Cross-Layer Peer-to-Peer Traffic Identification and Optimization Based on Active Networking*

I. Dedinski[1], H. De Meer[1], L. Han[2], L. Mathy[3], D.P. Pezaros[3], J.S. Sventek[2], and X.Y. Zhan[2]

[1] Department of Mathematics and Computer Science,
University of Passau, Passau, Germany, 94032
{dedinski,demeer}@fmi.uni-passau.de
[2] Department of Computing Science,
University of Glasgow, Scotland, UK, G12 8QQ
{lxhan,joe,xyzhan}@dcs.gla.ac.uk
[3] Computing Department, Lancaster University,
Lancaster, UK, LA1 4WA
{laurent,dp}@comp.lancs.ac.uk

Abstract. P2P applications appear to emerge as ultimate killer applications due to their ability to construct highly dynamic overlay topologies with rapidly-varying and unpredictable traffic dynamics, which can constitute a serious challenge even for significantly over-provisioned IP networks. As a result, ISPs are facing new, severe network management problems that are not guaranteed to be addressed by statically deployed network engineering mechanisms. As a first step to a more complete solution to these problems, this paper proposes a P2P measurement, identification and optimisation architecture, designed to cope with the dynamicity and unpredictability of existing, well-known and future, unknown P2P systems. The purpose of this architecture is to provide to the ISPs an effective and scalable approach to control and optimise the traffic produced by P2P applications in their networks. This can be achieved through a combination of different application and network-level programmable techniques, leading to a cross-layer identification and optimisation process. These techniques can be applied using Active Networking platforms, which are able to quickly and easily deploy architectural components on demand. This flexibility of the optimisation architecture is essential to address the rapid development of new P2P protocols and the variation of known protocols.

1 Introduction and Motivation

P2P overlays do not adopt any notions of centralised management nor do they employ the traditional static client/server paradigm. Most of the peers in a P2P network are symmetric and can allow their resources to be shared amongst other peers to deliver a common service [Ora01]. Consequently, within a file sharing P2P overlay every peer

* This work has been supported by the grant EPSRC GR/S69009/01 and EuroNGI NoE.

D. Hutchison et al. (Eds.): IWAN 2005, LNCS 4388, pp. 13–27, 2009.

can simultaneously act as a server and a client, fetching and providing data objects that range from a few megabytes to approximately one gigabyte in size [GDS03]. By exploiting a user-configurable, arbitrary amount of peers' end-systems resources, a P2P overlay can perturb the Internet in new ways by constituting random nodes and portions of the network highly loaded for an unpredictable amount of time. Internet Service Providers (ISP)s can hence experience rapidly varying traffic patterns at non-statically-provisioned portions of their networks, which can adversely impact the network performance experienced by all traffic flows. For example, a recent study that analysed traffic collected from a 1 Gb/s link connecting an ISP backbone to several ADSL areas revealed that at least 49% of the overall traffic is due to P2P applications, as this has been reported by well-known transport port numbers [AG03]. In addition, it has also been recently reported that the proportion of P2P traffic on Tier 1 networks is steady if not increasing the last two years [KBB04]. This, coupled with the dynamicity of P2P traffic, can impact not only the peering relationships among ISPs, but also the volume-based charging imposed by upstream providers. Traditional tools for network management support quite static forms of network and traffic engineering usually based upon offline post-analysis of monitored data and estimated approximation of path, traffic and demand matrices [GR02]. However, the rapidly varying traffic patterns expected by P2P flows are not addressed by such tools, since P2P requests are not guaranteed to be addressed to a few popular servers, as is the case for the client-server environment [SW02]. Rather, the main dynamic of P2P systems is the advertisement of new data objects, which can appear at arbitrary peers [GDS03], and hence operators are in need of more dynamic (real time) mechanisms to provide fine control over the network-wide P2P traffic flow.

A longer-term perspective of P2P dynamics is the constant evolution of P2P protocols and the creation of new P2P applications, which are rapidly spreading over the Internet. The P2P phenomenon is still relatively recent and does not conform to any standards or rules regarding program interfaces, connection behaviour, etc. The mutation of the P2P protocols as well as the appearance of new protocols makes tracking of P2P traffic steadily more complicated and static planning of network resources less successful. The P2P community is averse to ISP control of any kind and invents protocols that attempt to prohibit and avoid traffic identification, shaping and blocking. An ISP, therefore, needs to track actual P2P development and to adapt on new techniques and protocols quickly.

The *store-compute-and-forward* model of operation facilitated by network node programmability is particularly suitable for such dynamic, non-standardised and highly unpredictable systems. The additional intelligence and control that is integrated with the network's main forwarding operation can be exploited to provide for dynamic identification of P2P traffic, and consequently for network performance optimisation in the onset of P2P activity. Such traffic control enforcement can also employ application-aware programmable mechanisms that do not simply shape and block P2P traffic, but favour well-behaved P2P systems and optimise the overall resource utilisation of the network. A comparative study of different programmable network architectures can be found in [CMK99]

This paper focuses on the investigation and deployment of a synergistic network-layer and application-aware programmable framework aimed at measuring, managing, and optimising the performance in networks that support P2P applications. Section 2

discusses existing P2P identification and optimisation approaches and raises its limitations. Section 3 describes the architectural properties of an always-on programmable system that exploits network and application-level knowledge to synergistically detect and the onset of P2P activity and employ traffic optimisation algorithms over both the overlay and the physical network topologies. Preliminary analysis has initially focused on the network-level identification of P2P flows based on their internal traffic behaviour, and comparison between the performance characteristics of P2P and non-P2P flows are presented in section 4. In addition, wavelet analysis results that demonstrate the existence of discriminating information within P2P traffic behaviour are presented. Section 5 concludes the paper and outlines future work directions.

2 P2P Traffic Identification and Optimization Challenges

Currently, there are three, widely-used approaches for passive P2P traffic identification: application signatures, transport layer ports and network/transport layer P2P pattern recognition based on heuristics. The application signatures based approach [SSW04] searches for protocol specific patterns inside the packet payloads. The simplicity of this method is obvious, but it also introduces some important problems. First, it cannot be adapted automatically to unknown, recently-introduced P2P protocols. Enhancements to existing protocols, as well as the appearance of new protocols, occur frequently. Second, application-level pattern search in each transport packets creates a higher load compared to other network and transport-layer-based approaches. Finally, some P2P protocols avoid payload inspection by using encryption algorithms. The transport layer port identification [SW02] solves the last two problems. It is easy to use, does not produce too much load at the measurement nodes and does not rely on inspecting application payloads. This method suffers from inability to adapt to modified or recently-introduced protocols. Furthermore, many P2P applications have begun using variable or non-P2P port numbers (HTTP, FTP, etc.) to deliberately avoid port-based identification and allow P2P communication through firewalls. As a result, port-based P2P identification highly underestimates the actual P2P traffic volume [KBB03]. Heuristic based network/transport layer approaches [KBB03, KBF04] use simple network/transport layer patterns, e.g. the simultaneous usage of UDP and TCP ports and the packet size distribution of a P2P flow between two peers. This method gives good performance for existing P2P protocols and can even be used to discover unknown protocols. The problem here, however, is that it is straightforward to construct a new P2P protocol, effectively avoiding the proposed heuristics. Recently, it has been suggested that the observation of host behaviour and the identification of social, functional and application-level patterns therein can lead to accurate traffic classification that obviates the aforementioned concerns [KPF05].

Active P2P traffic identification approaches (active probing) have been used to traverse and gather topological information about different types of P2P networks [Gnu, Tuts04, KaZ]. These approaches use some probing peers called *crawlers* to connect to a desired P2P network. The crawlers then issue search requests and collect the IP addresses of the answering peers. By collecting these addresses, one can

reconstruct the overlay topology of the P2P network. One obvious advantage of constructing such a topology is that subsequent P2P traffic measurement and identification needs to concentrate only on flows coming from or directed to IP addresses collected by the crawler. This improves the identification performance considerably and is an example of how application-aware, active probing can support passive P2P identification approaches. On the other hand, active probing has its limitations. In the eDonkey network, for example, it is only possible to discover the eDonkey superpeers in an efficient way. Identifying eDonkey clients can be done efficiently by using passive P2P identification approaches, which track flows coming from and directed to eDonkey superpeers. Therefore, a combination of application-aware active probing and network-level passive identification techniques is a promising strategy.

As it has already been mentioned, network-layer controlling techniques do not consider preserving or even improving the functionality and performance of the P2P network. This goal can be achieved by using application-layer optimisation approaches [ADD04, Fry99, DeMeer03, GDS03, LHK04, LBB02, THH04], which all rely on P2P traffic redirection, shaping or proxy caching. These approaches work fine for a big number of P2P protocols, which are still widely open (not encrypted and can be reverse-engineered), so the redirection and shaping are possible. Application-aware programmable mechanisms can transparently provide for micro-services such as application-level routing, and application-specific resource discovery and differentiation. An ISP-controlled overlay mesh can be established in this way to manage the associated (P2P) traffic flows, through an understanding of application-specific data transfers without knowing the details of the underlying physical network. However, application-level approaches are strongly dependent on application-specific semantics. A programmable networking infrastructure that enables the deployment of specific application-aware optimiser and identification components for new, recently reverse-engineered P2P protocols is required,

On the other hand, network-layer controlling techniques (shaping and blocking) do not depend on application protocol internals. The combined use of both network-level and application-level optimisation techniques to enforce control mechanisms to optimise the overall network operation opens new promising grounds for research and, at the same time, yields many integration issues. A straightforward example of such synergy is to force P2P clients to use a certain application layer optimisation service provided by the ISP by blocking and shaping non-conforming P2P traffic at the network layer [MTT03].

3 Architectural and Experimental Design

This paper describes an always-on Monitoring Measurement and Control (MMC) architecture for P2P identification and network optimization deployed on programmable nodes at strategic points in the network. The choice of these points depends on numerous factors such as programmable node performance, and network topology and load. Instead of statically specifying strategic points, MMC relies on the dynamic instantiation of ALAN proxylets [Fry99] to allow the on-demand installation and removal of components at different programmable network nodes. Such approach also

allows the fast deployment of application-specific modules for P2P protocols that have been newly reverse-engineered. The ALAN infrastructure operates synergistically with the LARA++ active router framework, essentially offering an additional application-specific programmable layer. LARA++ is a software implementation of a programmable router that augments the functionality of a conventional router/host by exposing a programmable interface, allowing active programs -referred to as active components- to provide network level services on any packet-based network [SFS01]. Figure 1 shows the coarse structure of the proposed architecture. It is divided in three processing planes spread across the network and application layers. These planes try to synergistically address the identification and optimisation challenges presented in section 2. Additionally, a communication module is used to exchange locally collected data among programmable nodes. Its purpose is to enable global identification and optimization of P2P traffic. The communication module can for example be implemented in a centralistic way (programmable nodes exchange information through a central database server). Other possible communication approaches like the construction of a decentralized programmable node overlay structure. This paper does not rely on any particular design of the communication module.

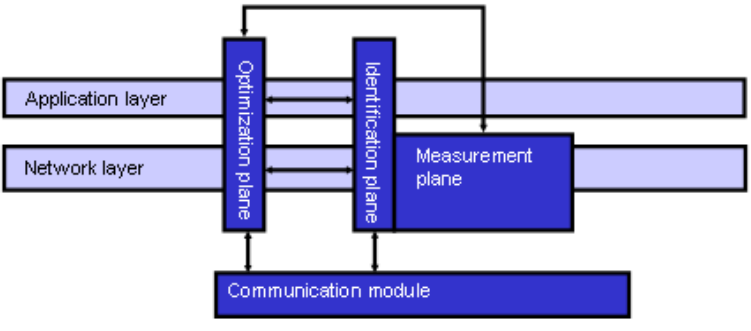
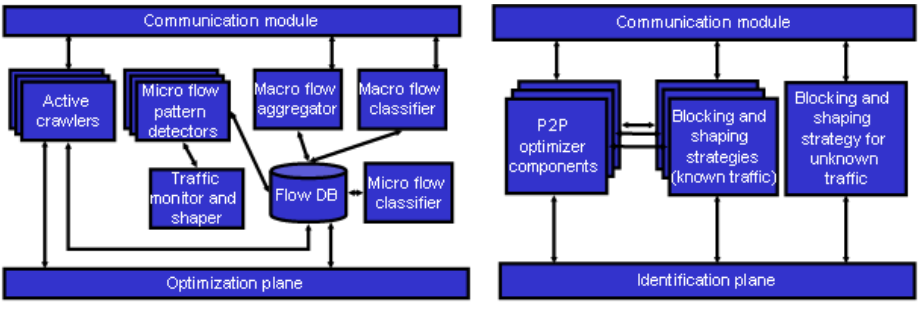


Fig. 1. Architecture – Two Layer Programmability

Measurement and Identification Planes

The measurement plane takes as input the traffic, passing through its network node. It captures and aggregates relevant microflow patterns used for traffic clustering. A microflow can be easily identified at the network layer by a 5-tuple including the source and destination IP addresses, transport protocol and transport layer source and destination port numbers, if not encrypted [CBP95, Cla94]. In contrast to common passive flow measurement systems that only record aggregate flow indicators [Bro97, NFL], the flow-based classification and measurement employed by this architecture needs to keep per-packet state in order to compute performance properties such as packet inter-arrival time and packet size distributions. Such state needs to be captured continuously but at the same time reduced at a minimum by periodically substituting per-packet information with aggregate statistics. Packet timestamps and lengths kept for each active flow are being periodically aggregated by the pattern detection measurement modules to distribution summary statistics. The raw indicators are subsequently

removed from the flow table. Further state reduction through sampling is considered with systematic count-based sampling schemes being appropriate candidates due to the simplicity of the sampling algorithm, but also due to its ability to capture the traffic's burstiness and produce accurate approximation of the parent population for both single and multi-point performance metrics [CPB93, Zse05].



(a) (b)

Fig. 2. (a) Identification-Measurement Plane and (b) Optimization Plane

Figure 2(a) presents a more detailed view on the components of the identification and measurement planes. Plug-in measurement modules that are implemented as ALAN proxylets act as microflow pattern detectors and periodically compute properties of the identified microflows in order to classify them into similarity classes, according to some per-packet and/or inter-packet characteristics. Estimates of the packet size distribution, for example, can be used to distinguish bulk from interactive and signalling TCP flows. Although this can prove more challenging than exploiting simple heuristics, interactive flows' dependence on user-behaviour can be revealed from the periodicity of their time series as well as from their distribution's heavy tails. The main dynamic behaviour of the measurement process lies in the ability of new instances of measurement-based pattern detector ALAN proxylets to be loaded on-demand to compute additional metrics. This also influences the operation of the traffic monitor and shaper which can be dynamically configured to record and deliver additional per-packet information passed to the corresponding microflow pattern detector. The microflow patterns collected at the measurement layer are stored in a flow database. The microflow classifier component, which is located at the identification plane, searches for correlations between microflows passing through this access point node. The microflows are clustered into *similarity classes* according to the patterns collected at the measurement plane. Supervised and unsupervised adaptive techniques for flow classification can be applied to discover similarity classes. A comparative study of classification (clustering) methods is presented in [Zai97]. Unsupervised techniques have the advantage of detecting new unknown traffic classes. The addresses of all source and the destination hosts producing traffic in the same similarity class are collected in the database. With this information, a topology can be constructed, containing all hosts that produce traffic belonging to that similarity class. The micro and

macroflow information (patterns, similarity classes) is stored with some history, which allows the correlation of flows that are not necessarily passing through the instrumented node at the same time. The macroflow classifier uses the topology information to distinguish between P2P like traffic and non-P2P traffic. For example in P2P systems the participating nodes are mostly acting both as client and server. A P2P topology collected by the macroflow aggregator would thus contain incoming and outgoing flows for the most of the nodes. On the other hand, a topology collected for the HTTP protocol would have a two level hierarchical structure, with each node uniquely identifiable as a server or a client. And a DNS topology would have a multi level hierarchical structure. The knowledge about the topology is a powerful traffic identification criterion, which can help to identify even unknown traffic. The macroflow aggregator exports its macroflow knowledge to the other programmable nodes by using the communication module. Respectively, the macroflow classifier uses macroflow information coming from the communication module to construct a local view of a certain traffic topology and to decide whether it is P2P like.

Finally active crawler ALAN proxylets are dynamically loaded at the application layer to traverse and discover the overlay networks of reverse-engineered P2P protocols. The results of the crawlers are stored into the flow database and are compared with the results of the identification components at the network layer in order to improve and verify the performance of the latter.

Optimization Plane

Based on the information collected and produced by the measurement and the identification planes, optimization and manipulation actions regarding identified P2P protocols can be taken at the optimisation plane (Figure 2(b)). The blocking and shaping component for unknown P2P traffic initiates network level actions, without semantic knowledge about a P2P protocol. Such actions can be priority-based routing, complete blocking or bandwidth limits for certain traffic flows (similarity classes). These actions may have regional or global character. The P2P optimizer components do not block or shape P2P traffic, but instead redirect it, thus avoiding network congestions and at the same time improving the P2P network performance. Different application-level optimization techniques are applicable for different P2P protocols, so the application optimizer component has to be adjusted to a predefined set of supported applications. Some of the application level techniques need to install blocking or shaping strategy components for the network layer to prohibit P2P traffic to run around the optimizing entities (P2P caches, proxies).

Experimental Design

A critical aspect of the methodology described above is to determine the network-level characteristics of P2P application traffic of relevance to different microflow pattern detectors. An isolated network tracing environment has been constructed to capture traces of synthetic traffic from a number of P2P applications. An eDonkey-specific setup using this environment is shown in Figure 3 below.

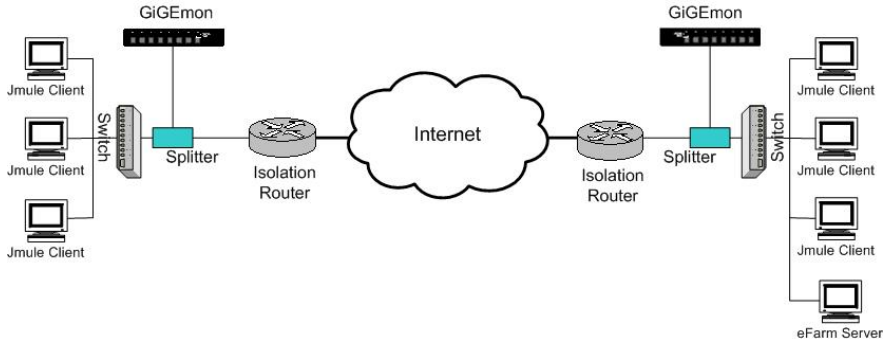


Fig. 3. Experimental Packet Capture Environment

Isolated experimental configurations have been setup in three participating sites (universities of Glasgow, Passau, and Lancaster), and initial tests using the eDonkey protocol have been conducted. No other network traffic was active on the sub-nets behind the isolation routers, although the traced traffic was subject to variable delays due to congestion in the campus intra-networks and the Internet. The analysis discussed below was with respect to a single content-providing peer interacting with a single downloading peer at a remote site. After a short search for the content at the superpeer, the downloading peer initiated the download of a 600 MB file from the providing peer. Full packet traces were recorded at the edges of each isolated configuration by GPS-synchronised GigEMON passive monitoring systems, which are engineered to perform lossless, full-packet capture of traffic in both directions to disk storage [End].

Pattern Detection Methodology

The initial approach to detecting network-level packet patterns is to look for specific temporal behaviours associated with the packets in a P2P micro-flow. For time-dependent processes that are stationary, the traditional approach is to perform a Fourier analysis of the signal, thus converting the large number of experimental data points to a small, bounded number of coefficients for the Fourier basis functions in the Fourier expansion of the signal.

Due to the time varying nature of the Internet, one does not expect the temporal behaviour of a micro-flow to be stationary. Wavelet analysis techniques [Chu92] have been developed to address temporal behaviour that is non-stationary. Wavelet techniques exhibit good time resolution in the high frequency domain (implying good localization in time) as well as good frequency resolution in the low frequency domain [AV98].

For a non-stationary signal, wavelet analysis can determine sharp transitions simultaneously in both frequency and time domains. This property of wavelet analysis makes it possible to detect *hidden* but highly regular traffic patterns in packet traces. The result of wavelet analysis is a small, bounded number of coefficients for scaling and wavelet basis functions.

Initially, the collected eDonkey traces have been subjected to wavelet analysis to understand whether such analysis provides the ability to distinguish eDonkey traffic from non-eDonkey traffic. To that end, analysis results for an FTP session transferring the same file in the experimental environment are provided for comparison with the eDonkey analysis results in the following discussion.

4 Preliminary Analysis and Results

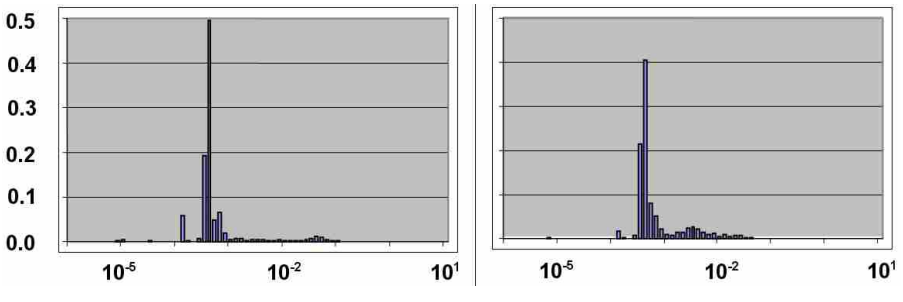
Before discussing the results of wavelet analysis, it is informative to first look at various statistical characterizations of the measured traces. The initial focus has been with respect to packet inter-arrival time and packet size distributions.

Each flow consists of control packets used by the applications to locate and initiate data transfers and data packets that correspond to the actual download of the requested content. The packet patterns for these two, different sub-flows are expected to exhibit significantly different characteristics, since the control/signalling traffic is an RPC-style interaction at the application level, while the data traffic is more characteristic of an asynchronous, reliable flow from the *server* to the *client*. Therefore, signalling and data traffic are considered separately below.

Inter-Arrival Time Distributions

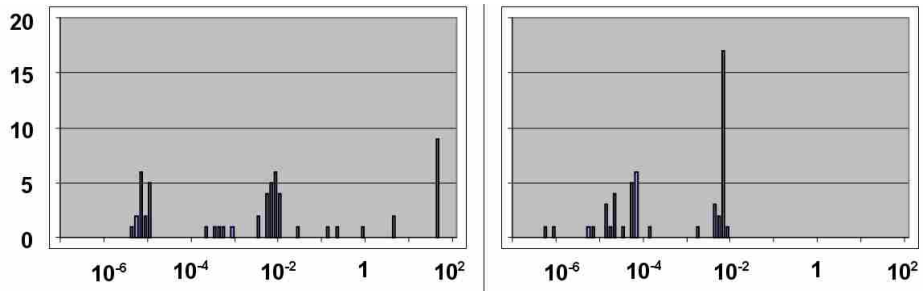
Figures 4 and 5 below show the probability/frequency distribution functions for the inter-arrival time distributions for the data and signalling sub-flows, respectively. Two observations are immediately obvious from these figures:

- The data streams exhibit resonances at the same values of packet inter-arrival time; even though the resonance at 10^{-4} seconds for the p2p data sub-flow is more pronounced than for the ftp flow, it is not sufficiently significant to confidently discriminate between eDonkey and FTP based upon this evidence alone.
- The signalling sub-streams, on the other hand, exhibit significant differences in their inter-arrival time spectra, especially for large inter-arrival times. If these differences persist over different congestion regimes of the intervening networks (to be established experimentally in future work), then it is feasible that high-confidence discrimination can be achieved with appropriate pattern matching filters.



(a) (b)

Fig. 4. Probability distribution as a function of packet inter-arrival time (in seconds) for (a) eDonkey data and (b) ftp data flows



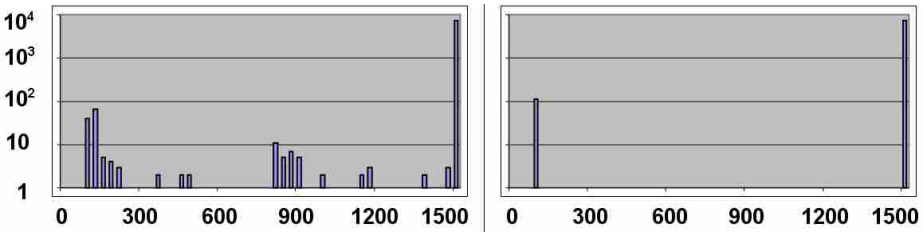
(a) (b)

Fig. 5. Frequency distribution as a function of packet inter-arrival time (in seconds) for (a) eDonkey control and (b) ftp control flows

Packet Size Distributions

Figures 6 and 7 below show the frequency distribution functions for the packet size distributions for the data and signalling sub-flows, respectively. Two observations are immediately obvious from these figures:

- The data streams exhibit strong resonances at essentially the same values of packet size; note that the p2p data sub-flow exhibits a small number of packets interspersed between the two strong resonances; it is not clear whether the presence of these intermediate packet size values is sufficiently significant to confidently discriminate between eDonkey and FTP based upon this evidence alone.
- The signalling sub-streams, on the other hand, exhibit differences in their packet size spectra, especially for large packet sizes. If these differences persist over different congestion regimes of the intervening networks (to be established experimentally in future work), then it is feasible that high-confidence discrimination can be achieved with appropriate pattern matching filters.



(a) (b)

Fig. 6. Frequency distribution as a function of packet size (in bytes) for (a) eDonkey data and (b) ftp data flows

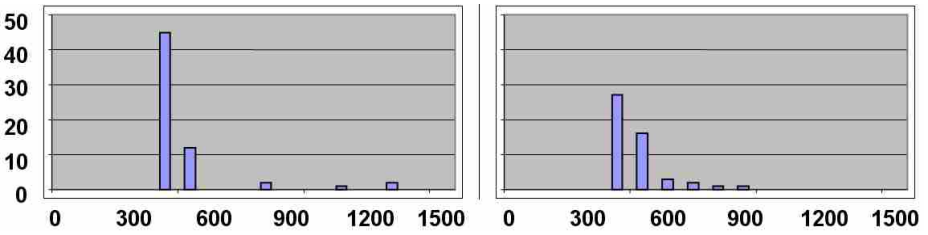


Fig. 7. Frequency distribution as a function of packet size (in bytes) for (a) eDonkey control and (b) ftp control flows

Note that analysis by others has yielded similar insights [Nla]. The results for the signalling sub-stream, if they hold across congestion regimes, augurs well for developing pattern matching filters for detection of control sub-flows when the packets are encrypted, as most encryption schemes are packet-size preserving, modulo padding introduced to make the packet size a multiple of 4 or 8 bytes.

Wavelet Analysis

Despite the fact that the distributions for the p2p and ftp data flows shown in figures 4 and 6 do not show significant differences, scatter plots of these traces with respect to both attributes do show significantly more variation in the p2p trace than in the ftp trace (not shown due to space reasons). This indicates that there is scope for discriminating between such traces. The first attempt at such discrimination has been through the use of wavelet analysis. Only aspects of wavelet analysis that are critical to this application are discussed below; interested readers are urged to consult [DAU92] for more details.

In terms of wavelet theory, a signal $Y^0(t)$ (e.g. bursty traffic) can be represented as:

$$Y^0(t) = Y^J(t) + \sum_{j=1}^{J} detail^j\{Y(t)\}$$
$$= \sum_k a_Y^J(k)\varphi_{J,k}(t) + \sum_{j=1}^{J}\sum_k d_Y^j(k)\psi_{j,k}(t)$$

where k and j denote time and frequency indices, respectively. The $a_Y^J(k)$ are scaling (approximation) coefficients, and the $d_Y^j(k)$ are wavelet (detail) coefficients. A scaling function with low-pass filter properties $\varphi_{J,k}$ is used to capture an approximation signal (low-frequency signal), and a wavelet function $\psi_{j,k}$ with band-pass filter properties is used to extract the detailed information (high-frequency signal).

Wavelet signal analysis consists of three primary phases:

- The **analysis** phase decomposes the data into a hierarchy of component signals by iteration. Starting with a signal *S*, the first step of the transform decomposes *S* into two sets of coefficients, namely approximation coefficients $a_Y^1(k)$ and detail coefficients $d_Y^j(k)$. The input *S* is convolved with the low-pass filter to yield the approximation coefficients. The detail coefficients are obtained by convolving *S* with the band-pass filter. This procedure is followed by down-sampling by a factor of 2, and this process is then applied to the down-sampled signal. At each iteration of this phase, the input is a signal of length N, and the output is a collection of two or more derived signals which are all of length N/2. We obtain the approximation signal at the highest level j and the collection of detail coefficients at each level until the end of decomposition.
- The **signal processing** phase compresses, de-noises and detects the underlying signal by modifying some wavelet coefficient values and then reconstructs the signal using these altered wavelet coefficients.
- The **synthesis** phase is the inverse of the iteration phase.

The wavelet coefficients are key to matching the spike pattern of a signal. By focusing on the Probability Density Function (PDF) of wavelet coefficients, one can determine the algorithm for selecting a suitable threshold and dropping non-significant coefficients when reconstructing the signal.

Tools to perform this analysis on micro-flow, inter-arrival time distributions obtained from the the experimental environment described in section 4 above have been developed and validated. These tools have been applied to the data sub-flow distributions shows in figure 4 above.

Exploration of the information resulting from these analyses is in its beginning stages; the initial focus has been to concentrate on measures of the significance of wavelet coefficients at each decomposition level, denoted by the index *j*. Table 1 below shows the significance for each decomposition level in the analysis of the p2p and ftp data sub-flow traces.

Table 1. Level-specific coefficient significance from wavelet decomposition

	j = 1	**j = 2**	**j = 3**	**j = 4**	**j = 5**	**j = 6**
p2p	-5.3490	-4.5921	-4.0269	-5.0712	6.7115	-1.1379
ftp	-9.6785	-5.3787	4.5109	-4.5486	4.8186	-6.0644

A particularly useful way to visualize the relationships between decompositions for different applications is to plot these values on a radar diagram. Such a diagram is shown in Figure 8 below. From this diagram, it is immediately apparent that the p2p trace is significantly different from the ftp trace at levels 1, 3 and 6. While Figure 4 did not provide sufficient information to discriminate between the two data flows, it is apparent that sufficient discriminating information is contained in the traces.

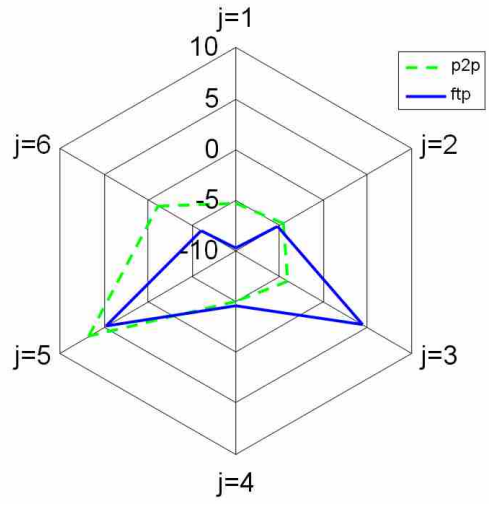


Fig. 8. Wavelet coefficient significance per level for eDonkey and ftp data sub-flows

Real-Time Considerations

A micro-flow detector must detect patterns in real-time to feed the optimization layer in the architecture described in section 4. It is important to estimate the ability of a detector based upon wavelet analysis to function in real-time.

In order to perform a wavelet analysis to level N, a sufficient number of packets must be accumulated in the inter-arrival time histogram for a micro-flow such that 2N bins have nonzero frequency counts. For example, if analysis to level 3 is sufficient to discriminate the flows of interest, then 8 non-zero bins are required. In addition to considerations regarding minimum number of observations, one must also assure that the time duration of the observed portion of the flow is sufficient to mask any startup transients and to yield a distribution that is representative of the entire flow.

As with any predictive technique, the expected accuracy of predictions will improve if more data is available over which to perform the analysis. Given that most p2p sessions are operational for a long period of time, there is a tradeoff to consider between rapidity of prediction versus accuracy of prediction. Reduction of false positives, consistent with *rapid enough* prediction, is the long-term goal.

5 Conclusions and Future Work

This paper has introduced an architecture for exploiting active/programmable networking techniques to manage p2p applications. Crucial to the success of an infrastructure based upon this architecture is the ability to detect onset of p2p activity by passively observing network-level micro-flows. Application-level probing mechanisms can support the network-layer identification process, which can in turn be the basis for application-layer optimisation techniques that improve P2P performance.

The project has constructed an experimental infrastructure that enables the full packet capture of synthetic micro-flow traffic. The traces resulting from this synthetic traffic enables the assessment of a number of p2p pattern detectors for driving such management activities.

The first analysis technique that has been assessed has been based upon the use of wavelets. Preliminary results indicate that these techniques may prove useful for constructing real-time p2p pattern detectors.

Future work will focus on extensive measurement and analysis of further invariant factors that can be measured in real-time to identify P2P activity in short timescales. Traces of a number of p2p and non-p2p applications will be captured and analysed to gain confidence in the efficacy of wavelet analysis.

References

[ADD04] Andersen, F.U., De Meer, H., Dedinski, I., Kappler, C., Mäder, A., Oberender, J., Tutschku, K.: Enabling Mobile P2P Networking. In: Kotsis, G., Spaniol, O. (eds.) Euro-NGI 2004. LNCS, vol. 3427, pp. 219–234. Springer, Heidelberg (2005)

[AG03] Azzouna, N.B., Guillemin, F.: Analysis of ADSL traffic on an IP Backbone link. In: Proceedings of IEEE Globecom 2003, San Francisco, USA, December 1-5 (2003)

[AH01] Akansu, A.N., Haddad, R.A.: Multiresolution signal decomposition –Transforms, Subbands, and Wavelets. Academic Press, London (2001)

[AV98] Abry, P., Veitch, D.: Wavelet Analysis of Long Range Dependent Traffic. IEEE Transactions on Information Theory 44(1), 2–15 (1998)

[Bro97] Brownlee, N.: Traffic Flow Measurement: Experiences with NeTraMet, IETF, Network Working Group, RFC2123 (March 1997)

[CBP95] Claffy, K.C., Braun, H.-W., Polyzos, G.C.: A Parameterizable Methodology for Internet Traffic Flow Profiling. IEEE Journal On Sketched Areas In Communications 13(8), 1481–1494 (1995)

[Chu92] Chui, C.K.: An introduction to the wavelets. Academic Press, London (1992)

[Cla94] Claffy, K.C.: Internet Traffic Characterization. PhD thesis, University of California, San Diego, CA (1994)

[CPB93] Claffy, K., Polyzos, G., Braun, H.-W.: Application of Sampling Methodologies to Network Traffic Characterisation. In: ACM SIGCOMM 1993, San Francisco, California, USA, September 13-14 (1993)

[Dau92] Daubechies, I. (ed.): Ten Lectures on Wavelets. S.I.A.M (1992)

[EH96] Erlebacher, G., Hussaini, M.Y., Jameson, L.M. (eds.): Wavelets: Theory and Applications. Oxford University Press, Oxford (1996)

[END] http://www.endace.com

[Fry99] Fry, M., Ghosh, A.: Application Level Active Networking. Computer Networks 31(7), 655–667 (1999)

[GDS03] Gummadi, K.P., Dunn, R.J., Saroiu, S., Gribble, D., Levy, H.M., Zahorjan, J.: Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In: Proceedings of the nineteenth ACM symposium on Operating systems principles, Boston, October 19-22 (2003)

[Gnu] Gnutella, http://www.gnutella.com/

[GR02] Grossglauser, M., Rexford, J.: Passive Traffic Measurement for IP Operations. In: Park, K., Willinger, W. (eds.) The Internet as a Large-Scale Complex System. Oxford University Press, Oxford (2002)

[Kaz] KaZaa, http://www.kazaa.com/

[KBB03] Karagiannis, T., Broido, A., Brownlee, N., Claffy, K., Faloutsos, M.: File-sharing in the Internet: A characterization of P2P traffic in the backbone. Technical report (November 2003)

[KBB04] Karagiannis, T., Broido, A., Brownlee, N., Claffy, K.C., Faloutsos, M.: Is P2P dying or just hiding? In: IEEE Global Internet and Next Generation Networks (Globecom 2004), Dallas, Texas, USA, 29 November - 3 December, 2004,

[KBF04] Karagiannis, T., Broido, A., Faloutsos, M., Claffy, K.: Transport layer identification of P2P traffic. In: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement 2004 (2004)

[KPF05] Karagiannis, T., Papagiannaki, D., Faloutsos, M.: BLINC: Multilevel Traffic Classification in the Dark. In: ACM SIGCOMM 2005, Philadelphia, PA, USA (August 2005)

[LBB02] Leibowitz, N., Bergman, A., Ben-Shaul, R., Shavit, A.: Are File Swapping Networks Cacheable? Characterizing P2P Traffic. In: 7th International Workshop on Web Content Caching and Distribution (WCW 2003), Boulder, CO (2002)

[LHK04] Le Fessant, F., Handurukande, S., Kermarrec, A.-M., Massoulie, L.: Clustering in peer-to-peer file sharing workloads. In: Voelker, G.M., Shenker, S. (eds.) IPTPS 2004. LNCS, vol. 3279, pp. 217–226. Springer, Heidelberg (2005)

[Mal01] Mallat, S.: A Wavelet Tour of Signal Processing. Academic Press, San Diego (2001)

[MTT03] de Meer, H., Tutschku, K., Tran-Gia, P.: Dynamic Operation in Peer-to-Peer Overlay Networks. Praxis der Informationsverarbeitung und Kommunikation -Special Issue on Peer-to-Peer Systems (PIK Journal) (June 2003)

[NFL] Cisco IOS Netflow, on-line resource, http://www.cisco.com/warp/public/732/Tech/nmp/netflow/index.shtml

[Nla] http://www.nlanr.net/NA/Learn/packetsizes.html

[Ora01] Oram, A. (ed.): Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology. O' Reilly (2001)

[SFS01] Schmid, S., Finney, J., Scott, A.C., Shepherd, W.D.: Component-based Active Network Architecture. In: IEEE Symposium on Computers and Communications (July 2001)

[SSW04] Sen, S., Spatscheck, O., Wang, D.: Accurate, scalable in-network identification of p2p traffic using application signatures. In: Proceedings of the 13th international conference on World Wide Web (2004)

[SW02] Sen, S., Wong, J.: Analyzing peer-to-peer traffic across large networks. In: Second Annual ACM Internet Measurement Workshop (2002)

[THH04] Tagami, B., Hasegawa, T., Hasegawa, T.: Analysis and Application of Passive Peer Influence on Peer-to-Peer Inter-Domain Traffic. In: Proceedings of the Fourth International Conference on Peer-to-Peer Computing. IEEE, Los Alamitos (2004)

[Tuts04] Tutschku, K.: A measurement-based traffic profile of the eDonkey filesharing service. In: Barakat, C., Pratt, I. (eds.) PAM 2004. LNCS, vol. 3015, pp. 12–21. Springer, Heidelberg (2004)

[Zai97] Zait, M., Messatfa, H.: Comparative study of clustering methods. Future Gener. Comput. Syst. 13(2-3), 149–159 (1997)

[zse05] Zseby, T.: Sampling Techniques for Non-Intrusive QoS Measurements: Challenges and Strategies. In: Computer Communications Special Issue on Monitoring and Measurement (to appear, 2005)

[CMK99] Campbell, A.T., de Meer, H., Kounavis, M.E., Miki, K., Vicente, J.B., Villela, D.: A Survey of Programmable Networks. ACM SIGCOMM Comput. Commun. 29(2) (April 1999)