Decentralized Resilient Grid Resource Management Overlay Networks

Yangcheng Huang, Saleem N. Bhatti
Y.Huang@cs.ucl.ac.uk, S.Bhatti@cs.ucl.ac.uk
Networks Research Group, Department of Computer Science
University College London, Gower Street, London WC1E 6BT, UK

Abstract

Recently, Grid and peer-to-peer overlays have attracted attention from research communities as well as industry. However, the overwhelming majority of industrial, commercial and academic activities are geared around applications, servers, and middleware. There have been some efforts in the management of the underlying networks (the connectivity for the overlays), but far from enough.

This paper proposes a resilient, fault-adaptive, overlay resource management framework, with lightweight state management infrastructure, taken from some ongoing work by the authors in this area. This research aims to solve the problems of: (a) managing distributed network resources in a decentralized way; (b) providing resilient QoS for highly dynamic networks.

1. Introduction

The Internet provides a single-class best-effort service. Increasingly, users require that IP networks be used to carry data with specific Quality of Service (QoS) constraints. The network will require, potentially, a large amount of control and management information to provision, maintain, validate, and bill for these new services. In addition, the devices using Internet connectivity are varied, having differing management capabilities, ranging from complex computing and switching platforms to personal, hand-held devices. The control and distribution of management information in such a network is a challenging task.

In the past, the common framework for solving the management needs has been based on a centralized approach using a (mainly) centralised, agent-manager (client-server) architecture, for example SNMP. Whilst providing a very straightforward and well-understood paradigm to the problem of management of networks, this approach and its variations have not proved to scale well or allowed the flexibility required in today's modern data networks [3].

In recent years, a decentralized approach to the management architecture has gained momentum. Overlay networks receive much attention as the basic architecture for managing connectivity in heterogeneous network [10] [11] [17]. The overlay network abstraction provides flexible and extensible application-level management techniques that can be easily and incrementally deployed despite the underlying network. However, there are several issues to be addressed.

A decentralized framework may have advantages with regards to scaling and speed of operation, but information and state management becomes complex in this approach, resulting in additional complication in developing such systems. The complexity of managing a network increases dramatically as the number of services and the number and complexity of devices in the network increases [3].

Also, although an overlay network architecture might facilitate end-to-end management and control, it introduces new issues in resource management.

To solve these issues, this paper puts forwards a decentralized resource management overlay framework, which aims at providing underlying network resource management for overlay networks and resilient Quality of Service (QoS) service.

The remaining sections are arranged as follows. In Section 2, we further define the problem and analyse potential challenges. Section 3 introduces past efforts within the problem space concentrating on several examples. In Section 4 and 5, we will present the proposed architecture and related mechanisms. Finally, we draw conclusions and list future work in section 6.

2. Problem definition and challenges

There are a number of challenges that must be overcome in order to develop a scalable, robust and manageable Grid system. These arise from user requirements such as massively scalable (*ad hoc*) collaboration – support for virtual organisation (VOs), multi-dimensional service searching systems in peer-to-peer like networks and redundant computing to tolerate failure or delays [18]. Achieving these requirements contribute to the complexity of a scalable and resilient Grid resource management system.

In order to realize network-wide resource management, the solution should have the following properties.

- Scalability
- Deployability
- Resilience
- Resource allocation control capability
- Support for overlays
- Self-contained security

2.1. Scalability

Current approaches to network management do not scale sufficiently to large numbers of network nodes or large numbers of users, so network operators often have difficulty in running their network(s) as successfully and efficiently as they may like. Hence, more work is needed to improve the scalability of network management systems. Up till now, there are few tools for managing a whole network as opposed to individual network elements. Current network management protocols such as SNMP rely on reading the status and values of well-defined objects from individual devices. However, managing the network as a whole requires a viewpoint that encompasses the *individual* network elements, but treats them as a *collective* providing a service within a large distributed system. An efficient management approach would be to support distributed management in a decentralized way.

2.2. Deployable

The failure of the adoption of Internet-wide QoS [19] teaches us that the deployment issue has to be considered as a critical part of the evaluation of a management framework. Nowadays, many Internet services, especially close to the edges and the end users, are run by commercial ISPs. This brings two problems for cross-domain management. Firstly, technological differences between domains make it difficult to find a common management mechanism to be used by all ISPs. ISPs utilize different tools and deploy different policies to manage their local networks. Secondly, even if commonly agreed mechanisms and policies exist, the commercial interests (or financial constraints) of ISPs may prevent their wide deployment. To make a mechanism or architecture deployable across the Internet, it should (1) make minimal changes to the existing (core) network; (2) work and co-exist well with existing protocols and network elements; (3) bring the least traffic overhead/effects on other applications and systems; (4) have obvious benefits for end users and also operators.

2.3. Resilience

Today's IP backbones are provisioned to provide excellent performance in terms of loss, delay and availability. However, performance degradation and

service disruption are likely in the case of failure, such as fiber cuts, router crashes, etc. According to recent research in link failures in IP backbones [16], link failures occur as part of everyday operation, and the majority of them are short-lived (less than 10 minutes). The existence of such link failures requires cross-domain network resource management to be flexible and dynamic to adapt to changes, especially for advanced reservations. When failures occur, we should find an efficient way to detect them and keep monitoring the status; if the link does not recover after a period, we need to re-arrange reservations using that link.

2.4. Coordination of Resource Allocation

Like the World Wide Web, future Grid networks will offer individuals and institutions "the opportunity to build virtual organizations which facilitates the access to the problem solving services of the community" [4]. This introduces a big challenge: how to coordinate the resource-sharing problem between different parties, without wasting potentially scarce or critical resources. One potential solution is to provide the ability of reserving resources in advance. In the *Grid Resource Allocation Agreement Protocol* (GRAAP) Working Group of the Global Grid Forum (GGF), the term *advance reservation* was defined as "a possibly limited or restricted delegation of a particular resource capability over a defined time interval, obtained by the requester from the resource owner through a negotiation process" [3]. Support for such advance reservations will be a necessary part of resource management infrastructure.

2.5. Support for Overlays

Overlay networks augment the Internet service delivery to the end user by virtualising the use of connectivity and resources for the user. Features like application-layer routing and QoS routing can have great impact on performance of grid applications. In addition to direct support for end users, the resource management architecture should also provide APIs for the control of overlay networks, and allow overlay networks to improve their functionality with resource management capability.

2.6. Self-contained Security

The management framework should aim to be self-contained with respect to security services and should be at least as secure as the connectivity service it is trying to manage. That is, it must not introduce new vulnerabilities to whole (or part of) the network. This may not always be possible. For example, the management system may use user credentials (including certificates) that need to be verified by authenticating through a third-party service.

3. Previous work

A huge amount of effort has been put into the area of network management to achieve some of the properties listed above ([6, 10, 17]), in different environments (Internet, peer-to-peer networks and Grids). Some of them have been successfully implemented into toolkits (such as the Globus Toolkit). In this section, we will introduce a selection of previous work in building resilient network management infrastructures (rather than a full survey, due to lack of space).

The *Globus Toolkit* [6, 7] is "a community-based, open-architecture, open-source set of services and software libraries that support Grids and Grid applications" [5], which have been adopted widely in Grid projects. It uses GRAM (Grid Resource Allocation and Management protocol) and related services to address the issue of resource discovery and management. GRAM is "reliable" and "secure" (with GSI [8] and "gate keeper" mechanisms). However, it is not resilient enough in resource (service) discovery and failure detection. It is based on a registry mechanism (that is, a resource/service can only be found after it registers itself to some resource manager), which is essentially a centralized management framework.

GARA (General-purpose Architecture for Reservation and Allocation) is a comprehensive and flexible

architecture for providing applications with QoS for different types of resources (network resources, CPU and storage resources). For network resources, GARA uses DIFFSERV with EF [9] to provide both advance reservations and immediate reservations. GARA relies on Globus, and does not provide any essential mechanisms to discover the availability of resources. So it can only be viewed as a high-level "user interface" layer of resource reservation. Additionally, GARA does not scale well to operation across multiple administrative domains.

SNMP (Simple Network Management Protocol) is an application layer protocol that facilitates the exchanges of management information between network devices. However, SNMP may not be able provide the troubleshooting information required on a per domain basis across a network as a whole. In addition, use of SNMP for multi-domain management is impractical for a number of reasons but especially because of the lack of the implementation of an agreed security framework. Although the RMON (Remote Monitoring) extension to SNMP can enable various network monitors and console systems to exchange network-monitoring data, it is from the view of "specific network devices", not the view of "domains" or whole networks. Furthermore, SNMP relies on vendors to provide extra management applications, which differ greatly in terms of functionality and visualisation of the data. Thus it is only suited for network monitoring and capacity planning within individual domains, not for resource management across domains.

X-Bone is designed for automated deployment, management, coordination, and monitoring of IP overlay networks to reduce configuration effort and increase network component sharing. X-Tend extends the X-Bone overlay deployment tool to support large-scale network emulation, dynamic topology control and additional new protocols and platforms for research and classroom use. It features security solutions and deployment of multiple concurrent overlays. However, X-Bone does not take cross-domain management issues into consideration, such as cross-domain signaling. Thus X-Bone can only work as an intra-domain overlay network management platform. In addition, it does not provide QoS support for overlays.

Virtual Network Service (VNS) is a network service that "would enable the provisioning of virtual private networks (VPN) that are guaranteed with quality of service (QoS)" [12]. It shares the same architecture ("logically separable to the control plane and the data plane" [12]) with our work. However, it aims at adding QoS features into secured connections (similar to tunnels) between multiple geographical sites, which means it is likely to scale poorly for interconnection between large numbers of sites. In addition, designs in VNS focus on an application's resource requirements without taking into account changes in network QoS, which is one critical point of our work.

Service Overlay Network (SON) [11] is an effective means to address the issue of end-to-end QoS. SON purchases bandwidth with certain QoS guarantees from an individual network domain via a service level agreement (SLA) to build a logical end-to-end service delivery infrastructure on the top of an existing data transport network. A SON is pieced together via service gateways, which perform service-specific data forwarding and control functions. The logical connection between service gateways is provided through the use of network domains with certain QoS guarantees. However, we can see that the SON architecture simply introduces the concept of an overlay into previous end-to-end QoS mechanisms to simplify the QoS management and scalability problem. It cannot handle dynamic resource management.

BGRP (Border Gateway Reservation Protocol) [13] describes a distributed architecture for inter-domain aggregated resource reservation. BGRP builds a sink tree for each domain, which aggregates bandwidth reservations from the nodes located in the domain. Thus it scales well. However, it still does not adopt adaptive solutions to manage unexpected events like link failure.

4. Architecture

In this section, we describe the architecture of our proposed system.

4.1. Outline

We assume that each autonomous system in the Internet has one or more Network Resource Agents (NRAs). These agents cooperate with each other to form an overlay management service network and provide support for overlay networks, including resource monitoring, negotiation and allocation. A signalling system is used to maintain network resource state information. With an optimized query mechanism, network state can be probed across the domains.

4.2. Services

A *Grid service* is defined in [5] as "a Web service that provides a set of well-defined interfaces and that follows specific conventions. The interfaces address discovery, dynamic service creation, lifetime management, notification, and manageability". Our proposed architecture manages network state information, which is distributed and decentralized across the links/paths of the network. We aim to provide a well-defined API to facilitate:

1. Network State Query service, that is, to learn real-time network state information, including resource usage and link characteristics;
2. Resource Management service, including resource reservation and resource sharing issues;
3. Failure Detection (Discovery) and Recovery service, to discover link or nodes failures and favour quick recovery.

4.3. Definitions

Network Resource Agent (NRA): Each autonomous system in the Internet has one or more NRAs. These agents cooperate with each other to form an overlay management service and provide support for overlay networks, including resource monitoring negotiation and allocation. The NRA is defined as an independent element in every domain. It acts as the network resource "access point", like a resource "advisor". Compared with a Bandwidth Broker, it does not simply allocate bandwidth, but also maintains other network state information. Its duties include:

(Intra-domain)

1. Discovery and maintenance of network resource state information;
2. Detection of unexpected failure and generation of alarms;
3. Control of the behaviour of packet forwarding and the configuration of reservation information;

(Inter-domain)

4. Send resource queries to other NRAs;
5. Respond to resource queries from other NRAs;
6. Send alarm messages to other NRAs in the event of failures (e.g. inter/intra-domain link failures);
7. Send reservation messages to resource requesters;
8. Send keep-alive messages to other NRAs;

Network Resource Engine (NRE): Network Resource Agents are connected together and cooperate with each other to form an overlay management service network. The collection of such NRAs for a control plane called a Network Resource Engine;

Packet Forwarding Engine (PFP): The functions of packet forwarding by network elements (routers, gateways etc.) form a data plane is called the Packet Forwarding Engine (PFE). The NRE realises network resource allocation by controlling the behaviours of the PFE.

Network Resource Engine API (NREA): We define the follows APIs for the architecture (see Figure 1);

NRA_Register: This API is used by NRAs to register soft-state of Grid service handles to their neighbours and acknowledge their existence. Every NRA will keep a list of its neighbouring NRAs;

NRA_Unregister: Unregister a Grid service handle;

NRA_SendQuery: This API is used by NRAs to probe inter-domain link state by sending out query messages to neighbouring NRAs and find a path to a destination AS.

NRA_ForwardQuery: If there is no local response possible for the received query message, the NRA will forward the query; when choosing next-hop NRAs, it uses

an optimized forwarding mechanism to find the most suitable NRA(s) in its neighbour, based on the *Nearest Match Forwarding Algorithm* (described later).

NRA_Monitor: This API starts a daemon process in the NRA to monitor the intra-domain network and collect local network state information, especially related to network failures. For NRAs located near edge routers, there will be another daemon process in the NRA to monitor the inter-domain link state.

NRA_SetSysHook: An NRA uses this API to bind itself into the hosting system environment(s) (*native operating system processes or* container/component hosting environments) with a call-back mechanism;

NRA_UnHook: An NRA unbinds from its hosting system environment(s);

NRA_Service_Callback: NRAs use this API to provide a "Grid service" (for a resource reservation or query on link status) for an overlay network.

NRA_DoService: This is a generic API for NRAs to interface to Grid services. Specifically, it will carry out service tasks by (1) communicating with a PFE to implement the service, including enabling inter-domain and intra-domain resource reservations; or (2) sending out query messages by calling *NRA_SendQuery* (to help make routing decisions).

NRA_listen: This API is used to start a daemon process to listen to NRA query messages from other NRAs and reservation requests from local domain hosts.

NRA_SendNotification: A collection of dynamic, decentralized agents must be able to notify each other asynchronously of interesting changes to their state. In a similar fashion to OGSA [5], NRA defines two common abstractions: *NRA_NotificationSource* and *NRA_NotificationSink* to deal with notifications (e.g., for errors and keep-alive messages) in standard ways. There are several common notifications, including a keep-alive message, request-leave (leave with a time period) and service-down (service stopped). In addition, there are also error notifications to deal with disruptive events such as link or node failure.

Other interfaces: We expect to define additional standard interfaces in the near future, to address issues such as authorization, policy management and concurrency control.

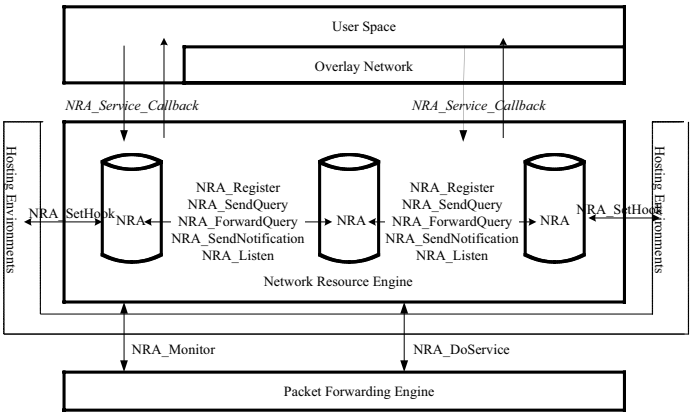

Figure 1: Interaction between different parties

5. Mechanisms

We describe here some of the specific mechanisms proposed for our architecture.

5.1. DHT-based Storage of Resource State

For a distributed system, storage and sharing of information is a critical issue. We propose the use of a system based on Distributed Hash Tables (DHTs) [14].

DHTs have been proved to be an effective way for decentralized information storage. In every NRA, there is one or more DHTs with the following data structure to store the link state information (see Figure 2).

Node i		
Flag_Bit (1bit at least)	KEY (64 bits)	Value (32 bits)
DATA_ITEM	(128.16.64.6, 128.16.64.7)	(60M, 100ns)
POINTER_ITEM	(10.1.24.0, 10.1,24.0)	(10.1.24.238)
...	...	...
POINTER_ITEM	(202.16.1.0, 202.16.6.0)	(202.16.1.1)

Figure 2: Data Structure used with DHT

In the KEY field, there are two types of data: IP address and IP sub-network. In Value field of are (1) values of link characteristics, if the targeted link is in the

local domain; (2) an other NRAs IP address, which stores the desired data, if the targeted link is in another domain.

Thus, there are two types of items. One is DATA_ITEM, which stores desired data, such as link/path characteristics. The other is a POINTER_ITEM, which stores the address of the desired data. To illustrate, we explain this in the following two examples.

D (key, value) = ((128.16.64.6, 128.16.64.7), (60,100ns))

In this example, the characteristic of the link (or path) from node (128.16.64.6) to node (128.16.64.7) is 60M (bandwidth) with (maximum) delay 60 ms.

D (key, value) = ((202.16.1.0, 202.16.6.0), (202.16.1.1))

Here, the network state information can be found in node with IP address 202.16.1.1.

In this way, these DHTs construct a distributed data storage network as follows (see Figure 3).

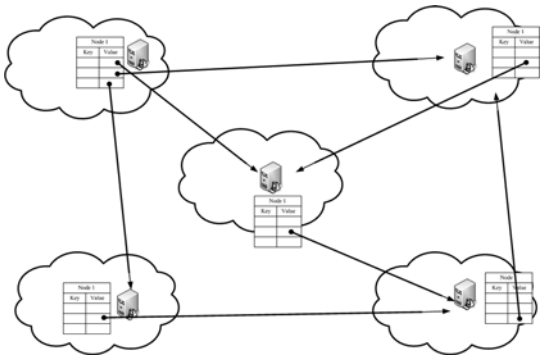

Figure 3: Storage Network using DHTs

5.2. Signalling

To discover network information across domains, we design a light-weight query/signalling mechanism, to maintain network resource state information. Since network state is stored and managed in a decentralized way, we need to send queries to the NRAs in a domain if we want to its network characteristics.

However, an NRA only knows about its neighbouring NRAs. When an NRA receives a query and cannot provide the desired data, it should forward the query to other NRAs. Simply, broadcasting or multicasting can be used, but could be inefficient. We propose the *Nearest Match Forwarding Algorithm* for such a situation.

When an NRA receives a query, if it does not find a match, in its own state table, it will match the destination link with DHT's key fields and find the nearest match in its DHT(s). It will use the value of POINTER_ITEM to forward the unresolved query.

Here we pick out a special case for signalling. NRAs located only near edge routers need to maintain inter-domain network state information. When there is a failure between edge routers, the NRAs can detect it by collecting traffic information or sending keep-alive messages. Then NRAs will look up the stored link state information. If there is another edge router NRA connected to it, it will send a query to that edge router. The queried NRA accepts the query request, and then does the same thing: looks up its own state information database and then sends a new query to the next edge router NRA. If there is no other edge router NRA, the NRA will return a NO_PATH_AVAILABLE message.

If the queries are successful, the NRAs will manipulate the routing tables and then set up a new route for the transmission.

6. Conclusion and future work

Through this proposed research effort, we aim to build a dynamic failure-adaptive network resource management framework, which aims at underlying network resource management of overlay networks and providing resilient network service.

In this approach, the infrastructure is

1. Scalable, since it has a decentralized architecture. To add a new domain (or network elements) into our management architecture, we only need to insert NRAs into the domain and make relative configurations;
2. Deployable, there is no need to make changes in core routers;
3. Resilient, when there is a failure in one or several NRAs, there should be little effect the system as a whole; NRAs work in self-organizing way, the system itself is flexible enough to deal with failures.

This is ongoing research, a theoretical framework. Work is needed to implement this architecture and evaluate its performance. There are many topics needed for further investigation, such as:

- Edge-to-edge Network Measurement to collect network resource state information; many existing tools do not work well in common network topologies, especially under heavy traffic loads, in high-bandwidth environments (e.g. OC-12 and OC-48), and with the presence of layer 2 devices of different forwarding ability;
- Detailed design of the framework, including APIs of Network Resource Agent;
- Quantitative evaluation of the proposed architecture.

References

[1] Ian Foster and Adriana Iamnitchi, "On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing", 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03), February 2003, Berkeley, CA

[2] Huston, G., "Next Steps for the IP QoS Architecture", RFC 2990, November 2000.

[3] M. Eder, S. Nag, Service Management Architectures Issues and Review, RFC 3052, January 2001

[4] V. Sander, W. Allcock, P. CongDuc, I. Monga, P. Padala, M. Tana and F. Travostino. Networking Issues of Grid Infrastructures. Working Draft, Grid High Performance Networking Research Group, GGF (Global Grid Forum).

[5] Ian Foster et al., "The Physiology of the Grid- An Open Grid Services Architecture for Distributed Systems Integration ", Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.

[6] Foster, I. and Kesselman, C. Globus: A Toolkit-Based Grid Architecture, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1999, 259-278.

[7] Foster, I., Kesselman, C. and Tuecke, S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications, 15 (3). 200-222, 2001.

[8] V. Welch et al, "Security for Grid Services", Twelfth International Symposium on High Performance Distributed Computing (HPDC-12), IEEE Press, June 2003

[9]V. Jacobson et al, "RFC 2598: An Expedited Forwarding PHB", June 1999

[10] X-Bone http://www.isi.edu/xbone/

[11] Zhenhai Duan, Zhi-Li Zhang, and Yiwei Thomas Hou, "Service Overlay Networks: SLAs, QoS, and Bandwidth Provisioning", IEEE Transactions on Networking, 2003

[12] Virtual Network Service
http://www-2.cs.cmu.edu/~hzhang/VNS/vnsmain.html

[13]P. Pan, E, Hahne, and H. Schulzrinne, "BGRP: A Tree-Based Aggregation Protocol for Inter-domain Reservations", Journal of Communications and Networks, Vol. 2, No. 2, June 2000, pp. 157-167

[14] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, Scott Shenker, "A Scalable Content-Addressable Network", ACM SIGCOMM 2001

[15] Andrew A. Chien, "Architecture of the Entropia Distributed Computing System", International Parallel and Distributed Processing Symposium, April 15-19, 2002.

[16] A. Markopoulou, G. Iannaccone, S. Bhattacharrya, C.-N. Chuah, C. Diot. "Characterization of Failures in an IP Backbone". IEEE Infocom 2004. 7-11 March 2004. Hong-Kong

[17] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, Robert Morris," Resilient Overlay Networks", Proc. 18th ACM SOSP, Banff, Canada, October 2001

[18] FutureGRID: a programme for long-term research into GRID systems architecture
http://www.escience.cam.ac.uk/projects/futuregrid/

[19] Gregory Bell, "Failure to Thrive: QoS and the Culture of Operational Networking", RIPQoS, ACM SIGCOMM 2003, Germany.