A Framework for Developing Reflective and Dynamic P2P Networks (RaDP2P)

Daniel Hughes, Geoff Coulson, Ian Warren
Computing Department, Lancaster University, Lancaster, UK
d.r.hughes@lancs.ac.uk | geoff@comp.lancs.ac.uk | iw@comp.lancs.ac.uk

Abstract

There has been a significant body of research conducted into various structured and unstructured overlay network protocols. Both paradigms have advantages for specific application domains and researchers are beginning to examine the benefits of using hybridized systems. We hypothesize that resource awareness and adaptation are essential to the efficient exploitation of the resources available on the diverse nodes which compose peer-to-peer networks. To support this, we propose a hybrid peer-to-peer model which uses an unstructured decentralised network layered on top of a structured overlay to provide support for multiple levels of adaptation.

1. Introduction

In typical peer-to-peer (P2P) environments, nodes have diverse capabilities and requirements. Furthermore, the resources made available to the network from each node will change over time. Adaptation can maximize the contribution that peers make to the network by changing their role. The benefit nodes accrue from the network can also be increased by adapting network services to better suite their needs.

The Reflective and Dynamic P2P Framework (RaDP2P) provides support for resource awareness in both a static and dynamic context. Static resource awareness represents the capabilities of a node, while dynamic resource awareness represents the resources this node is currently making available to the network.

RaDP2P is a combination of two models. A structured decentralised network provides an efficient routing substrate upon which an unstructured resource discovery layer is overlaid. Research such as Structella [1] has shown the potential of using such schemes; however, we suggest that the inherent structure of the routing layer could be further exploited to support a range of novel adaptation techniques:

- **Network restructuring adaptation:** Based on meta-information about a node; its position in the structured routing overlay may be modified.
- **Routing behaviour adaptation:** Based upon a node's state, or the state of its neighbours, a node may adapt its routing behaviour.
- **Peer selection adaptation:** Following resource discovery, multiple peers may be found providing the desired service. Meta-data about each node can used to inform better peer-selection.

2. RaDP2P

The base layer of our system is formed by a Key Based Routing (KBR) protocol such as Pastry [2] or CAN [3]. These allow for efficient routing of messages and look-up of nodes. Resource discovery services are provided using unstructured decentralised networks such as Gnutella [4] or AGnuS [5]. These are overlaid on top of the base routing layer.

2.1 Supporting Adaptation

Key allocation in RaDP2P differs from most structured overlays, in that key value is used to reflect information about each node. This information is used for network restructuring and routing behaviour adaptation.

Network restructuring adaptation is accomplished using a globally defined network structure policy together with meta-information harvested from each node to generate the most significant bits of each node's key. The KBR layer is ordered by key-value, and the most significant bits of the key are derived from meta-information, hence the network structuring policy defines each nodes relative position.

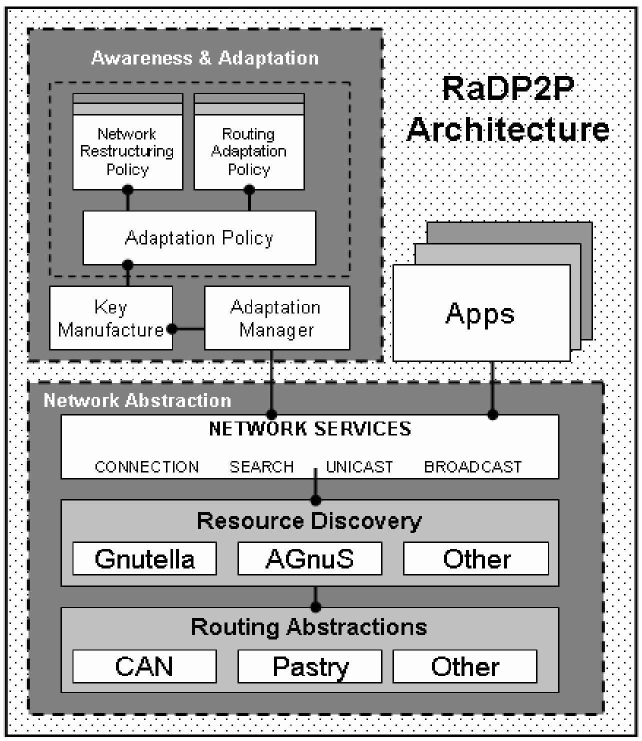
Routing behaviour adaptation is accomplished using a globally defined routing policy together with meta-information to generate the least significant bits of each node's key. In this case, the goal is not to modify the relative position of the node on the network, but simply to mark nodes for differential treatment by their peers.

Peer selection adaptation will occur via the exchange of meta-data and requirements between peers following the resource discovery phase.

As the state of each node changes over time, the nodes' key will be remanufactured and it will reconnect to the most appropriate area of the network tagged for specific routing treatment. In this way, the structure of the network and the routing behaviour of nodes is dynamically maintained.

A complete policy component must define a *network restructuring policy*, a *routing adaptation policy* and a general *adaptation policy* containing the supporting meta-data harvesting methods.

2.2. Implementation



[Figure 1 – RaDP2P Model]

The Framework's core modules are written in Java and policy components are be defined by implementing a Java interface. Policies are loaded at run time from a policy directory using the Java reflection API.

It was considered important that RaDP2P be able to use different resource-discovery and routing substrates. This serves two purposes; it allows the developer to select the most appropriate substrate for any given environment and makes it easy to perform performance evaluation on the different substrates which may be used to underpin a RaDP2P network.

Applications interact with the system through the API of the network services layer, which abstracts over the specific complexities of the underlying peer-to-peer substrates and provides a simple set of generic functions. We hope that this will allow the rapid development of both novel applications and adaptation strategies. The API is still under heavy development, however, it currently provides for the following common peer-to-peer services; connection, point-to-point messaging, broadcast messaging and plain-text search.

Alongside the core system model described in Figure 1, RaDP2P provides supporting utility components including standard meta-information harvesting tools such as CPU benchmarking, network bandwidth measurement and storage media performance testing. All components extend the RaDP2PComponent class which provides common system-wide functionality.

3. Future Work

The RaDP2P framework is currently at an early prototype stage. Currently, the main focus of our development work is the substrate abstraction layer. This layer currently supports only the Pastry [2] routing substrate, though we are working towards support of CAN [3] and Lancaster University's peer-to-peer application framework [6].

We anticipate that a full release of RaDP2P will be made available by the third quarter of 2004. Further details are available at Lancaster's P2P site:

http://polo.lancs.ac.uk/p2p/

4. References

[1] M. Castro, M. Costa, A. Rowstron. "Should we build Gnutella on a structured overlay?" - 2nd Workshop on Hot Topics in Networks. Cambridge, MA USA. November 20-21, 2003

[2] A. Rowstron, P. Druschel. "Pastry: Scalable, Decentralised Object Location and Routing for LargeScale Peer-to-Peer Systems" – Conference on Distributed Systems Platforms, Heidelberg, Germany 2001.

[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker. "A Scalable Content Addressable Network" - Proceedings of ACM SIGCOMM, 2001.

[4] Gnutella Community. Gnutella Protocol Specification v0.4. dss.clip2.com/GnutellaProtocol04.pdf.

[5] D. Hughes, I. Warren, G. Coulson. "Improving QoS for Peer-to-Peer Applications through Adaptation."- 10th International Workshop on Future Trends in Distributed Computing Systems. Suzhou, China. May 26-28, 2004.

[6] Walkerdine, J., Melville, I., Sommerville, I., A Framework for P2P Application Development, Technical Report COMP-004-2004, Computing Department, Lancaster University, 2004.