Application Layer Active Networks

Ian Marshall, BT Labs, Martlesham Heath, Ipswich, IP5 3RE
Ian.W.Marshall@bt.com

Position

The deployment of new communication services is currently restricted by slow standardisation, the difficulties of adding new capabilities to existing equipment, and the complexity of the existing network. It has been proposed [1] that these difficulties could be overcome by adding programmes to packet headers which are intended to run on network devices the packet encounters. However service operators will never permit third party programmes to run on their equipment without prior testing, and a strong guarantee that the programme will not degrade performance for other users. Since it will be extremely hard to create interesting programmes, in a language which is simple enough to enable termination guarantees, we regard this approach as unrealistic.

A somewhat different flavour of active networking in which the packets do not carry programmes, but flags indicating the desirability of running a programme has also been proposed [2]. This would enable the service operator to choose a programme to load, which matches the indicated need, but is sourced from his own database of tested programmes. We believe that this second approach may be valuable in the long term, however in order not to degrade router performance the number of flags will need to be small and the number of possible programmes will thus also be small. In addition it will not resolve the immediate need for increased flexibility, as it will require standardisation, and will thus take time.

We therefore propose a third alternative, which we call application layer networking [3]. In this system the network is populated with application level entities we refer to as service nodes, or dynamic proxy servers. These could be thought of as equivalent to the http caches currently deployed around the internet, but with a hugely increased and more dynamic set of capabilities. This approach relies on redirecting selected packets into the application layer device, where programmes can be run to modify the content, or the communication mechanisms. The programmes can be selected from a trusted data source, and can be run without impacting router performance. Packets are redirected on a session by session and protocol by protocol basis, so there is no need for additional flags or standardisation. Programmes are chosen using the mime type of the content, so again no additional data or standards are required.

In our initial implementation, programmes to be run on the dynamic proxy server are implemented as java beans for portability reasons, and the proxy server itself is a java application. The beans can be loaded on demand and are referred to as proxylets (an analogy with applets and serverlets). A proxylet is a specific type of bean, which supports the operations Load, Run, Stop and Modify. If stateful programmes are required it is necessary to add further operations to the list such as Suspend (stores state pending restart – possibly at another location). The redirection of packets in a session carrying a particular protocol (such as http) to the proxy is initiated either by the client of the session or by the server. This avoids the issues associated with transparent redirect, and in our view is preferable, but transparent redirect would be possible.

We have shown, by implementing appropriate proxylets, that our approach offers benefits for streaming of retreived audio and video files, for link compression, for enabling multicast content to propagate over unicast links, and for bridging tcp sessions. We are currently developing proxylets which perform authentication (for mobile access), content transcoding (for caching dynamic content and for adapting to low functionality clients), dynamic redirection, and message queuing. We anticipate demonstrating benefits for all the proxylets under development. We are aiming to release code for wide area experiments in 12-18 months, after we have designed a management and configuration system for the dynamic proxy server, and after we have created a performance optimised implementation of the server.

We believe that application layer active networking has a crucial role to play, both relatively soon and in the long term, particularly for applications where latencies of a few hundred milliseconds are acceptable. If application layer active networking is deployed the relatively small number of higher performance requirements could be handled at lower layers by placing flags in packets without requiring large numbers of flags and potentially negating the benefits

References

[1] D.Tennenhouse, D.Wetherall, “Towards an active network architecture” Computer communication Review, 26, 2 (1996), pp5-18

[2] Alexander, Shaw, Nettles and Smith “Active Bridging” Computer Communication Review, 27, 4 (1997), pp101-111

[3] M.Fry and A.Ghosh “Application layer active networking” HIPPARCH ’98 Workshop