

Dynamic Link Measurements using Active Components

D. P. Pezaros[†], M. Sifalakis[†], S. Schmid[‡], D. Hutchison[†]

[†] Computing Department
InfoLab21, South Drive
Lancaster University,
LA1 4WA, U.K.
{dp, mjs, dh}@comp.lancs.ac.uk

[‡] NEC Europe Ltd.
Network Laboratories
Kurfürsten-Anlage 36
69115 Heidelberg, Germany
schmid@netlab.nec.de

Abstract. Active and programmable network technologies strive to support completely new forms of data-path processing capabilities inside the network. This in conjunction with the ability to dynamically deploy such active services at strategic locations inside the network enables totally new types of applications. In this paper we exploit these network-side programming capabilities to realise a new active network application that dynamically evaluates network link costs based on in-line traffic measurements. The performance experienced by the data packets (e.g. delays, jitter and packet loss) along network or virtual links is used to compute link costs based on multiple cost metrics. The results are published by means of a routing metric broker, which enables available routing protocols to calculate different sets of routes for different QoS metrics – as for example suggested for ToS-based routing (RFC 1583).

Introduction

Active networks research over the last decade has led to new developments in a number of areas ranging from secure programming languages [1,2], mobile code techniques [3], execution environments [3,4], active node platforms [5,6,7,8,9], service composition models [9,10,11], and so forth. Despite these valuable advances, the number of genuine applications where active network technologies are provably useful in real world networks is still limited.

A large number of applications proposed so far [12,13,14] aim to demonstrate the functionality of certain active platforms, while others try to address problems that are best solved with conventional techniques such as (mobile) agents. Active networks are often regarded as a neat technology in seek of genuine applications, which would persuade operators that the benefit of active networks exceeds the corresponding cost and risk involved in deploying and managing them.

We anticipate that active network services expose some properties that allow problems arising from e.g. network operation and management or service deployment, to be tackled in a more generic/elegant way. Being dynamically deployable on-demand, in a transparent and potentially automatic fashion at relevant points in the network, active services are suitable for a much wider range of applications/problems.

In this paper we focus on the use of the in-line traffic measurement framework [15] as an active service to facilitate dynamic routing link-cost updates that reflect fluctuations in traffic performance attributes. Routing adjustment in response to varying service quality characteristics can improve overall network stability and performance, and presents a challenging task that really benefits from active and programmable networks. In-line traffic measurements are used to assess the performance experienced by the flows along a transmission path, and measurement results are used to periodically adjust the network link costs in the routing protocol. As traffic measurements typically encompass a range of different characteristics (i.e. delay, jitter, packet loss, etc.), the calculation of link costs can be based on a multitude of different cost metrics. This allows for route optimisations tailored to specific applications or classes of applications with different QoS requirements (e.g. real-time synchronous applications vs. asynchronous applications).

The remainder of this paper is organised as follows: In section 2 there is a brief presentation of in-line IPv6-based measurements technique, and a discussion of why it is a particularly well-suited application of active networks technology. Furthermore, we describe the LARA++ active router framework, which has been used for the deployment and operation of the proposed service. In section 3, we present the design of the proposed active service, and we show how it can be implemented and deployed using LARA++. Section 4 presents some experimental results demonstrating the applicability of the in-line measurements and the proposed active service. In section 5, we examine the related work in the area of network measurements, and on dynamic metrics-based routing. Finally, in section 6, we conclude this paper by summarizing the proposed work and motivating some directions for future work.

Background

In-line IPv6 Traffic Measurement

In-line measurements [15] is a technique to assess the QoS properties experienced by IPv6 flows accurately, independent of a particular network topology and transparent to the end-user applications. The in-line measurements are carried out between two (or more) points in the network by piggybacking the relevant measurement data onto the actual data packets that are observed.

IPv6 extension headers [16] allow Type-Length-Value (TLV)-encoded data to be inserted between the main IPv6 header and the upper (transport) layer header. Depending on which type of extension header is used for the traffic measurements (for example, destination options header or hop-by-hop options header [16]), one can control where and when to trigger the measurement activity. For example, in case the destination options header is used, traffic measurements will only be triggered end-to-

end; whereas in the case of the hop-by-hop options header, any node along the transmission path could be involved. Moreover, the use of measurement information in the destination options header in conjunction with the routing header allows precise definition of where the traffic measurements in the network should take place.

The main benefit of this technique is that the traffic measurements are based on the actual user traffic rather than on general measurements based on other traffic flows. In addition to this property, by enforcing option processing only at identified nodes in the network and not hop-by-hop, in-line measurements eliminate the concern of instrumented packets being treated differently than the rest of the traffic in the network. Consequently, the measurements really reflect the performance experienced by the user data transmitted.

At the same time, the header extensions for the traffic measurements are defined by the network layer protocol itself, making the technique native, and equally applicable to any type of traffic (independent of the actual transport or user application).

Several measurement TLVs have been defined to be encoded within the IPv6 destination options header, which is examined by the final destination or optionally pre-defined intermediate nodes (based on the routing header) of a packet. Different TLVs implement a variety of performance metrics¹ by carrying packet departure/arrival timestamps, IP-based sequence numbers, trace information, etc. [17].

The clear separation of concerns between the measurement mechanism and particular analyses engines or post processing measurement applications, makes in-line measurements a promising candidate-application for active and programmable networks; measurement instrumentation is deployed only where and when required, and the results are used as input for a variety of network operations tasks.

Figure 1 show different points along an end-to-end transmission path, where in-line traffic measurements can be deployed. End-systems as well as selected intermediate network nodes can be equipped with in-line measurement functionality. The node that starts the traffic measurement process inserts the desired extension header into the relevant data packets. These packets are then processed by the instrumented nodes along the transmission path. The measurement information is recorded, amended and/or extracted accordingly.

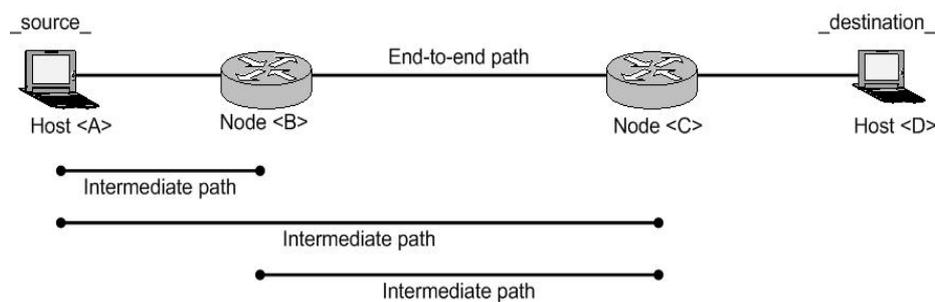


Fig. 1. The different notions of end-to-end

¹ A set of performance metrics are defined within the IETF's IPPM WG

In this paper, we focus on a particular application of in-line traffic measurements, whereby the measurements of up-to-date transmission characteristics (such as delay, jitter, packet loss and so forth) are used to dynamically update network link costs, so that routing decisions can be made more accurately.

Suitably selected active routers are used to deploy the measurement modules as they are needed, in order to adjust network link costs according to the current transmission performance. In the context of figure 1, active routers can be nodes and <C>, and the intermediate paths can be a point-to-point link or a virtual overlay link that spans across several hops in the underlying network.

The proposed traffic measurement application is particularly well-suited for active networks for a number of reasons:

- It is directly deployed on the data path
- It relies on direct access to data packets on the forwarding path (to support transparent measurements – independent from the applications)
- It needs to be deployed, activated, and configured dynamically whenever and wherever there is need for it

These characteristics advocate the realisation of the add-on service as an active service, offering the flexibility (and the ability) of on-demand deployment within the network.

The LARA++ Architecture

The LARA++ [9] active node framework is a software implementation of a programmable router that is designed for commodity operating systems. It augments the functionality of a conventional router/host by exposing a programmable interface which allows active programs, referred to as active components, to provide network level services on any packet-based network.

Since LARA++ “hooks” directly into the router’s operating system, it enables the transparent interception of packets traversing the node. Intercepted packets can be processed by active components and then be re-injected back into the host OS for the default processing on the node. In this way, LARA++ can flexibly extend (as opposed to alter) the functionality of a router’s conventional network services, enabling lightweight augmentation of existing network services and allowing for gradual replacement of conventional router functionality. We consider this feature especially useful for our in-line traffic measurements, since it facilitates transparent processing of the relevant data traffic within selected nodes.

LARA++ treats a router as a resource shared by all its users. The extent of programmability can be adjusted on a per-user or group-of-users basis, as well as based on resource availability. Active components of different LARA++ users are protected from each other by a safety model that gives each component a sandbox called a processing environment (PE). For performance reasons, however, LARA++ allows users that trust each other to execute their active code inside the same PE.

LARA++ uses a sophisticated model for service composition [18]. Each component that is to become part of the service composite on a running active router installs one or several packet filters into nodes of a directed graph, referred to as the classification graph. These packet filters specify rules that LARA++ uses to determine

if traversing packets need to be processed locally. Once a packet is matched, LARA++ delivers the packet to the components that registered the filter for processing. The use of a configurable classification graph allows LARA++ to process packets of any type ranging from standard IP packets over active packets with ANEP-style [19] headers to completely bespoke packet formats. As a result, service composition on LARA++ active router is defined implicitly by the classification graph and the packet filters installed by the active components. This type of composition approach provides a means to control both co-operation and competition among active components [18]. Figure 2 illustrates this concept in more detail.

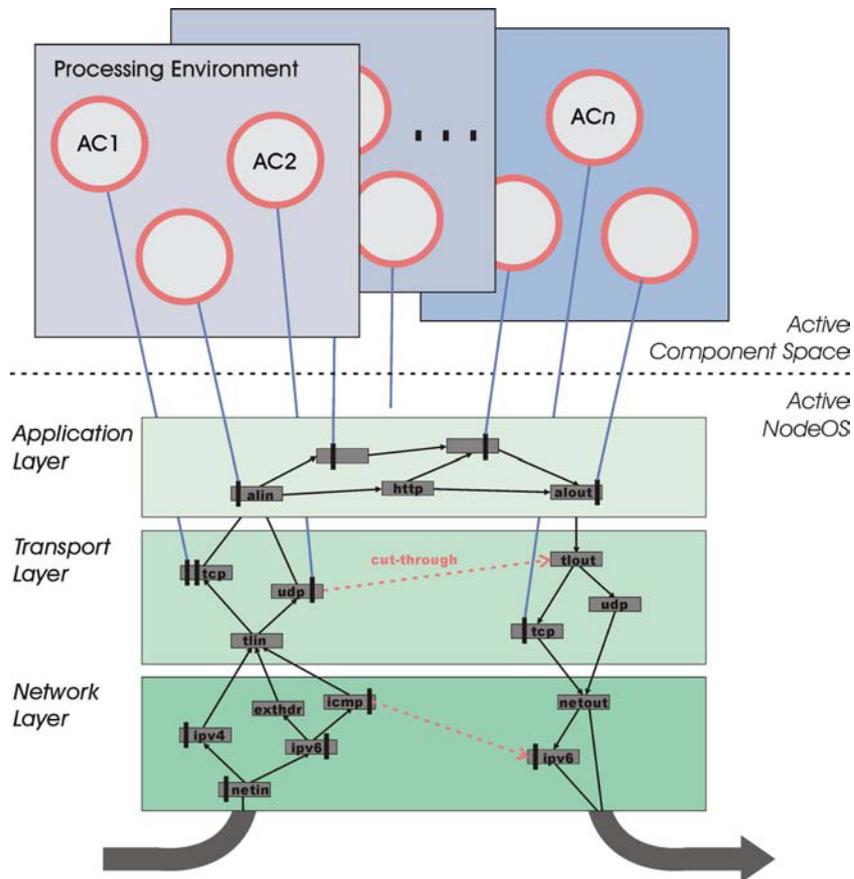


Fig. 2. The LARA++ Classification Graph

Packet filters are extremely flexible from the component developer point of view, because they enable the description of packets that are subject to active processing based on any bit or byte pattern. Yet, in most common cases it is sufficient to consider the flow information and/or the existence of specific header values in the packet. LARA++ filters are easily specified and installed by active components using an XML-based mark-up language.

The LARA++ active router framework also encompasses a generic service deployment protocol, called ASDP [20], that allows dynamic deployment and control of active services on remote active routers. We consider this particularly valuable in order to deploy traffic measurement support inside the network where desired.

Design and Implementation

The proposed mechanism consists of two main modules. The first module is responsible for carrying out the in-line measurements. It has been implemented as a standalone LARA++ active component that registers the relevant packet filters (depending on the flows of interest) at the IPv6 node of the classification graph [9]. It exposes an API that allows other components or user applications to use it. The second module is a user-space application that reads and processes the measurement data, and maintains a data structure with separate metrics for the different types of measured attributes in the network (e.g. delay, jitter, packet loss, and so on). In order for the two modules to interface effectively, the evaluation module registers a callback interface with the measurements module. Periodically, the measurements module contacts the broker through the callback interface to stream the raw measurement data. Figure 3 illustrates the design of the proposed mechanism and shows how the two modules interface with each other.

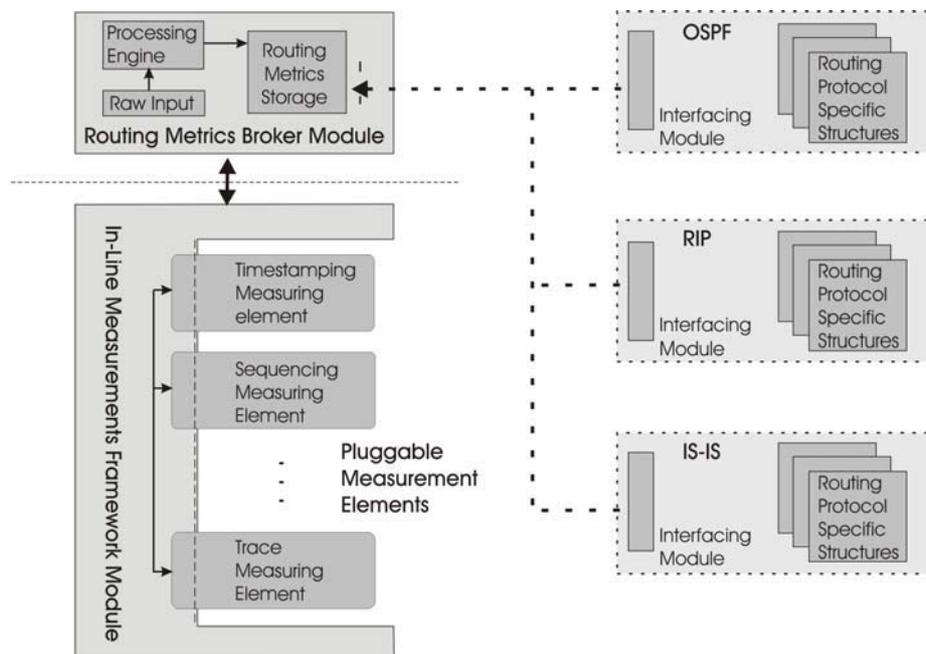


Fig. 3. Basic Design of the Active Service

The following sections describe in more detail the internal implementation and functionality of these two modules.

In-line measurements active components

The central functionality of this module is to perform the in-line measurements. As shown in figure 3, this module consists of two main parts: i) a plug-in framework and ii) a set of measurement plug-ins. The plug-in framework provides the functionality for creating the appropriate destination options extension headers and for inserting/extracting the measurement data of previous nodes (which are encoded as TLV options). It exposes the necessary API for other applications to access the measurement data, to manage (add/remove/configure) the plug-ins, and to configure the filtering parameters. Filtering can be based on the source and destination addresses and ports, on transport protocol, traffic class, and on flow label values.

The plug-ins are the code elements that carry out the actual measurements and generate the appropriate data that are inserted in the IPv6 packet header by the framework. Separate plug-ins are used for different types of measurements (e.g. transmission times and packet loss). The framework can accommodate several plug-ins simultaneously for performing different measurement (which results in more than one TLV options records in the IPv6 destination header), although this comes at the cost of reducing the data payload size. However, different plug-ins can create separate measurement TLVs for different data of interest.

The framework API provides an IOCTL-based interface for registering/attaching the plug-ins, and also adjusts the configuration parameters of the plug-ins such as the filtering and sampling granularity. The sampling rate can be configured by defining whether the module should instrument all packets matching the filtering criteria, 1-in-N, or act at a specific temporal sampling rate.

For the purposes of our prototype implementation, we have used two plug-ins to perform time and loss-related measurements accordingly. The first plug-in has been designed to measure one-way delay (OWD) between two points along a transmission path, as well as more synthetic time-related parameters such as jitter and throughput. This first plug-in is used to insert and record departure and arrival timestamps of packets at the respective measurement nodes along the transmission path. The two measurement nodes (that add and remove the packet timestamps) synchronise their time through the Network Time Protocol (NTP) [21].

The second plug-in enables one-way loss measurements by means of IP-based sequencing of packets. A source node inserts incremental sequence counters to packets belonging to the same flows, which are then observed at the destination. Packet loss as well as out of order delivery can be effectively measured by computing the differences of the TLV sequence numbers between successive packets.

A flow in this context can be defined at different levels of granularity. At a fine granularity level, it can be the sequence of packets with the same source and destination IP addresses and transport ports. On the contrary, a flow can also be defined by all the packets traversing a certain point-to-point or virtual/overlay link. The next hop will also have to run the corresponding measurement module to keep track of sequence numbers as the packets arrive.

The in-line measurement component implemented for our LARA++ active router architecture registers the following packet filters with the classification graph: one or more filters for the outgoing packets of interest (the number of filters here depends on the filtering parameters configured by the measurement application) and one filter for the incoming IPv6 packets that contain our measurement header. Once a packet of interest is filtered, it is pulled out of the forwarding path and handed to our in-line measurement component. Depending on which filter captured the packet, TLV-options are either inserted or extracted accordingly. The packets are then inserted back to the classification graph for further processing.

The information extracted from the incoming packets is delivered to the external broker module that has expressed interest in the respective measurement data. The role of the broker module is further described in the following section.

Routing metrics broker module

This control module accesses the in-line measurement active component in order to configure the in-line traffic measurements and collect the results. It is responsible for extracting and processing the appropriate raw measurement data, and for updating the costs table according to the routing metrics of interest. Node-local running routing protocols can then access these up-to-date cost metrics and optimise their routing information. In this way, routing protocols can always decide optimal routing paths based on up-to-date link quality information (with regard to the chosen metrics).

The current prototype of the broker module is implemented as a user-space application. The traffic measurement process starts by initialising the broker module where the user specifies the (virtual) link and the packet flows that should be used for the in-line traffic measurements. The user also selects which measurement plug-in instruments which flows.

At start-up, the broker module instruments the installation and activation of the in-line measurement components on the respective active routers on both ends of the (virtual) link. Note that in the case of a virtual link (tunnel), the in-line measurements module will be installed several routing hops apart from each other, which enables the measurements for a whole routing path as opposed to a single physical link. This process takes place using the existing active network loading mechanism supported by LARA++.

The broker module then establishes the necessary communication channels with the in-line measurement components, to pass configuration parameters such as the packet filters and sampling rate, and to receive the measurement results. Based on this data, the broker computes the appropriate link cost metrics that have been registered by the routing protocol(s) or other applications.

Once the in-line measurement component starts performing the measurements and delivering the measurement results, the broker module processes the data and updates the cost metrics data structure. This data structure stores single-value link costs for each measured attribute. It is accessible by the routing protocols through a “well-known” API so that they can update their internal data structures periodically, in order to reflect the dynamic link cost changes.

Since our main goal is to demonstrate the proposed functionality, our current implementation simply sets the cost values by averaging the N most recent measurements. More sophisticated calculations could be based on averaging a set of past cost values combined with the N most recent results or any other algorithm that would deliver a less fluctuating set of cost values.

Furthermore, since existing routing protocols typically do not use generic data structures among them and neither share a common representation of link costs or routing metrics, those ones that want to benefit from the in-line traffic measurements have to be extended. As shown in figure 3, we propose that the routing protocol will interface with our broker module through its own proprietary interface adapter. For example, the interface adapter for OSPFv3 would calculate OSPF-specific link costs from the measurement results and update the internal data structures accordingly.

Evaluation of the in-line IPv6 measurements mechanism

For the evaluation of our mechanism we used the IPv6 testbed infrastructure [22] at Lancaster University, where we have deployed two LARA++ active nodes (at points A and B) as illustrated in figure 4. We created an artificial, yet realistic, network condition, where we stressed the ADSL uplink (at the tunnel connection) by generating 512-byte TCP/UDP traffic at an exponentially increasing rate of up to 62 packets per second. The WaveLAN link on the other hand, being part of the campus WiFi network was subject to the usual (relatively congested) traffic encountered at midday hours.

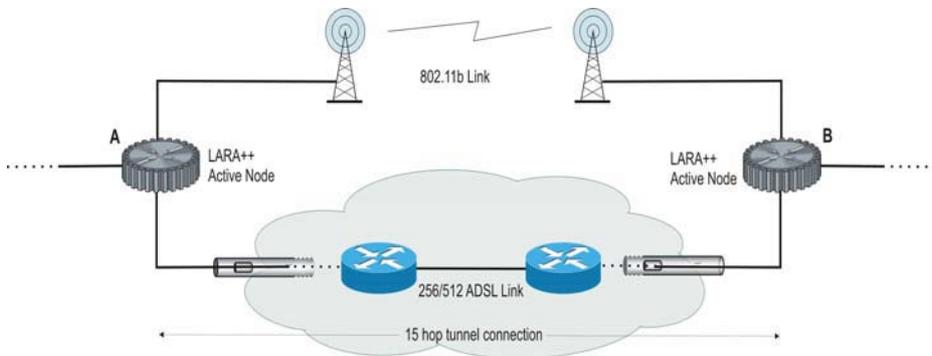


Fig. 4. Experimental Network topologies

We then triggered the installation of the proposed active service on node A and B according to the process described in section 3, and deployed the timestamping plug-in (for one-way delay and jitter) to instrument UDP traffic and the sequencing plug-in (for packet loss) for measurements on TCP traffic, respectively. These choices are justified by the fact that TCP performance is known to be vulnerable to packet loss (continuous back-off), whereas UDP performance is impacted by increasing delays and delay variations (buffer adjustment requirements).

After processing the measurements for appropriate time intervals using the broker module, we got the results illustrated in figures 5 (ADSL downlink), 6 (ADSL uplink) and 7 (802.11b), with regard to packet loss, delay and jitter. Table 1 summarises these results.

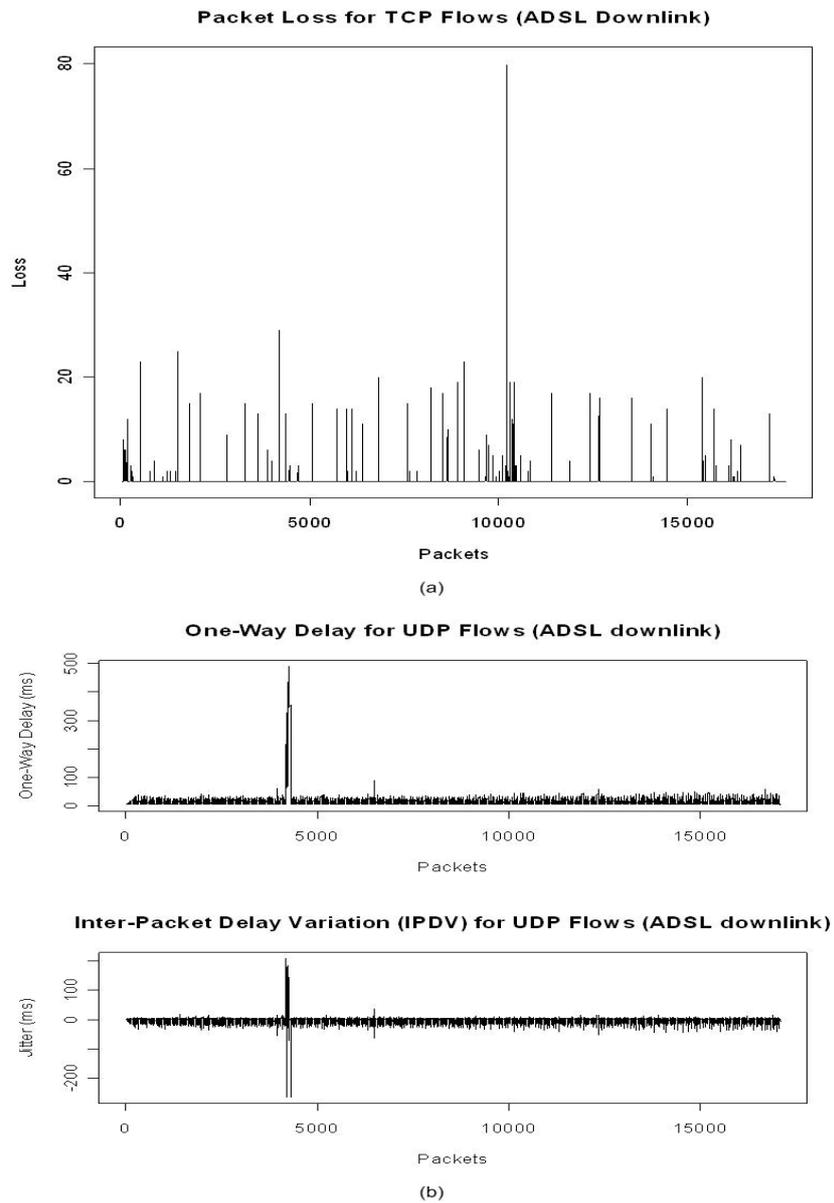


Fig. 5. ADSL Downlink Measurements - (a) TCP Packet Loss (b) UDP OWD and Jitter

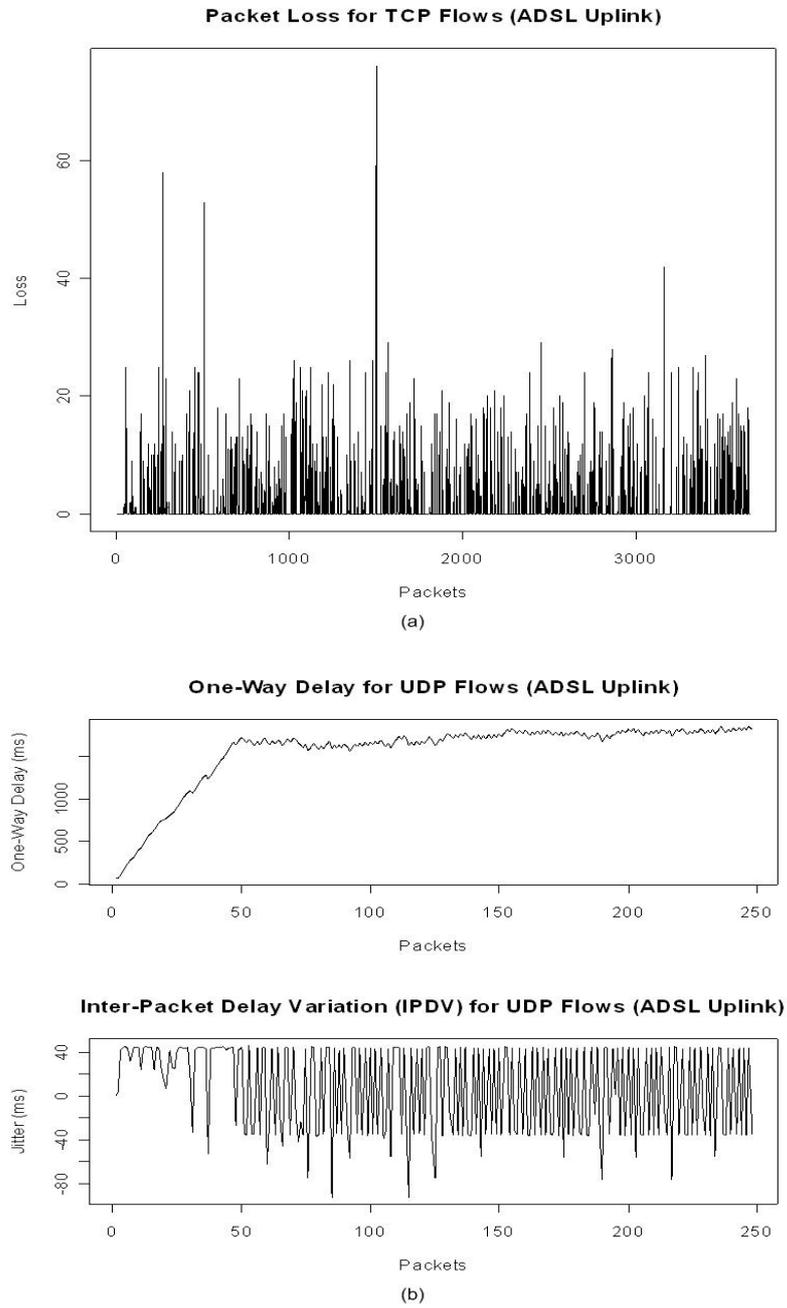


Fig. 6. ADSL Uplink Measurements - (a) TCP Packet Loss (b) UDP OWD and Jitter

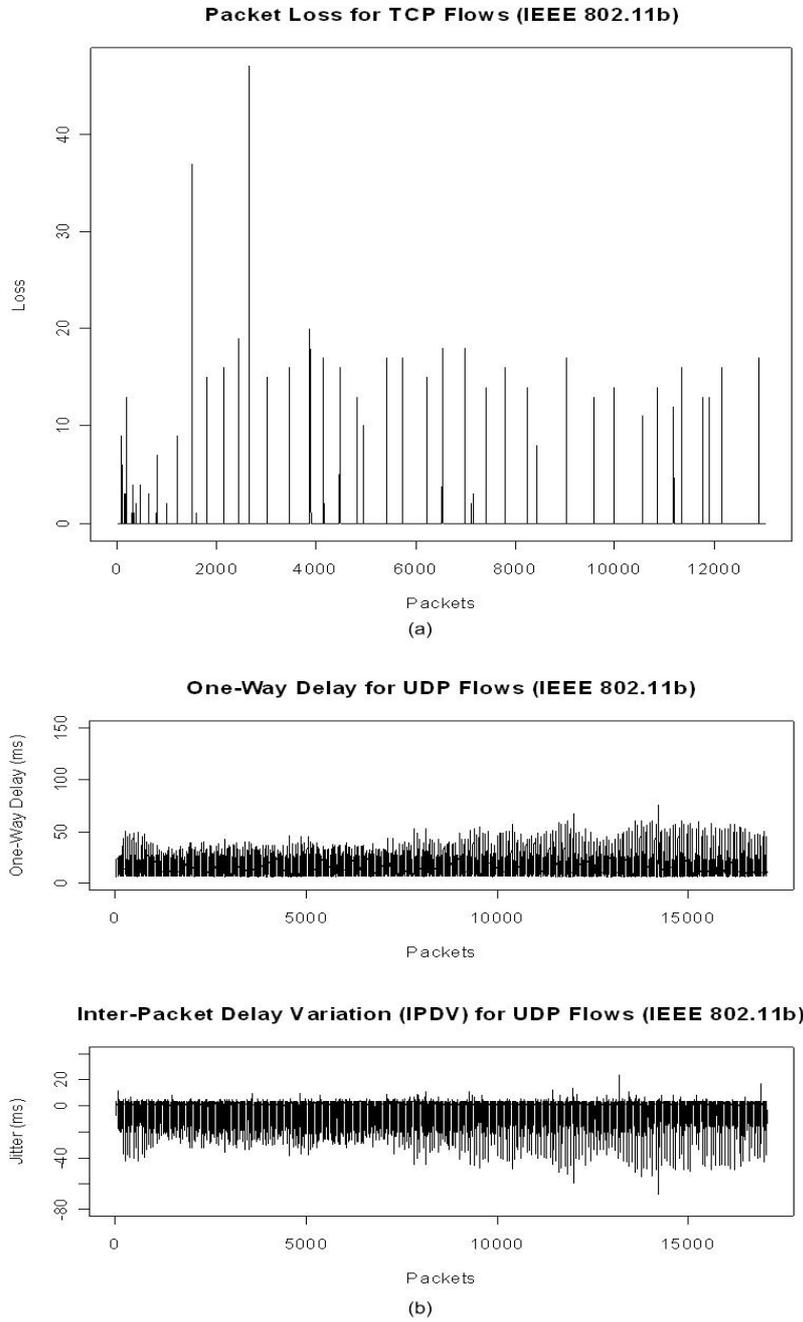


Fig. 7. IEEE 802.11b Measurements - (a) TCP Packet Loss (b) UDP OWD and Jitter

Comparing figures 5 and 7, we observed that the packet loss on the tunnel link is approximately 5% versus 4.8% for the WaveLAN link, yet the wireless link exposes more bursty characteristics. The delay experienced by the UDP flows over the tunnel connection was slightly better than on the WaveLAN link: the mean delay was 15.4 ms over the tunnel link versus 19 ms on the wireless link and 75% of the measurements yielded values less than 18 ms over the tunnel as opposed to 23 ms on the wireless link. Finally, jitter in both cases is almost the same, with most values laying between 1 and 2 ms. As a result, under the current congestion patterns, the WiFi network and the one (downlink) direction of the tunnel link exhibit similar characteristics, with the tunnel link having slightly better and more stable behaviour.

Figure 6 shows the performance of the ADSL uplink while becoming increasingly saturated by the artificially introduce data traffic. Under the very high stress, the tunnel link is hardly usable: 53.6% packet loss and rapidly increasing delays. This can be observed in the upper plot of figure 6(b).

Under these traffic conditions, we can derive that the fittest routing configuration in our testbed is the asymmetric routing of traffic from A to B over the tunnel link and from B to A through the wireless link.

Table 1. Performance Statistics for the Different Links

	Delay		Jitter		Packet Loss
	Mean	75% Quantile	25% Quantile	75% Quantile	
ADSL Uplink	1558 ms	1768	-35 ms	44 ms	53.6%
ADSL Downlink	15.4 ms	18 ms	1 ms	2 ms	5 %
IEEE 802.11b	19 ms	23 ms	1 ms	2 ms	4.8 %

Currently, according to the conventional operation of routing protocols, a router would select either the WiFi link or the tunnel connection to transport traffic between the points A and B. The choice would be based on static costs assigned to the two links based on their media type. In our setup it would always select (unless otherwise instructed) the WiFi link, since by default it is preferred over a virtual link, even if the latter had bigger capacity. This happens because the link cost assigned to the virtual link in absence of any other qualitative information is based on the distance metric. This default configuration can only change through the manual and static intervention of the administrator.

Based on our proposed mechanism, the routing protocol can dynamically adjust the link costs of both the tunnel and the wireless links based on the dynamic in-line traffic measurements (in response to their varying characteristics). In the case of our particular setup, the routing protocol would be able to detect the need for asymmetric routing and adapt the routing accordingly. Also, the ability of performing in-line measurements for specific classes of traffic, for example based on the transport protocol or Type-of-Service (ToS), is expected to benefit the implementation of ToS-based routing.

Related Work

Existing traffic measurement techniques and infrastructures fall into two main categories, namely active and passive techniques. Active measurement techniques inject additional traffic with known characteristics into the network to test particular attributes of a service [23, 24, 25, 26], and they have been focusing on characterising properties of end-to-end network paths between instrumented systems. Passive measurements give highly accurate results by observing and analysing real traffic on a link without disruption of the service. They mainly operate at a single observation point within an administrative domain and try to provide feedback for network operations tasks [27, 28, 29, 30].

Active and passive measurements rely either on the performance experienced by dedicated traffic or on the costly correlation of one-point observations to yield one-way performance results, and do not provide a framework for performing accurate and transparent service-quality measurements for different traffic flows that can be deployed on-demand in the network.

Work on QoS Routing research has considered adaptive routing based on dynamic cost metrics. Some early work focused on ToS routing [31], which is either based on using multiple instances of routing protocols or maintaining routing tables with multiple metrics for different network attributes (i.e. delay, throughput, etc). Nevertheless, the link costs considered were static according to the natural characteristics of the link as it is the case with most routing protocols today.

Other solutions that have been proposed in this area advocate the use of destination-driven/initiated routing path computations and updates towards them [32, 33]. Clearly, these solutions are neither scalable nor pervasive, since they typically involve flooding mechanisms that cause both significant traffic overhead and high complexity. As a result, they suggest viable solutions only for maintaining routing paths to a small number of frequently used destinations. And, although they are quite dynamic, they often don't account for multiple metrics needed for the different attributes.

A different approach for tackling congestion problems and therefore improving communication, has led to the idea of multipath routing. Work, such as the one presented in [34] and [35], propose probabilistic or other methods of load balancing the traffic across multiple routing paths. Although these solutions differ fundamentally from our approach, yet, we believe that our mechanism can complement these solutions to improve their performance through dynamic adaptation.

Finally, the deployment of reconfigurable middleboxes or active network-based solutions has been considered in [36] and [37] in order to adapt or change the network configuration to match current traffic requirements. However, most of these solutions are not embeddable in general routing fabrics, but rather focus on out-of-band allocation of QoS resources in order to improve communication for individual flows.

Conclusions and Future Work

In this paper we have presented a new service for active networks based on the concept of using in-line traffic measurements to improve intra-domain routing. The main idea and contribution of this work is to provide the necessary mechanisms to accurately measure dynamically changing link (or virtual link) properties including delay, jitter and packet loss. This measurement information is then fed to available routing protocols so that they exchange more accurate, performance information about the links, enabling more prudent routing decisions. We have focused on the functionality and the design of the proposed mechanism to advocate its feasibility and applicability in an active network environment.

Future work will investigate the impact of the proposed mechanism on the routing protocol behaviour, as well as on how the measurement results can be used to generate meaningful link cost values (for specific protocols). We aim to tackle the fine-tuning of the mechanism and its viability when used in conjunction with today's routing protocols, in wired, wireless, and overlay network environments. Steps in this direction include the investigation of optimal time and sampling intervals for the measurement processes, as well as of novel algorithms for the link cost calculations. Application-specific routing based on performance properties of interest to different flows can also be facilitated.

We expect the proposed traffic measurement solution to be particularly applicable to overlay networks and mobile ad-hoc networks, since it enables the deployment of an always-on active service at strategic locations, where network characteristics change rapidly. In overlay networks these rapid changes result from the combination of a (often varying) number of underlying physical links that form the virtual network links, whereas in the case of mobile ad-hoc networks it is a result of the often changing mobility patterns and the environment affecting the wireless interfaces. We anticipate the proposed solution to be particularly valuable in both these cases.

Acknowledgements

We are grateful to Agilent Technologies for the support of Dimitrios Pezaros' work through an industrial fellowship. We would also like to acknowledge the support of the EPSRC under grant number GR/R31461/01.

References

- [1]. Wakeman, I., Jeffrey, A., Owen, T., Pepper, D., SafetyNet: A Language-Based Approach to Programmable Networks, in *Computer Networks and ISDN Systems*, 36 (1). 2001.
- [2]. The Caml Language. Online Reference, INRIA, <http://caml.inria.fr/>.
- [3]. Wetherall, D., J., Guttag, J., Tennenhouse, T., L., ANTS: A toolkit for building and dynamically deploying network protocols, in *Proc. of IEEE Openarch*, April 1998.

- [4]. Hicks, M., W., Kaddar, P., Moore, J., T., Gunter, C., A., Nettles, S., PLAN: A Packet Language for Active Networks, In Proceedings of the 3rd ACM SIGPLAN International Conference on Functional Programming, pages 86-93, 1998.
- [5]. Paterson, L., Gottlieb, Y., Hibler, M., Tullmann, P., Lepreau, J., Schwab, S., Dandelkar, H., Purtell, A., Hartman, J., An OS Interface for Active Routers, IEEE Journal on Selected Areas in Communications, Volume 19, Issue 3, March 2001, pp. 473-487.
- [6]. Merugu, S., Bhattacharjee, S., Zegura, E., Calvert, K., Bowman: A Node OS for Active Networks, in Proceedings of IEEE INFOCOMM'00, Tel Aviv, Israel, March 26-30, 2000.
- [7]. Keller, R., Choi, S., Decasper, D., Dasen, M., Fankhauser, G., Plattner, B., An Active Router Architecture for Multicast Video Distribution. In Proc. of IEEE INFOCOM (3), pp 1137-1146, 2000.
- [8]. Keller, R., Ruf, L., Guindehi, A., Plattner, B., PromethOS: A Dynamically Extensible Router Architecture Supporting Explicit Routing, in Proceedings of the 4th International Conference on Active Networks (IWAN), Zurich, Switzerland, December 4-6, 2002.
- [9]. Schmid, S., Finney, J., Scott, A., C., Shepherd, W., D., Component-based Active Network Architecture, IEEE Symposium on Computers and Communications, July 2001.
- [10]. Merugu, S., Bhattacharjee, S., Chae, Y., Sanders, M., Calvert, K., Zegura, E. Bowman and CANEs: Implementation of an Active Network, In Proc. of 37th Conference on Communication, Control and Computing, September 1999.
- [11]. Bossardt, M., Antik, R., H., Moser, A., Plattner, B., Chameleon: Realising Automatic Service Composition for Extensible Active Routers, in Proceedings of the 5th International Conference on Active Networks (IWAN), Kyoto, Japan, December 10-12, 2003.
- [12]. Bassi, A., Gelas, J-P., Lefevre, L., A sustainable Framework for Multimedia Data Streaming, in Proceedings of the 5th International Conference on Active Networks (IWAN), Kyoto, Japan, December 10-12, 2003.
- [13]. Lefevre, L., Pierson, J-M., Guebli, S., Collaborative Web-Caching with Active Networks, in Proceedings of the 5th International Conference on Active Networks (IWAN), Kyoto, Japan, December 10-12, 2003.
- [14]. Hand, S., Harris, T., Kotsovinos, E., Pratt, I. Controlling the XenoServer Open Platform, in Proceedings of IEEE OpenArch'03, San Francisco, California, April 4-5, 2003.
- [15]. Pezaros, D., P., Hutchison, D., Garcia, F., J., Gardner, R., D., Sventek, J., S., In-line Service Measurements: An IPv6-based Framework for Traffic Evaluation and Network Operations, in Proceedings of IEEE/IFIP NOMS 2004, Seoul, Korea, April 19-23, 2004.
- [16]. Deering, S., Hinden, R., Internet Protocol Version 6 (IPv6) Specification, IETF, IPNG Working Group, RFC 2460, December 1998.
- [17]. Pezaros, D.,P., Hutchison, D., Garcia, F.,J., Gardner, R., Sventek, J.,S., Service Quality Measurements for IPv6 Inter-networks, to appear in International Workshop on Quality of Service (IWQoS), Montreal, Canada, June 7-9, 2004.
- [18]. Schmid, S., Chart, T., Sifalakis, M., Scott, A., C., "Flexible, Dynamic and Scalable Service Composition for Active Routers", In Proc. of IWAN 2002, pp 253-266, December 2002.
- [19]. Alexander, D., S., Braden, B., Gunter, C., A., Jackson, W., A., Keromytis, A., D., Minden, G., A., Wetherall, D., A., Active Network Encapsulation Protocol (ANEP), July 1997.
- [20]. Sifalakis, M., Schmid, S., Chart, T., Hutchison, D., "A Generic Active Service Deployment Protocol". In Proc. of ANTA 2003, pp 100-111, Osaka, May 2003.
- [21]. Mills, D., Internet time synchronisation: the Network Time Protocol, IEEE Transaction on Communications, Volume 39, Issue 1, October 1991, pp. 1482-1493.
- [22]. MSRL – Mobile-IPv6 Systems research Lab". Research Project funded by Cisco Systems, Microsoft Research (Cambridge), and Orange Ltd, Lancaster University, 2001.

- [23]. Matthews, W., Cottrell, L., The PingER project: Active Internet Performance Monitoring for the HENP Community, IEEE Communications Magazine, Vol. 38, Issue 5, May 2000, pp. 130-136.
- [24]. Kalidindi, S., Zekauskas, M., J., Surveyor: An Infrastructure for Internet Performance Measurements, in Proceedings of the ninth Annual Conference of the Internet Society (INET'99) INET'99, San Jose, California, June 22-25 1999.
- [25]. Georgatos, F., Gruber, F., Karrenberg, D., Santcroos, M., Susanj, A., Uijterwaal, H., Wilhelm, R., Providing Active Measurements as a Regular Service for ISP's, in Proceedings of Passive and Active Measurement Workshop (PAM2001), Amsterdam, NL, April 23-24 2001.
- [26]. NLANR Active Measurement Project (AMP) Homepage, <http://watt.nlanr.net/active/intro.html>.
- [27]. Apsidorf, J., Claffy, K., C., Thompson, K., Wilder, R., OC3MON: Flexible, Affordable, High Performance Statistics Collection, in Proceedings of the seventh Annual Conference of the Internet Society (INET'97), Kuala Lumpur, Malaysia, June 24-27 1997.
- [28]. Fraleigh, C., Diot, C., Lyles, B., Moon, S., Owezarski, P., Papagiannaki, D., Tobagi, F., Design and Deployment of a Passive Monitoring Infrastructure, in Proceedings of Passive and Active Measurement Workshop (PAM2001), Amsterdam, NL, April 23-24 2001.
- [29]. Feldmann, A., Greenberg, A., Lund, C., Reingold, N., Rexford, J., True, F., Deriving Traffic Demands For Operational IP Networks: Methodology And Experience, in Proceedings of ACM SIGCOMM'00, Stockholm, Sweden, August 28 – September 1 2000.
- [30]. Claffy, K., C., Miller, G., Thompson, K., The Nature Of The Beast: Recent Traffic Measurements From An Internet Backbone in Proceedings of the eighth Annual Conference of the Internet Society (INET'98), Geneva, Switzerland, July 21-24 1998.
- [31]. Matta, I., Shankar, U., A., Type-of-Service Routing in Dynamic Datagram Networks, in Proceedings of IEEE INFOCOMM'04, Toronto, Ontario, Canada, June 12-16, 1994.
- [32]. Chen, J., Druschel, P., Subramanian, D., A New Approach to Routing with Dynamic Metrics, in Proceeding of IEEE INFOCOMM'98, San Francisco, USA, 29 March- 2 April, 1998.
- [33]. Di Fatta, G., Gaglio, S., Lo Re, G., Ortolani, M., Adaptive Routing in Active Networks, Proceedings of IEEE OpenArch 2000, Tel Aviv, Israel, March 2000.
- [34]. Nelakuditi, S., Zhang, Z-L., Tsang, R., P., Du, D., H., C., Adaptive Proportional Routing and Localised QoS Routing Approach, in Proc. of IEEE INFOCOMM'00, Israel, March 26-30, 2000.
- [35]. Bohacek, S., Hespanha, J., P., Obraczka, K., Lee, J., Lim, C., Enhancing Security via Stochastic Routing, in Proceedings of the 11th International Conference on Computer Communications and Networks (ICCCN'02), Miami, Florida, October 14-16, 2002.
- [36]. Matta, I., Bestavros, A., QoS Controllers for the Internet, In Proceedings of the NSF Workshop on Information Technology, Cairo, Egypt, March 2000.
- [37]. Vrontis, S., Sygkouna, I., Chantzara, M., Sykas, E., Enabling Distributed QoS Management Utilising Active Network Technology, in Proc. of Net-Con'03, October 2003.