New Peer-To-Peer System Approach for Service Deployment

A. R. Cardoso, A. Serhrouchni
Télécom-Paris, LTCI-UMR 5141 CNRS, Paris, France +33(0)1.45.81.80.90, +33(0)1.45.81.79.94
M. Salaün
FranceTelecom R&D, CORE/M2I, 2 22300 Pierre Marzin Av. Lannion, France
J.Celestino Jr
State University of Ceara 1700 Paranjana av. Fortaleza-CE, Brazil

Abstract- Active networks have been researched for years and many studies have been done in the active node architecture, the active networks protocols and the needed functionalities. However, some of these functionalities do not achieve satisfactory results. The deployment of services in active nodes still remains a challenge, and consequently inhibits the widespread use of such networks. In this paper, a new concept based on a Peer-to-Peer system to enhance the service deployment in active networks is presented. This P2P system is used by active nodes to advertise theirs capabilities, which enables "control active nodes" to perform the service deployment, depending on the service requirements and constraints. The concept of administrative domains is also taken into account for widespread active networks. The originality of this work lies on the complete integration of the active networks and P2P systems with the goal of rendering service operators the capacity to deploy new services according to each request and the availability of the resources.

I. INTRODUCTION

The context of the deregulation and the increasingly number of alternate service operators stimulate and accelerate the introduction of new services. Competition among service operators has simplified data transmission turning it into an added-value service. The transport of information and services is currently at different levels. The needs of the service operators are to deploy demanded services on networks managed by them. It is necessary "to open" the equipment to achieve this goal. Another need of service operators is to place resources nearby their requests to achieve better performance.

Active networks are composed of active nodes, located inside the network, with the ability to perform processing tasks on flows passing through them. This enables to deploy new added-value network services and to enhance the quality of end-to-end application services. This concept increases the complexity of current networks, since it features the “open of equipments” and distributed computation throughout the network. However, most active networks deployment process requires an explicit specification of which nodes will be used by the application. This means a complete understanding of the underlying network infrastructure by the entity in charge of deploying the code (the administrators of the active networks, the service providers or the end-user), and so it inhibits the widespread use of active networks.

A Peer-to-Peer (P2P) network may be characterized by the capacity of the participants to share their own hardware resources (CPU, memory, bandwidth, printers). These shared resources provide the service and content offered by the network.

P2P systems can be viewed as systems that apply P2P networks [1] and may be defined as distributed systems consisting of nodes that can join and leave the (P2P) network at any time and that have equal capabilities and roles, without any centralized control. They provide interesting features such as wide-area routing architecture, efficient search of data items, redundant storage, permanence, fault-tolerance, self-organization and massive scalability. However, generally targeting on filesharing and instant messaging, existing P2P systems do not provide a generic support for arbitrary distributed applications. This means that they are specialized towards specifics purposes. Therefore, new P2P systems have to be created and installed to support new purposes.

Combining the technologies of P2P and active networks, this paper presents an innovative concept, the use of a P2P system to enhance the service deployment in an active network. The concept of administrative domains has been taken into account in the solution presented, electing one active node as a "control node", which is the contact point for other administrative domains. The P2P system is used by nodes and by "control nodes" to advertise their capabilities inside one administrative domain and among different administrative domains. This work is related to the control plane of active networks, based on P2P systems and can be classified as a new approach for service deployment.

This paper is structured as follows: Section II presents active networks highlighting the deployment of services. Section III identifies the classes of P2P systems. The layered P2P and active network is described in section IV. Section V focuses on implementation issues. Finally, the conclusion of this paper and future works are presented in section VI.

II. ACTIVE NETWORKS

The difficulties in deploying new services in traditional IP networks led to the idea of active networks. The key idea in such networks is to deploy services, inside the network, which perform processing tasks in intermediate nodes. In order not to waste node resources, those services must be dynamically deployed when needed and removed when no longer necessary.
A. Active Network Services

Services in Active Networks can be implemented in two different ways: (i) a capsule (or active packet) form in which data and program are carried together in each packet and (ii) a more conventional form in which the program is referenced in the packet and uploaded separately in the nodes traversed by the packet. This leads to three approaches [2]: (i) Active Packets, fundamentally characterized by the fact that the program is carried inside the packets; (ii) Active Nodes, in which packets do not carry the program but instead carry some identifiers to predefined functions that reside in the active nodes; and (iii) Active Packets and Nodes, in which packets carry only lightweight programs and more complex programs (heavyweight programs) reside in the active nodes.

This paper is related to the Active Nodes approach and it deals with the task of up-loading service codes in active nodes. In other words, it deals with the challenge of deploying services in active nodes.

B. Deploying Services in Active Networks

Deploying new services in Active Networks has been covered in the literature [3] [4] [5]. These studies take into consideration two levels for service deployment in an active network and a comprehensive framework must support them at both levels: (i) the Network Level (or macro deployment) comprehends the identification of the nodes that match the service requirements and are suitable to execute the requested service; and (ii) the Node Level (or micro deployment) consists of the installation and configuration of the appropriate service code within the node environment. Whereas the former has a network-wide scope, the latter has a node-local scope. Due to the loose coupling between the two levels, the design of the service deployment mechanisms at each level can be considered independently.

The system designer has the choice between in-band and out-of-band service deployment. The former refers to a system in which the service is deployed in the same way as the payload data, and the latter refers to a system in which the service deployment and the payload data use different communication channels.

Reference [6] states that providing a new service within an active network environment involves several tasks.

- Service specification defines the components required by a service and describes how components are interrelated.
- Resource discovery identifies the location and capabilities of processing resources embedded within the network.
- Service mapping translates the service specification into the physical network graph while taking into account all service-specific constraints.
- Service allocation task reserves and configures appropriate physical resources.
- Service provisioning is the final task that executes all required components to provide an operational service for users.

In this paper, the focus is on the resource discovery task and the means to provide node information for the active node selection when deploying services in active networks. By providing information about their own capabilities and resources, active nodes enable control nodes to select through which nodes the packets should be forwarded in order to execute the desired service. Once active nodes have been selected, packets should be forwarded to those active nodes, possibly bypassing conventional routing. This leads to a controlled deployment as opposed to existing best-effort deployments.

III. PEER-TO-PEER SYSTEMS

Nowadays, three main classes of P2P systems may be identified: centralized, decentralized, and hybrid. The decentralized system may still be divided into decentralized but structured and, decentralized and unstructured [7]. The main difference between them is the mechanism employed to look up resources in the P2P network.

In centralized P2P systems the description and location of resources are stored in a directory of a central server. Thus, nodes query this central server to find out which node hosts the desired resource. The content always stays on the client side, never passing through the server. Such centralized approach does not scale well and has single points of failure.

Decentralized and unstructured P2P systems have neither a central directory server nor rules to define or constrain the connection among nodes. By unstructured we mean that there is no control over the placement of files that stay at local nodes. To find a resource, a node queries its neighbors employing a flooding-based mechanism. Such an approach is not suitable for extensive networks since each node evaluates the query locally and if the desired resource is not present, queries must be sent to several nodes. The lack of structure leads to scalability problems because each node must maintain a route to another node in the network. To limit the flooding, a mechanism like a TTL (Time To Live) is used. The drawback of this approach is that queries may not find a positive reply if the TTL expires before reaching the node hosting the resource. Gnutella [8] and Freenet [9] are some examples of this class of system.

In decentralized but structured P2P systems, the connection among nodes is tightly controlled and the resources are placed at specific locations. It employs the DHT (Distributed Hash Table) abstraction in the context of storing and retrieving data in the P2P network so that data object (or value) location information is placed at the node with the identifier that corresponds to the unique key data object. This structure enables efficient data discovery by using the given keys and is especially suitable for the development of large-scale network. Queries are forwarded across overlay links to nodes whose nodeIDs are progressively closer to the key in the identifier space. Content Addressable Network (CAN) [10], Chord [11], Tapestry [12], and Pastry [13] are the main examples of this class of system.

Hybrid P2P systems combine the characteristics of both centralized and decentralized models. Decentralization means extensibility, fault-tolerance, lawsuit proofing, and scalability. Partial centralization makes the system more
consistent as there are relatively fewer hosts that hold authoritative data. Every user elects its super-peer, which is the "central server" for "local nodes" and can communicate with other super-peers. FastTrack/KaZaA, eDonkey are some examples of this system.

IV. A LAYERED PEER-TO-PEER AND ACTIVE NETWORK

In current works, service deployment is often closely coupled with the underlying active network technology. The proposed architecture herein is more open since the service deployment infrastructure can be deployed over any active network platform. This can be achieved by using a P2P layer over the active network layer.

Fig 1 depicts the layered P2P and active network. The active network layer, placed over the traditional network layer, is composed of two planes: the active network plane containing a subset of network layer nodes and the control plane assembling some of these nodes as control nodes of active domains. Above the Controlled Active Network Layer, it is the P2P layer, used by the service deployment, running in all active nodes.

A. The Controlled Active Network Layer

As show in section II, services in active networks are implemented in both capsule and referenced form. These services are activated and operated as the packet passes through the network. This can be considered as a best-effort deployment, as the resources available at each node for the execution of the packet cannot be predicted: they depend on the route taken by the packet. This paper proposes to address the problems of best-effort deployment and, like [14] [15], to manage the active network and control the code loading by adding a control plane over the active network plane (Fig. 1).

Fig 1. Layered Architecture of P2P and Active Networks

- **Active Network Plane:** this plane gathers the standard active nodes which offer the basic primitives allowing access to the local resources and the processing of data passing through the node. They access the capsule fields (addresses and data), consult the routing tables or apply the routing functions to route the capsules according to their destinations. They also manage the resources assigned to the capsules, control their right to execution, and communicate with the control nodes.

- **Control Plane:** this plane gathers the control nodes in the active network and a single administration node. The role of the administration node is the configuration and the administration of active domains and control nodes as well as the authentication of incoming active nodes. The administration node must also store services in control nodes in order to guarantee the control of which services may be deployed in the overall system.

The role of control nodes is the network control at the active domain scope, by storing the services, active node information (attributes), and deploying code when it is not (or no more) present in an active node. Code loading occurs either at the beginning of the application, after an active node request, when the control node deploys the service code in all suitable active nodes or at the reception of an unknown capsule type by an active node when such service code is no longer stored in local cache. All these control nodes communicate among each other to have a global view of the network.

In order to better structure our architecture, the notion of administrative domain, called Active Domain (AD), has been designed in the proposed system. In our definition, an AD consists of a control node and one or more active nodes. Each control node has autonomy in its AD managing and controlling the active nodes located in this AD. Each control node updates the routing table, which corresponds to the active nodes of its AD, according to the deployed active services. It controls the deployment of services in the system identifying appropriate active nodes that match the service requirement and are suitable to execute these services.

The following steps describe the arrival of an active node at the system:

- The active node connects to the administration node in order to be authenticated;
- After the authentication process, the administration node sends the id of the control node through which the active node must connect;
- The active node connects and sends its attributes (CPU, bandwidth, etc.) to the control node.

After the authentication process, the administration node does not participate in the suite of “active nodes/control nodes” interactions. Thus, each time an active node wants to start an application, it sends requests to the control node that is enabled to control the deployment of the desired service.

In most active networks design, the active nodes directory is centralized. This study proposes a new approach based on hybrid (decentralized at inter-domain and centralized at intra-domain) system to get node information. A Peer-to-Peer system is proposed to achieve the proposed functionality.

1. What Active Nodes Become Control Nodes

Active nodes with high capacity (bandwidth, CPU, memory) and availability are strong candidates to be control nodes.

The choice to stay an active node or to become a control node is usually an AD local decision. When an active node starts up it becomes an active node. At the moment an active node detects enough capacity and availability, it can change to a control node if an existing control node leaves or has reached the capacity limit. However, some active nodes that are known to have enough capacity and availability can immediately transition to control node upon startup.
Service deployment in Active Network based on Active Nodes approach involves pre-installation of programs in active nodes before packet arrivals. The installation procedure can be done with a signaling protocol that loads a trusted program into active nodes, and the program can be applied to subsequent packets [6]. According to this paper, the P2P system is used as a resource discovery process to get network resources from active nodes in order to determine the active network topology. Due to the lightweight features of P2P systems, these systems are suitable for disseminating node specific capabilities to the service deployment modules in all active nodes. As it is difficult to distribute all the useful node information throughout the networks or the available resources that may vary rapidly, our proposal gathers only relevant information for service deployment. We followed the service requirements presented in [16] to guide our selection: topological requirements dealing with information about location, and resource requirements dealing with capabilities and availability of network resources.

As in [17], our P2P system locates specific nodes and obtains information about their properties, and can be divided into the following tasks:

- Monitoring local resources: Each active node is responsible for its local available resources.
- Advertising local resources: Each active node distributes information about its local available resources. This task may be done following either a centralized model among active nodes of the same active domain, or a decentralized model among control nodes of different active domains.
- Gathering information: Control node stores information about the active nodes of its Active Domain and information advertised from other control nodes.
- Propagating information: All information cached from other control nodes is also propagated throughout the active domains. That is, when a control node receives a state advertisement from a neighbor control node, it sends this information to all its directly attached neighbors.

To allow the Resource Discovery task, we propose a solution based on a hybrid P2P system. Locally, it consists of “small” centralized P2P systems, related to administrative domains, being controlled by one control active node acting as a super-peer. To allow a scalable active network, these super-peers are interconnected implementing a decentralized but structured P2P system (Fig. 2).

V. IMPLEMENTATION

A. The Active Network Implementation

Active Network Transport System 2 (ANTS 2) with some extensions based on [14] has been used. The extended ANTS 2 is an Active Node approach where packets carry references to the code. Thus, at the network level, service deployment can be classified as a distributed/in-band approach [3]. In the node level, since packets carry only the reference instead of the code, our extended ANTS 2 uses a centralized/out-of-band approach. That is, as a packet arrives in a node, if the code is not locally cached, it is retrieved from a central code server.

In classical ANTS 2 architecture, the active nodes introduce new services in the system. However, in presently proposed architecture this is a task of control nodes, which must possess the service code. A client must first obtain the code that implements the desired service before it is able to use that service and thus anyone should be able to deploy his own service anywhere. An active node which requires a service must belong to an Active Domain, and establish a relation with a control node responsible for such AD.

In order to achieve the solution proposed in section IV, we extended some core classes of the ANTS 2. However, no modification in the node runtime (Node.class) was needed. The core changes are the ConfigurationManager, PrimordialNode and ProtocolBooster classes. They must be extended to deal with the new situation of the proposed solution (e.g. send demanding-load requests to control node instead of the previous node, etc.).

As in [14], the modified version of ANTS 2 provides some APIs corresponding to the component of control (control node application) added to the platform, through which different, useful and short processing routines can be implemented (e.g. addService(), removeService(), viewService(), setControl(), getControl()…).

B. The Peer-to-Peer Implementation

The hybrid P2P system is chosen because of its capacity to deal with several types of nodes and situations. Therefore, it is suitable to our Controlled Active Network Layer.

As previously mentioned, after an AD local decision, active nodes with high bandwidth, CPU and memory are made control nodes, responsible to manage the connection of the other nodes in a centralized way. Only these control nodes form a DHT. The Chord [11] system was chosen “to structure the decentralized system” due to its features of lookup performance.

Besides being responsible for the ServiceTable, the control node, at this layer referred as a server peer (Fig. 1), deals with the Chord Tables (e.g. FingerTable, SuccessorTable, PredecessorTable), as well as the LocalAttributesTable and the ExternalAttributesTable.
Both the Local and the External tables store active node capabilities needed to service deployment. The former is used by control nodes on intra domain service deployment. Thus, once connected to a control node, each active node registers its IP and Attributes in the LocalAttributesTable. The latter is used by control nodes on inter domain service deployment. Thus, after obtaining a key k by hashing the IP of the active node, the binding k/attributes is stored in the ExternalAttributesTable of the control node which its controlnodeID succeeds immediately k.

Like [18], our proposed decentralized P2P system is subdivided into three sub layers: Chord Lookup, DHT Storage and Evaluation Application. The main idea is to use a layer that translates high-level names into Chord identifiers (or keys). Chord is only used as a lookup service. By layering additional features on the top of a core lookup service, the whole system gains robustness and scalability. The functionality of each sub layer is described here:

**Chord Lookup:** this sub layer only provides mapping functionalities. As defined in [18], to implement these mapping functionalities, this sub layer maintains a routing table (Finger Table) at each control node with information about O(log N) other control nodes, where N represents the number of control nodes. It is not responsible for storing the binding key/attributes; instead it provides to the upper sub layer an efficient way of looking up the control node responsible for its storage.

**DHT Storage:** placed over the Chord Lookup sub layer, the DHT Storage uses the SHA-1 hash function to produce identifiers on a circular space. According to the proposal outlined in this paper, each control node has a unique ID (controlnodeID) obtained by hashing its IP address. Each active node IP is also hashed to obtain a key k and the key/attribute binding is stored at the successor of k, that is, the controlnodeID that follows the k value. The DHT uses the Chord lookup service to locate the control node responsible for an active node.

**Resource Discovery Application:** this sub layer is responsible to provide interfaces of active node attributes to the control node application. It is consisted of five main APIs: insert (key, attributes), lookup (key), update (key, newattributes), join (activenodeID), and leave (activenodeID).

### C. Service Deployment

The service deployment begins when an active node sends a request to the control node specifying the service to load, as well as the source and the destination nodes.

To deploy the service, the control node needs to know the capable active nodes to support such service as well as the route to be taken. After choosing the route to be taken by active packets, the control code installs the required service in all active nodes that will belong to the service up to the destination.

Active services are initially stored on each control node. However, they are purged from the local node cache, after a fixed amount of unused time.

The network administrator can add and remove active applications, in a very simple way. Once this work is based on an Active Node Approach, for the development of active applications we provide a base class with a key function that the active application developers can customize by inheriting from the base class.

Once the needed custom processing routines are injected into the required nodes, the users will send their active packets through the active nodes the way they do in legacy networks. When an active node arrives at an active node, it is executed and, forwarded to the following hop if needed.

In our proposal, we choose to separate mechanisms for loading and execution, in order to be able to carefully control program loading.

The position of active nodes defines the level of service deployment. That is, if active nodes selected to be part of an application belong to the same active domain, the service deployment will occur in the intra domain level; otherwise it will occur in the inter domain level. These two levels are described in the next subsections.

#### 2. The Intra Domain Level

The Intra Domain Level uses a centralized system, in which the attributes of active nodes of an AD A are centrally stored in a table (LocalAttributesTable) of the control node A. Thus, it is enabled to provide the ability to search and choose suitable active nodes for service deployment on its active domain by executing a local search.

#### 3. The Inter Domain Level

It uses a decentralized but structured system. Two approaches can be employed at this level: the lookup of the control node responsible for the administrative domain by launching searches in a Chord ring; or a copy of active node attributes is stored in a table (ExternalAttributesTable) of other control nodes.

This allows the communication among ADs and consequently, service deployment in active nodes of different ADs. As this storage employs a structured solution of new decentralized P2P systems presented in section III, each control node deals with small tables containing information of only few active nodes of other ADs.

### VI. CONCLUSION AND FUTURE WORKS

We have presented a layered network architecture to improve the dynamic deployment of service for active networks. Defining the active/control planes, the controlled active network offers a more reliable and efficient service deployment. Placing a P2P layer over the controlled active network, active node attributes can be disseminated throughout the network without suffering from scalability issues. Thus, control nodes are enabled to select through which active nodes packets should be forwarded in order to execute the desired service.

Once the control node identifies a suitable set of active nodes using the P2P layer, the service code is automatically and dynamically deployed to those specific active nodes in a single transfer. This leads to a controlled deployment as opposed to the best-effort deployment.

However, further work on the current prototype is needed in order to get a complete interoperability with many active network platforms, not only ANTS2. Some performance tests must be performed in order to know how efficient the proposed architecture is, in terms of the resource discovery process based on the Chord lookup.
Works on an active network simulator like [19] is also being developed in order to achieve these goals.

The contribution of the proposal outlined is the integration of P2P and Active Networks to make available for a service operator the capacity to deploy new services according to each request and the availability of the resources.

REFERENCES


