Dependability Properties of P2P Architectures

James Walkerdine, Lee Melville, Ian Sommerville
Computing Department, Lancaster University, Lancaster, UK
{l.melville, walkerdi, is} @comp.lancs.ac.uk

1. Introduction

A system's dependability can be thought of as being the trustworthiness of the system. The difficulty when attempting to measure dependability is that it is typically a context sensitive property. While one user might regard a system to be dependable for the particular activities they use it for, another user might regard it to be totally undependable for their activities.

As well as being context sensitive, dependability is also regarded as being multi-dimensional, in that it can be influenced by a variety of other architectural properties. These properties include security, reliability, availability and performance.

Because of this influence it is difficult to consider dependability without also considering these additional properties. Furthermore, these properties can in turn be affected by other sets of properties. Consequently, a system's dependability can often be influenced by factors that would not typically be regarded as dependability properties in their own right. Because of this network of property influences within a P2P system, it is necessary to identify and consider all properties that can exist within P2P architectures, and to relate these to the dependability of the system.

Identifying dependability properties and achieving dependability within P2P systems is further complicated by the fact that numerous P2P logical architectures exist and no single architecture is likely to be suitable for all application types. For example, applications such as Napster [1] ultimately benefit most from a semi-centralised architecture, where as FreeNet [2] is most suitable to be run over a decentralised architecture.

The type of architecture used can influence the dependability properties of the system. Take for example, security. Fully decentralised P2P systems are likely to be better suited at handling denial of service attacks, semi-centralised P2P systems would be better suited for handling peer certification.

This paper aims to identify the main dependability properties (and related properties) that can play a part within P2P systems. This, in turn, can be used to help inform the creation of more dependable systems. Given the influence the choice of architecture can have, this paper first provides an overview of the main P2P architectures, before going on to identifying the different properties. Future work will provide a detailed analysis of the effect the architectures can have on these properties.

2. Overview of P2P Architectures

Peer-to-peer systems are built up around a collection of peers that are networked together in some fashion. These peers are typically personal computers but there are no reasons why they cannot be anything with a 'digital heartbeat'.

From examining existing peer-to-peer systems it is apparent that two core types of architecture exist. *Decentralised*, where each peer within the architecture is regarded as an equal and no control nodes exists, and *Semi-centralised*, where there exists at least one control peer that performs an authoritative role within the network. Figure 1

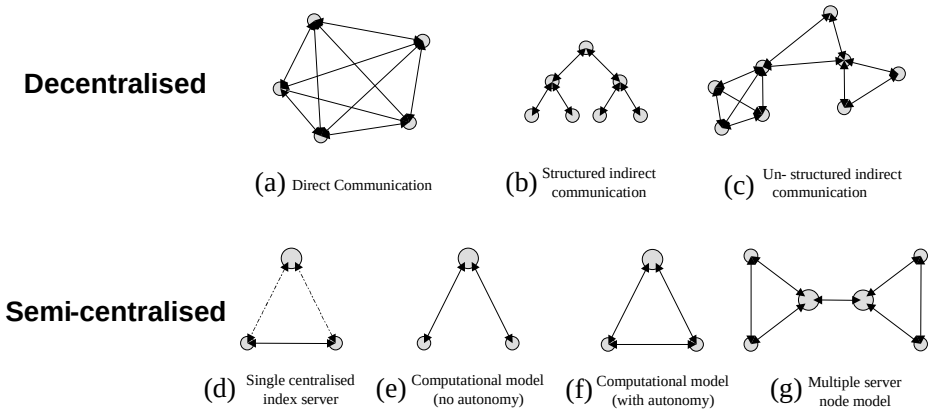

Figure 1 - P2P Architectures

illustrates seven possible P2P logical architectures.

3 Properties of P2P Architectures

This section identifies the main properties that can exist within P2P systems. The properties have been split them into two categories.

Architectural properties – properties that can be specifically affected by the type of architecture

Emergent properties – properties that emerge over time as the architecture is used.

3.1 Architectural properties

Reliability – the perceived reliability of a system.
Scalability – the ability of a system to operate without a noticeable drop in performance despite increases or decreases in its overall operational size.
Security – the level of security within a system represents its ability to protect itself against accidental or deliberate intrusion [5].
Survivability – the capability of a system to fulfil its mission in a timely manner in the presence of attacks, failures, or accidents [6].
Safety – a systems ability to operate without catastrophic failure [3].
Maintainability – the ease in which the system can be changed after it has been delivered and is in use [3].
Responsiveness – not only includes latency, jitter and other system performance attributes, but also how the end user perceives this performance and to what use the system is being employed (i.e. real time constraints).
Responsibility, Accountability and Reputation [4] – the enforcing of rules for social responsibility within a system
Availability – the probability that a system, at a point in time, will be operational and able to deliver the requested services [3].
Fault Tolerance – the ability for a system to continue giving a correct service following the manifestation of a fault or faults either through errors in the system design, implementation, or introduced following an attack [3].
Political and legal independence – how easy it is to forcibly shut a system down [4].
Data integrity – maintaining the integrity of the data that is stored and manipulated by a system.
Connection bandwidth – the varying amounts of connection bandwidth, peers within a system can possess.
Intermittent node connectivity – the dynamic connectivity of peers within a system.
Peer Discovery – a system's mechanisms used for discovering other peers on the network.
Anonymity – the ability of a system to hide a user's identity, or to keep stored data in an anonymous state [4].
Peer Addressing – the peer addressing mechanism used within a system.
Load Balancing – the load that is place on peers within a system is balanced to ensure that a component is not overworked or underused.
Manageability – the ease in which the system as a whole can be managed.
Adaptability - the systems ability to adapt to a dynamically changing environment

3.2 Emergent properties

Person centric addressing – using an addressing mechanism that is based on the users of the system, rather than on the physical peers.
Network evolution – P2P architectures have been known to automatically adapt over time due to their environment
Legacy versions – the ability of a system to still function despite different versions of the application operating on peers within the system.
Trust – a highly subjective property that represents how much a user trusts a system.

4. Summary and Conclusions

This paper has attempted to identify the properties that can have an influence on a P2P systems dependability.

Dependability is a difficult attribute to measure. Not only is it context sensitive but a range of interconnected properties can also influence it. Achieving dependability within P2P systems is further complicated by the fact that numerous underlying network architectures can be used, and these are likely to have an impact on the dependability properties.

This paper has provided an overview of the key P2P network architectures that are used, before going on to identify properties that can influence a P2P system's dependability.

Currently only an initial analysis has been performed in determining the affects the different architectures can have on the dependability properties. This has shown that the different architectures can provide advantages and disadvantages. Direct communication architectures might be the most responsive, but do not scale well. Semi-centralised allow for a more managed system, but suffer from single points of failure.

It is our intention to extend the work by assessing any possible affects in more detail. The results of this work will be presented in a future paper.

5. References

[1] Napster. More info at the URL http://www.napster.com

[2] FreeNet. More info at the URL http://freenet.sourceforge.net

[3] Ian Sommerville, *Software Engineering*, Addison Wesley, 2001

[4] Andy Oram (editor), *Peer-to-Peer: Harnessing the power of Disruptive Technologies*, O'Reilly publishing, 2001

[5] Ian Sommerville, *The DISCOS Method presentation*, Lancaster University

[6] Ellison, R. J., Fisher, D. A., Linger, R. C., Lipson, H. F., Longtaff, T. A., Mead, N. R., SURVIVABILITY: Protecting Your Critical Systems, In IEEE Internet Computing,, 3 (6), 55-63, Nov/Dec, 1999.