Accordingly, a solution to support scalability is needed in such large scale environment. In this work, we propose to adopt the divide and conquer concept inspired from RINA in order to have a hierarchy of smaller clouds providing connectivity between the pairs of the system in an efficient way. The main idea of our proposal is to divide the clouds into groups or regions. These groups are created and managed by the authorities based on a specific criterion, e.g., the group size, the country and/or the ISP membership. Furthermore, connectivity between the groups is ensured by inter-connecting a set of VIFIB nodes of each group. This set of VIFIB nodes, namely “Groups Leaders”, is elected to act as relays between the groups and to form specifically what we call the inter-groups. At the same time they preserve their membership to their original groups. In order to further scale, this “logical” organization could be repeated recursively adding other levels that will be forming a logical hierarchy. To avoid link failure problems due to the bandwidth limitation of the nodes, several VIFIB nodes could be elected from the same region as Group Leaders providing more resilience. The way these Group Leaders are chosen needs further investigation.

Figure 3 illustrates an example of two levels of Inter-Groups hierarchy. Group S is the group where the originating VIFIB node A belongs. Group D is the group where is the destination VIFIB node (F). To represent this scenario in RINA logic, we draw Figure 4. We assume that for each region a DIF is created to manage connectivity inside the group. Consequently, Each VIFIB node has at least one IPC Process in the groups of the overlay (the lower level of the hierarchy). Some of the overall VIFIB nodes that we called “Group Leaders” will have also IPC Processes in the inter-groups on the upper logical levels apart from the IPCPs belonging to the Group DIF. Suppose that VIFIB node A in Group S is the source node and node H in Group D is the destination. Node A has one IPCP connected to the Group S DIF which connects to node B that acts here as the group leader. Accordingly, VIFIB node B has one IPCP within Group S and one additional IPCP within InterGroup1 that connects it to node C in the scope of the InterGroup1. Node C has three IPCPs: One within InterGroup1, one within InterGroup2 and at the same time one within Group1 in the lower group DIFs. Node E has two IPCPs: One within InterGroup1 at level 1 allowing connection with the node D. Moreover, it has in particular one in Group D where the destination VIFIB node F belongs. Node E will use the Group D DIF to reach directly the destination.

In Figure 4, we illustrate the DIF architecture of the overlay cloud in the considered example. Services provided by the distributed cloud system are deployed using “App-DAFs”.

Fig. 2. RINA Reference Model [9].
App-DAF is a collection of Distributed Application Processes (DAPs) that will be sharing information (DAP 1 and DAPN in the example). These DAPs use specialized overlay “Tenant App-DIFs” that are tailored to the needs of the App-DAFs. Tenant App-DIF is destined to connect DAP1 and DAP2 in order to support their communication process. On the other hand, the tenant Cloud DIF is designed to adapt to the dynamic network connectivity. Especially for distributed clouds where VIFIB node could act as Border routers and at the same time as customers application. The tenant Cloud DIF ensures scalability and flexibility as it maintains a global view of the network to manage dynamically the possible suppression/appearance of the lower DIFs structure which could be very frequent in the distributed clouds scenario. At the bottom, each group is mapped to a DIF which is created accordingly to manage connectivity inside the region.

To summarize, there are basically four types of DIFs:

- **Tenant App DIFs**: DIFs that provide the direct connectivity between hosts. Mainly they are used by the customers of the Distributed Cloud system. These DIFs are directly supported over a tenant Cloud DIF.
- **Tenant Cloud DIFs**: Medium-sized DIFs that provide connectivity between VIFIB nodes from different regions. These DIFs could be created dynamically on demand in order to adapt to the frequent change in the network connectivity.
- **Inter-group DIFs**: Small DIFs that provide connectivity to Group Leaders of some Group DIFs.
- **Group DIFs**: Small DIFs that provide high connectivity and low latency between the VIFIB nodes of the same small region.

**Dynamic Creation of Tenant Cloud DIFs**

Some DIFs could be pre-configured to support the connectivity between customers that are frequently communicating. However, in order to support scalability, Tenant Cloud DIFs should be created on demand which means that when resources are requested to be allocated between source and destination, the DIF allocator will be in charge to build (if not existing already) the required Tenant cloud-DIFs to ensure the connectivity. If a customer’s VIFIB node do not find a common DIF where it can see the destination, it should query peer DIFs which may have a DIF with the destination Application process. The path followed by the search request will be the sequence of the DIFs to use given by the DIF Allocator. In this way, we can efficiently manage the use of cloud DIFs as they are created/used on demand.

**Routing Policies**

The routing algorithm to be run in the DIFs will depend essentially on the built DIF hierarchy. In the Groups DIFs at the lowest level of the hierarchy, VIFIB nodes have to store routes leading to the Group Leaders. Consequently, traditional routing e.g. link state or distance vector could be used. Then, a Group Leader can determine the next hop based on topological addresses. Traditional routing might be used also inside the groups of the upper level layers of the hierarchy if needed. In the next section, we will investigate the address assignment aspect.

**Topological Addressing**

Figure 5 illustrates an example of topological address configuration that could be deployed for each layer. Basically, each layer has its own address space that is independent of the adjacent layer. However, the upper branches of the hierarchy may have a topological relation with lower layers which could simplify routing calculations. Addresses belonging to the same authorities or located in the same geographic place could be similar. Moreover, for each layer in the hierarchy more granularities should be provided. In the example in Figure 5, in the lowest level of the hierarchy the address is built from country prefix (FR, TN), ISP prefix (SFR, TNTEL) and number of nodes (150, 160). As the scope becomes larger in upper levels, we lose more and more granularity in the address configuration i.e., the address space at the upper level in the example is built from only country prefix and node number and there we can find VIFIB nodes belonging to different countries where addresses start with FR, TN and so on. Following the whole address, the needed path could be found. Accordingly, the topological address defines the concept of nearness. This provides location dependence without route dependence [12].

**V. PERFORMANCE EVALUATION**

In this section, we evaluate the performance of our scheme for distributed clouds. We assess the benefit of the application of RINA to the VIFIB System in terms of limiting the routing table size. Moreover, we perform a comparison of SFR with a simple Distance Vector routing protocol in order to show how it outperforms the current routing architecture that the VIFIB
System is using. In the following, we introduce the simulation setup and present the results of our experiments.

A. Simulation Scenario

We have conducted a set of experiments to analyse the performance of our proposal. We have used RINASim [10]. It is a simulation platform implementing RINA architecture in Omnet++ [14]. It is intended to enable the study of RINA architecture and also to perform simulation experiments with RINA applications. Figure 6 represents the scenario that we set up in RINASim. It consists in a medium size network of 120 nodes, divided into four regions, within each region 30 VIFIB nodes including the group leader. All the nodes inside the regions are interconnected randomly and connected to the group leader. All the Group Leaders are interconnected among each others. The DIF architecture is organized as follows:

- A Group DIF is constructed to regroup all the VIFIB nodes inside each region.
- Three inter-group DIFs are designed to interconnect region 1/2, region 2/3 and region 3/4.
- A cloud tenant DIF that contains all the nodes that are communicating.

Figure 7 shows how the configuration of routing is performed for each DIF. In RINASim, several policies have been implemented in order to handle routing within RINA networks. For example, in Figure 7, “SimpleDV”, a distance vector routing policy, is used in the scope of each DIF. Table I summarizes the different configuration parameters used for running the simulations. In this scenario, we consider that VIFIB nodes use a ping application to communicate where the maximum packets size is set to 1500 bytes.

### Table I. Simulation Parameters Configuration

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of VIFIB nodes</td>
<td>120</td>
</tr>
<tr>
<td>Number of regions</td>
<td>4</td>
</tr>
<tr>
<td>Number of VIFIB nodes per region</td>
<td>30</td>
</tr>
<tr>
<td>Application</td>
<td>Ping</td>
</tr>
<tr>
<td>Packet size</td>
<td>1500 Bytes</td>
</tr>
<tr>
<td>Ping Starts at</td>
<td>140s</td>
</tr>
<tr>
<td>Ping rate</td>
<td>5</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>300s for each run</td>
</tr>
</tbody>
</table>

B. Simulation Results

In this section, we demonstrate the assets of applying RINA to distributed clouds and present the results obtained by simulations. Figure 8 illustrates the PDU forwarding table size with regards to the simulation time. Basically, it provides a comparison of SFR with a distance vector routing protocol. The distance vector routing protocol used in this comparison is similar to the routing protocol used in the rest architecture of the VIFIB distributed cloud system [8]. We observe that SFR shows better results. In case of SFR, as expected, the routing table size does not exceed around 30 entries which corresponds to the number of VIFIB nodes in the regions. Only Group Leaders, will have additional entries corresponding to the links between other Group Leaders covering the intergroups DIFs. We can see that in case of Distance vector protocol, the PDU forwarding table size goes to around 120 entries corresponding basically to the whole network size. This is mainly due to the beneficial impact of the use of RINA in the forwarding scheme and specially its "divide and conquer" strategy that helped to bound the routing table size.

We have assessed also the dynamic creation of tenant cloud DIFs. Figure 9 depicts the PDU forwarding table size in tenant cloud DIFs considering several flows (involving 5, 12 and 21 nodes). The size of the forwarding table is plotted with respect the simulation time. We can see from this figure that...
the size of the forwarding table is proportional to the number of nodes constructing the flow. Only nodes participating in the communication have entries in the forwarding table of the tenant cloud DIF. Red line in Figure 9 represents the results for a flow between only five nodes, so we can observe that at the end of the simulation the entries of the forwarding table of each node is filled with only five entries. So, only nodes actively communicating appear in the forwarding table. Accordingly, we can efficiently manage the use of cloud DIFs as they are created when needed and on demand. We conclude that managing dynamically the tenant DIFs ensures more limitation to the forwarding table size and thus, more flexibility and scalability.

VI. Conclusion

In this paper, we presented a new and generic architecture for routing and addressing tailored to cope with the distributed clouds requirements most notably in terms of scalability, reliability and efficiency. We have identified the limitations and issues of the current implemented solutions for distributed clouds. We have investigated the case of a real distributed cloud use case (VIFIB). We then described SFR, our generic networking architecture which has been designed to benefit from RINAs recursive architecture features. SFR assumes that all nodes in the distributed clouds are located in an overlay. SFR then builds a hierarchical DIF architecture from the overlay to manage efficiently the forwarding in the network. The obtained simulation results showed that SFR achieves its design goal by limiting the routing table size compared to the simple distance vector routing protocol used currently by the VIFIB system. An aspect that we further investigate is the dynamic behaviour of SFR, we showed how it adapts to scalability as DIFs in the architecture are created on demand.

In future works, we plan to further study our SFR architecture and evaluate the assets of using RINA in larger distributed clouds scenarios. Moreover, we intend to consider other performance evaluation metrics (i.e. latency and throughput) to efficiently assess our proposal and demonstrate its applicability to distributed clouds services. Moreover, based on the evaluation results, we intend to implement SFR in the RINA Linux Software Development Kit being developed currently by the FP7 PRISTINE project[10] which is based on the prototype implemented within the FP7 IRATI project [15], [16]. We also plan to deploy our solution within the VIFIB infrastructure in order to compare it with the current refast overlay.

ACKNOWLEDGMENT

This research is partly funded by the European Commission through the PRISTINE project (Grant 619305), part of the Future Networks objective of the Seventh Framework Programme (FP7).

REFERENCES