QoS Assurance on Decentralized Peer-to-Peer Networks

Daniel Hughes, Ian Warren and Geoff Coulson
Computing Department
Lancaster University
Lancaster, LA1 4YR, UK

Email: hughesdr | iw | geoff@comp.lancs.ac.uk

Abstract
*This paper gives an overview of The **A**ltruistic **GNU**tella **S**erver Project (**AGNUS**). AGNUS is a Gnutella host that benefits both its primary user and the Gnutella community as a whole. AGNUS improves Quality of Service (QoS) across the Gnutella network by layering content based routing, caching and load balancing on top of the core Gnutella file sharing protocol. The QoS that can be ensured on Gnutella is limited and we therefore propose a more generic QoS model that provides for resource awareness, QoS reasoning, and adaptive behaviour.*

Key words: Quality Of Service, Content Based Routing, Gnutella, Load Balancing, Reflection

1. Issues with Decentralized P2P Networks

Like most decentralized peer-to-peer networks, Gnutella [1] suffers from performance issues when compared to the centralized alternatives. These performance issues include:

Scalability
The underlying network topology associated with decentralized peer-to-peer systems such as Gnutella, Freenet [2] and ALPINE [3] has an inherent limit to its scalability [4]. These systems sidestep scalability issues by segmenting the network using a time-to-live (TTL) value, which limits how far messages will propagate through the network. While this ensures scalability, it significantly reduces the volume of resources available to each user.

Resource Distribution
One of the key issues faced by decentralized networks is efficient dissemination of popular material within the community. For example, while there are always a large number of files available on the Gnutella network, a small portion of these attract by far the greatest number of downloads (flash crowds), often making it difficult for users to obtain a download slot for popular files. Such files are typically introduced onto the network via a few discreet servers and slowly propagate through the network as users locate, download and share these files. Demand for popular material on these systems often exceeds the capacity of the community to supply it.

Host Participation
Most users do not share files on networks such as Gnutella either due to the security worries associated with running a server on their computer or (as in the majority of cases) because of the performance penalty that it imposes on their local system, particularly where bandwidth is limited. Free Riding on Gnutella [5] discovered that 70% of Gnutella users share no files at all and 90% of all files are served by just 1% of hosts. The resulting network configuration is much closer to a client-server rather than a peer-to-peer model. This introduces points of failure into the network and makes large servers legitimate targets for legal recourse.

2. Project AGNUS

AGNUS was developed to address some of the shortcomings of decentralized peer-to-peer networks as typified by Gnutella. AGNUS is a specialised Gnutella host, which attempts to improve the QoS users experience across the whole community.

2.1 Addressing Scalability Issues

The underlying structure of the Gnutella network is a Cayley tree [6]. Cayley trees have an inherent limit to their scalability imposed by the bandwidth available on the network infrastructure [7]. The way that Gnutella segments the network to circumvent this problem means that users typically have a search horizon of 10,000 nodes – which is a small subset of all Gnutella nodes.

Segmentation reduces the volume of resources available to hosts. AGNUS counteracts this by introducing intelligent routing of requests. While messages still only reach a limited number of hosts, AGNUS attempts to ensure that the most productive search horizon is used, increasing the number of search hits users receive.

When interacting with a network of Gnutella hosts, AGNUS implements intelligent routing by continually monitoring the density of different file types across the network and forwarding incoming requests to the area of the network that is likely to produce the greatest volume of results.

For example: Incoming requests for MP3 files would be forwarded to an area of the network where MP3 files are more plentiful.

Evaluation suggests that searches performed on the current Gnutella network using this system tend to produce more results, however, for significant improvements a large proportion of nodes are required to implement intelligent routing functionality.

When interacting with a network of AGNUS nodes finer-grain intelligent routing is possible. Each node queries its immediate peers to discover what kind of files they are sharing and will route incoming requests only to hosts serving the same file type. This effectively creates a series of virtual networks, one for each file type. Using this system a significant increase in the volume of search hits returned has been demonstrated.

Example 1: When a user searches for video clips on a network of AGNUS nodes their query is only routed through hosts sharing video files and therefore reaches 10,000 hosts that may be able to respond.

Example 2: When a user searches for video clips on Gnutella: Their query reaches approximately 10,000 hosts. 7,000 of these share no files at all and of the remaining 3000, one in four are likely to serve video clips. The query reaches just 750 hosts that may be able to respond.

2.2 Addressing Resource Distribution Issues

The current topology of the Gnutella network is dependent upon a relatively small number of servers; typically around 100 high-volume servers within each search horizon. As discussed earlier, this client/server topology reduces fault tolerance and makes such large servers legitimate targets for legal recourse. AGNUS takes a two-pronged approach to addressing resource distribution issues in Gnutella:

1) AGNUS nodes search for, cache and serve popular files; increasing the availability of these files across the network. The introduction of more hosts serving popular files reduces the number of points of failure in the network, increasing fault tolerance and working to restore the peer-to-peer model one associates with Gnutella.

   File popularity is calculated by sampling query messages that each AGNUS host routes. Heuristic analysis is periodically performed on these samples and a list of the most popular search terms is generated and maintained. AGNUS continually locates, downloads and serves these files to the community, increasing their availability.

2) AGNUS employs load-balancing techniques to improve the QoS experienced across the community. AGNUS periodically monitors the volume of traffic that each of its peers is routing and based on this dynamically reconfigures it's routing policy. In particular, search requests may be forwarded to uncongested hosts and files may be served to congested hosts. In this manner, AGNUS provides a levelling effect on the QoS users experience across the whole system.

The graphs below illustrate the levelling effect that AGNUS is able to provide for two hosts under differing traffic load on the Gnutella network:

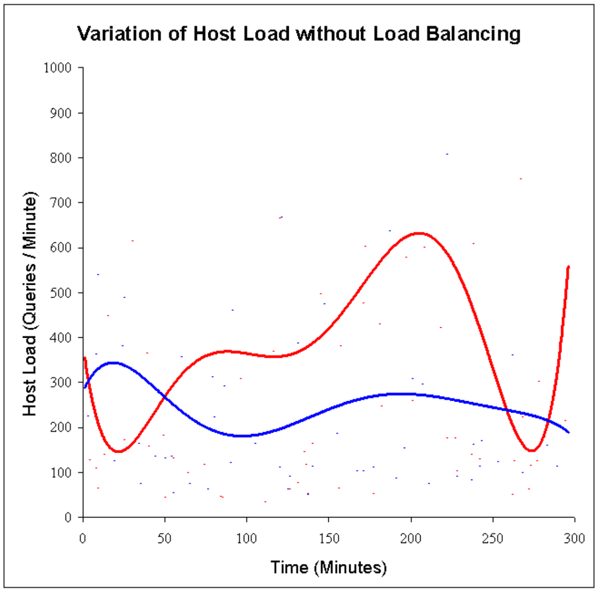

Figure 1

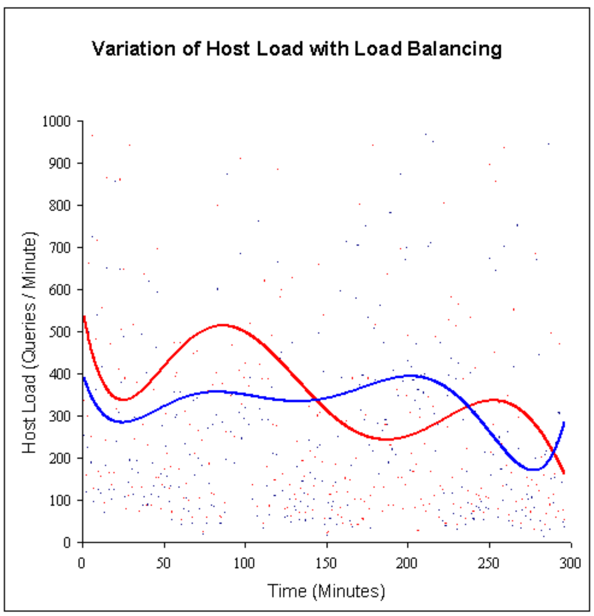

Figure 2

Figure 1 shows the load variation on two Gnutella hosts connected to an AGNUS node and the Gnutella network for a period of 5 hours with load balancing deactivated. Figure 2 shows the load variation on the same two hosts over 5 hours with the load balancing system activated. It can be seen that variation in host load is significantly reduced when the load balancing system is active. This levelling effect spreads load more evenly across Gnutella nodes, leading to more predictable traffic across the network and improving QoS.

2.3 Introducing Motivation to Share

As Free Riding on Gnutella [5] illustrates, the majority of users do not share files on Gnutella. There are several possible explanations for this; including security fears and concern over the legal implications of sharing copyrighted media. However, the most compelling explanation is the tragedy of the digital commons [8].

Gnutella suffers from this phenomenon since users incur no penalty for each file they download nor do they receive rewards for each file that they upload. It is thus rational for individuals to maximise the benefit they draw from their limited resources and not to contribute to the community. The effect of this behaviour is to degrade network performance as a whole, making every member of the community worse off.

Studies on other peer-to-peer systems such as Seti@Home [9] demonstrate that users do behave altruistically in cases where they do not experience a significant performance penalty from sharing their resources. With AGNUS, the caching system automatically downloads files during periods when the user is not using the host. Automatic caching means that all hosts offer some benefit to the community as a whole. Furthermore, there is direct benefit to AGNUS users since on return to their computer they are presented with popular files that have been cached. To retain control over storing files on a local host, we envisage the use of filtering mechanisms to prevent inappropriate files from being cached.

3. Supporting QoS on Decentralized Networks

If we are to seriously consider implementing general purpose applications on decentralized peer-to-peer networks, there are two QoS issues that need to be addressed:

1) The poor levels of QoS exhibited by many existing peer-to-peer applications. While our initial work with AGNUS aims to improve QoS, we advocate the use of a more generic model for tackling this problem.

2) The lack of QoS assurances that can be provided by existing peer-to-peer systems. Currently,

nodes cannot offer assurances to their peers over the QoS levels they are able to offer. This means that peer-to-peer systems are inappropriate for hosting certain classes of application, such as streaming media. Furthermore, nodes are prevented from making intelligent decisions about interacting with their peers.

General measures of QoS include reliability and availability. For distributed computation, reliability might be the likelihood of receiving a correctly processed work-unit and availability could be the probability that the node will remain connected to the network long enough to process the work-unit. In the case of file sharing, reliability might be the likelihood of receiving a desired file by routing a request through a given peer. While notions of QoS tend to be application specific, reasoning about QoS can be performed where knowledge of basic resources is made available.

We advocate a departure from the purists' peer-to-peer model which suggests that all nodes are equal (as in Pastry [11] and Freenet [2] for example). In practice, individual nodes differ in terms of their resources, and consequently in terms of the QoS they can offer. For example, a typical peer-to-peer system may comprise PCs with low bandwidth connections in addition to a range of high performance servers connected to the network using high-speed links. Resources that affect QoS include memory, processing capacity, storage capacity and bandwidth. We propose to make resource information explicit so nodes can assess the QoS a particular peer is likely to offer.

By allowing nodes to differentiate their peers in terms of QoS, they will be able to adapt to the capabilities of their peers. For example, a particular node may require a level of QoS that cannot be met by its available peers. In this case, the node might negotiate or change its behavior to suit the level of QoS available. In addition, where there are several candidate peers that a node may interact with, the node may use resource information to select the most appropriate peer.

To illustrate *adaptive* behavior, consider distributed computation. To successfully process a work-unit, a node, in response to a computation request, may need to assure its peer that it has sufficient disk space to handle the computation. Where the required amount of disk space exceeds the node's capacity, the requesting peer could adapt by partitioning the work-unit and requesting that the processing node take part of the original computation that it can now accommodate. For *discretional* behavior, consider a file-sharing scenario where several hosts are known to offer a particular file. In this case, a peer could use information about bandwidth to decide which node to download the file from.

Our initial work with AGNUS has led to augmented Gnutella hosts which adapt their behavior based on traffic content and flow information. In particular AGNUS hosts change the way they route requests to effect load balancing and to improve the quality of file search results. We propose to develop a more generic and comprehensive model which provides for *resource awareness*, *quality of service reasoning*, and *behavioral adaptation*. To develop this model, we are investigating the use of reflection.

"Reflection is the capability of a system to reason about and act upon itself. A reflective system contains a representation of its own behaviour, amenable to examination and change, and which is causally connected to the behaviour it describes. Causally connected means that changes made to the system's self representation are immediately reflected in its actual state and behaviour, and vice versa". [10]

We envisage the use of reflective mechanisms that allow resource information to be discovered and reflected out to interested nodes at runtime. Nodes can use this information to assess the QoS their peers are likely to offer and to adapt their behavior when required. To realize the model, we expect to develop a layer of middleware that manages resource information and provides appropriate abstractions to applications. In this way, applications have the opportunity to perform application specific processing based on resource information acquired by the underlying middleware.

In addition to the use of reflective mechanisms for providing resource awareness, we expect that the middleware will enable a node to specify the proportion of its resources it is prepared to offer to others. This gives individual nodes control over their own resources, for example by specifying the proportion of bandwidth available to peers. This is in contrast to other peer-to-peer systems such as Freenet where nodes have no control over how their resources are used. While other systems such as Gnutella give individual nodes more freedom (a node can choose not to share files for example), it is not possible for nodes to discover information about their peers.

While resource information is useful in contributing to QoS judgments and for supporting behavioral adaptation, additional information is required to increase the effectives of decision-making. For many hosts, their availability is likely to fluctuate. This is particularly true of home PC users who may connect and disconnect regularly. We propose the use of *operational profiles* for hosts which will reveal connectivity patterns and other run-time information, such as measures of actual performance. We view this dynamic information as complementary to the static resource information described earlier.

4. Conclusions

In this paper we have reported on the AGNUS project which has significantly improved QoS for Gnutella. In particular, AGNUS hosts improve Gnutella hosts with:

- *Intelligent routing*. This increases the number of search hits typically received by search queries.
- *Load balancing*. The introduction of load balancing reduces the fluctuations in traffic that peer nodes experience.
- *File caching*. File caching increases resource distribution across the network and reduces the problems caused by 'flash crowds'.

Our initial work with AGNUS has led us to begin work on a more generic and comprehensive QoS model for peer-to-peer systems. Fundamental to our position is a departure from pure peer-to-peer thinking where all nodes are considered equal. We advocate a model which differentiates nodes with the aim of enabling communities of nodes to work together more effectively and respecting their QoS constraints.

We aim to develop a model which offers the following capabilities:

- *Resource awareness*. Using reflection, a node's peers can discover it's available resources.
- *QoS reasoning*. Based on acquired resource information, a node can assess the QoS that it is likely to receive from its peers.
- *Adaptive behaviour*. A node can adapt its own behaviour based on resource information and QoS judgements.
- *Capture and discovery of operational profiles*. To complement static resource information, operational profiles capture run-time information, such as periods of connectivity. This information contributes to QoS assessment.

5. References

[1] The Gnutella Protocol Specification: http://dss.clip2.com/GnutellaProtocol04.pdf

[2] I. Clarke, O. Sandberg, B. Wiley, T.W. Hong; Freenet: A distributed anonymous information storage and retrieval system": Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, pp. 311—320, 2000.

[3] The ALPINE home page: http://cubicmetercrystal.com/alpine/

[4] J. Ritter; Why Gnutella can't scale – no really: http://www.darkridge.com/~jpr5/, 2002.

[5] B. Huberman, A. Adar; Free Riding on Gnutella: Technical report, Xerox PARC, 2000.

[6] E.M. Rains, N.J. Sloane; Cayley's Enumeration of Alkanes: *Journal of Integer Sequences,* 1999.

[7] N. Gunther; Hypernets, Good (G)News for Gnutella: Technical report, Performace Dynamics Consulting, 2002.

[8] G. Hardin; The Tragedy of the Commons: Science volume 162, pp. 1243-1248, 1968.

[9] Seti@Home Home Page: ttp://setiathome.ssl.berkeley.edu/

[10] G. Coulson: What is Reflection? http://dsonline.computer.org/middleware/

[11] A. Rowstron, P. Druschel: "Pastry: Scalable, distributed object location and routing for p2p systems". IFIP Distributed Systems Conference, Heidelberg, Germany pp. 329- 350, 2001.

6. Acknowledgements

We would like to thank James Walkerdine for his comments and suggestions on this work.