Exploiting Diversity in Peer-to-Peer Systems

Daniel Hughes
Lancaster University,
Computing Department,
Lancaster. LA2 4WA
+44 (0)1524 510351
d.r.hughes@lancs.ac.uk

Geoff Coulson
Lancaster University,
Computing Department,
Lancaster. LA2 4WA
+44 (0)1524 510306
geoff@comp.lancs.ac.uk

Ian Warren
Auckland University,
Computing Department,
Auckland. 92019
+64 9 37 37599
Ian-w@cs.auckland.ac.nz

ABSTRACT
The capabilities of the nodes which compose peer-to-peer networks vary significantly in terms of connection speed and local resources. In such an environment, it is essential that peer-to-peer systems efficiently exploit the resources available on strong nodes, while at the same time allowing weaker nodes to participate in the network. To accomplish this, it is necessary to be aware of the resources available to nodes and to adapt the role that each node plays in the system. This paper gives a brief overview of RaDP2P, a framework for developing adaptive peer-to-peer systems. RaDP2P uses a hybrid peer-to-peer model similar to Structella as a novel mechanism for supporting resource awareness and adaptation.

Keywords
Diversity; P2P; Adaptation

1. INTRODUCTION
Peer-to-Peer (P2P) applications use resources available on nodes around the edge of the network to provide services; for example Napster [1] uses the disk space of home PCs to provide a large library of music files, while Seti@Home [2] uses the CPU power of home PCs to process extraterrestrial radio signals.

The nodes which compose P2P networks are highly heterogeneous, ranging from mobile nodes with restricted local resources and low-bandwidth, unreliable connections to powerful workstations connected to the Internet by fixed, high-speed links. Where networks are composed of such diverse nodes, it is essential that systems allow participation for 'weak' nodes, while efficiently exploiting the resources available on 'strong' nodes.

Just as the requirements of the nodes which compose a P2P network are diverse, so are the requirements of the applications which run on these networks. Popular P2P applications such as file sharing [1], Internet telephony [3] and distributed computation [2] each have specific and different requirements of the underlying P2P network.

By adapting the role that each node plays in the system to better exploit its capabilities, it is possible to maximize the contribution that each node makes to the system as a whole. Similarly, by adapting the way that each node is treated by the system, it is possible to maximize its suitability for supporting different node and application classes. This is a common idea in real world cooperative systems; *"from each according to their abilities, to each according to their needs"* [4].

2. CLASSIFYING ADAPTATION
We classify the adaptation demonstrated in P2P systems into three discrete levels; network restructuring adaptation, routing behavior adaptation and service selection adaptation.

Network restructuring adaptation adapts the relative position of nodes on the network through selective (re)connection. For example: in a peer-to-peer resource sharing network, a node may wish to modify its position in the network so that it is closer to the content it is seeking [4].

Routing behaviour adaptation adapts the routing behaviour of nodes on the network. For example, if the message passing load on a neighbour peer is known to be high, a node may choose to route fewer messages to that peer based upon this information. [5]

Service selection adaptation adapts which service a peer selects following the resource discovery phase. For example, a node may discover several peers offering a service they wish to use. Meta information provided about these peers may be used to inform the decision about which service to select. [1]

We argue that current P2P systems are not adaptive *enough*. Most systems do not support adaptation and those that do are limited in the scope of adaptation they allow; typically engaging in only one class of adaptation (network restructuring, routing behaviour or service selection adaptation) and in any case the policy used to inform this adaptation is fixed. The resource awareness that existing systems offer is not extensible; restricting the factors that can be used to inform adaptation.

We argue that a generic framework for building adaptive P2P systems is required and that it must provide:

- Support for each class of adaptation.
- Support for multiple adaptation policies.
- Extensible support for resource awareness.

3. SUPPORTING ADAPTATION
We use a hybrid network architecture as a novel and powerful mechanism for supporting adaptation. We use a distributed hash table (DHT) similar to Pastry [6] for message routing, overlaid by an unstructured decentralised network similar to Gnutella [7] in order to support complex queries. The benefits of this kind of architecture have been shown by Structella [8], however, unlike Structella, we use the underlying structure of the KBR layer as a powerful mechanism to support 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 then used for network restructuring and routing behaviour adaptation.

Network restructuring adaptation is accomplished using a globally defined network structure policy together with a resource-awareness policy, which harvests meta-information from each node to generate the most significant bits of each node's key. The KBR layer is structured 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 in a very fine-grained manner. Applications include structuring by content and incentive schemes.

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. Applications include content based routing and load balancing.

Peer selection adaptation will occur via the exchange of meta-data and requirements between peers following the resource discovery phase and is therefore outside of the scope of the RaDP2P network architecture; though this adaptation may be informed by the same extensible resource awareness components as the two levels previously discussed.

4. SYSTEM ARCHITECTURE

The RaDP2P architecture is separated into three primary concerns; awareness and adaptation, network abstraction and applications as shown in Figure 1.

The awareness and adaptation sub-system is responsible for the adaptation behaviour of each node, which is defined by a global adaptation policy and informed by extensible monitoring components.

A Network restructuring policy defines a monitoring component (which harvests meta-data used to form the most significant bits of the key), the interval at which adaptation should be performed, and how this meta-data should be used in key manufacture. In this way, a network restructuring policy defines the desired structure of the network and the level of dynamicity.

A Routing adaptation policy defines a monitoring component (which harvests meta-data used to form the least significant bits of the key), the action to be taken, (e.g. lower the volume of messages being routed on this connection) and an interval of adaptation. In this way, each node remains tagged for the most appropriate routing treatment.

Any factor which can be monitored may be used to inform adaptation at each level, providing flexible, extensible support for adaptation and resource awareness.

Applications interact with the system through the network services abstraction, which provides a high-level interface to the underlying network, supplying common functionality such as connection, search, broadcast and point-to-point message delivery, thus abstracting over the specific resource discovery and routing abstraction used.

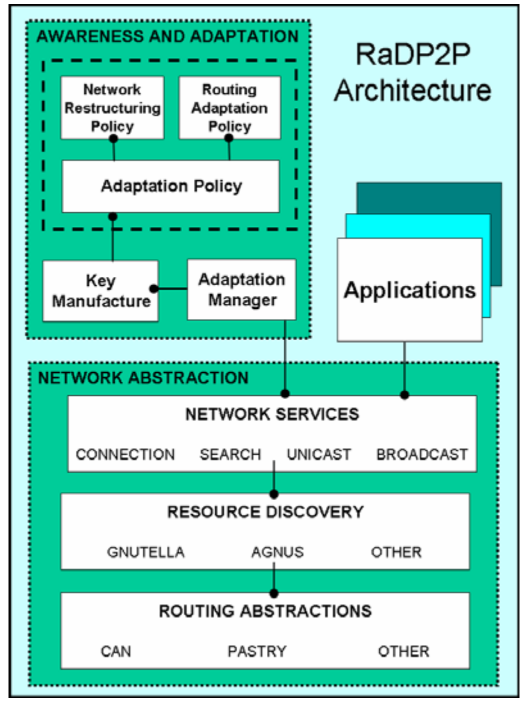

Figure 1 – The RaDP2P Architecture

RaDP2P is implemented in Java and all policy files and resource awareness components are implemented from supplied Java interfaces.

By developing appropriate policies, it is possible for developers to define adaptation at any level, informed by any resource awareness factor and at a specified level of dynamicity. We hope that this will encourage the development of both novel adaptation strategies and novel applications.

6. ACKNOWLEDGMENTS

The authors would like to thank James Walkerdine and Jo Mackie for their comments and suggestions on this work.

7. REFERENCES

[1] Napster, www.napster.com

[2] Seti@Home, www.setiathome.com

[3] Skype, www.skype.com

[4] A Critique of the Gotha Program, Karl Marx, 1874.

[5] "AGnuS: The Altruistic Gnutella Server" Hughes D., Warren I., Coulson G., published in the proceedings of the 3rd IEEE International Conference on Peer-to-Peer computing (P2P'03). Linköping, Sweden, September, 2003.

[6] "*Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems.*", Rowstron A, Druschel P. available at http://research.microsoft.com/ antr/PAST/, 2001.

[7] Gnutella, www.gnutella..com

[8] "Should we build Gnutella on a Structured Overlay", Castro M, Costa M, Rowstron A. published in the proceedings of the 2nd Workshop on Hot Topics in Networks (HotNets-II). Cambridge, MA USA, November 2003.