ANGEL - Architecture

Jason But, Lawrence Stewart, Sebastian Zander, Grenville Armitage
Centre for Advanced Internet Architectures, Technical Report 070228A
Swinburne University of Technology
Melbourne, Australia
{jbut, lastewart, szander, garmitage}@swin.edu.au

***Abstract*—The Automated Network Games Enhancement Layer (ANGEL) project aims to leverage Machine Learning (ML) techniques to automate the classification and isolation of interactive (e.g. games, voice over IP) and non-interactive (e.g. web) traffic. This information is then used to dynamically reconfigure the network to improve the Quality of Service provided to the current interactive traffic flows and subsequently deliver improved performance to the end users. Within this scope, the project will develop protocols that allow the adjustment of Consumer Premises Equipment (CPE - eg. cable/ADSL) configuration to provide better quality of service to interactive flows detected in real-time.**

This document provides the motivation and describes typical use cases for the ANGEL architecture. It also defines the basic building blocks of the architecture and specifies the requirements for them.

I. Introduction

With the increase in and affordability of broadband access in the forms of Asymmetric Digital Subscriber Loop (ADSL), Cable Internet and 802.11x infrastructure, comes the real possibility of useable, real-time services to the home. Examples include multiplayer online networked games, streaming audio/video content and voice over IP (VoIP). There is also the next generation of high speed broadband network architectures to consider such as ADSL2 and IEEE 802.16 etc. which are going to support high definition digital television on demand and the like, along with the previous suite of services.

However, a heterogenous network traffic environment such as a home or small business LAN makes demands on the underlying network infrastructure that can cause it to become a bottleneck e.g. someone playing a game or making a VoIP call whilst someone else uploads a large file. The game or VoIP traffic, being real-time and interactive, is far more sensitive to network delay and jitter than a TCP file transfer. The net result in this scenario is that the game play or phone conversation degrades significantly, whilst the only difference observed for the file transfer is a throughput decrease and therefore an increased download time.

So how can we ensure these two types of network traffic can coexist happily? The solution we propose involves separating them into two groups: real-time/interactive and the rest. Once separated, we can prioritise the first group and ensure any special requirements for this traffic class are met. This should occur dynamically and without any user/administrator or application intervention.

For most home Internet users, the bandwidth bottleneck in Internet paths is typically found in the last-mile link between the ISP network and the customer. The Automated Network Games Enhancement Layer (ANGEL) project aims to develop a real-time network traffic classification and prioritisation system for use in Internet Service Provider (ISP) broadband access networks. The system aims to address the issue of traffic Quality of Service (QoS) for real-time/interactive traffic at the network edge i.e. customer to ISP links. The system will specifically target the upstream CPE to ISP link, which often tends to be at least 4 times slower than the downstream speed (e.g. for ADSL) and therefore the primary bottleneck in end to end communication. ANGEL can also be applied to other potential bottleneck links including the downstream consumer link and ISP's egress Internet connections.

The key feature of ANGEL is that it is completely transparent to higher layers in the protocol stack, and the QoS features will be provided without the need for the user or networked applications to have any knowledge of ANGEL. Existing end user applications will not need to be modified to take advanatage of ANGEL. The system will also be able to work in parallel with legacy CPE equipment that is not ANGEL enabled. Although ANGEL's main target is the CPE to ISP link, the system

From February 2007 and July 2010 this report was a confidential deliverable to the Smart Internet Technologies CRC. With permission it has now been released to the wider community.

could also be used to provide QoS in the whole ISP domain or even end to end QoS by using existing QoS techniques such as RSVP or Diffserv.

The document is structured as follows. Section II discusses our motivation for building ANGEL. In Section III we discuss different usage scenarios for ANGEL while Section IV lists the ANGEL requirements. Section V describes the ANGEL System Architecture while Section VI covers details in the protocol to be used between difference ANGEL modules.

II. Motivation

We are focussed on the CPE ISP link as this is the most likely bottleneck. However, other points of congestion are possible, perhaps in a poorly designed ISP network or at the the link between the if ISP and Internet core. We assume that all core Internet links are highly over-provisioned and there is basically no congestion on these links. We also assume that customers have high speed local networks (100Mbps or 1000Mbps) so there is no congestion inside customer networks.

Many access technologies provide assyemtric bandwidths. Often upload bandwidth is smaller by a factor of 4. Therefore the most likely bottleneck is the upload direction on the CPE ISP link.

The major problem being addressed by ANGEL is the bottleneck being caused by the 100Mbps local LAN squeezing traffic onto the slower upstream broadband link, which is typically orders of magnitude slower than the local LAN. For example, The serialization delay for a 1500 byte packet being sent upstream at 128kbps from the CPE is 93.75ms. This is significant in networking terms, where inter-packet arrival times tend to be sub 100ms. A packet arriving half way through the serialization process will be queued before being sent, and the queue will grow if more packets arrive during the serialization time. This can result in unpredictable packet queuing delays and jitter, caused by waiting in a queue that has heterogenous packets of different sizes in it.

Consider the following example: if a 100 byte packet gets stuck behind a 1500 byte packet that has just started getting serialized onto a 128kbps upstream link, the result is a 94ms wait for the 100 byte packet. Real-time/interactive services such as VoIP and online interactive gaming, tend to rely on small packets (typically well under 500 bytes) being sent at quick, regular intervals [2].

Now consider a real-time/interactive traffic flow sending 100 byte packets at regular intervals of 47ms, with serialization delay 6.25ms at 128kbps. One of the 100 byte packets gets stuck behind a 1500 byte packet that has just started getting serialized onto the upstream link. This leads to a 94ms wait for the 100 byte packet, in which time two more 100 byte packets join the queue behind the first. The first 100 byte packet begins being transmitted 94ms after it arrived in the queue, the second begins being transmitted 47ms + 6.25ms after it arrived in the queue, and the third begins being transmitted 6.25ms + 6.25ms after it joined the queue. The end result is that a traffic flow that should have consistent 47ms inter-packet arrival times (when not subjected to queuing delays), experiences inter-packet arrival times of 94ms, 53.25ms and 12.5ms. If the real-time traffic flow belonged to a VoIP conversation, for example, these large deviations from the expected inter-arrival times would result in degraded quality and an unpleasant phone conversation.

Of course, this example only considers the case of a single large data packet delaying the transmission of realtime traffic. Since non-realtime traffic typically uses the TCP protocol, large (1500 byte) packets are usually generated in bursts, filling the send queue with a number of large packets. This in turn will increase the serialization delay for real-time traffic even more.

Queuing at the ISP for the downstream link is typically not as much of a problem as it as at the CPE for the upstream link. This is because the downstream speeds tend to be much higher (typically at least 4 times greater) than upstream speeds for typical broadband access Internet plans. This means that packets are able to be sent downstream 4 times faster than they can upstream, and the serialization delay is 4 times shorter. The serialization delay for a 1500 byte packet being sent downstream at 512kbps from the ISP is 23.43ms. Even if a packet from a real-time/interactive flow gets queued behind a 1500 byte packet, it will only have to wait 24ms instead of 94ms as before. This observation validates the need to particularly focus on a solution that reduces upstream real-time/interactive network traffic queuing delays and jitter. However, the downstream link can still benefit from the ANGEL solution in the same way described for the upstream link above. Utilising ANGEL on the downstream link will be considered an optional part of the framework. This is to cater for situations where the costs of implementing ANGEL on the downstream link may outweigh the benefits provided.

III. Stakeholder Analysis and Deployment Scenarios

A. Stake Holder Analysis

There are four primary stakeholders that could benefit from the ANGEL system depending on how it is deployed: the broadband access (end) users, Internet service providers (ISPs), real-time/interactive services providers and broadband networking equipment manufacturers.

1) Broadband Access Users: Broadband access users will be most directly affected by the system, as they will perceive the end result in day-to-day activities. ANGEL will allow end users (especially those that are not technical) to appreciate the quality of service that makes using services like online multiplayer gaming, VoIP and real-time audio/video streaming reliable and user friendly.

2) Internet Service Providers: ISPs can stand to make potential gains from ANGEL in numerous possible ways. They can use it to create a range of new broadband plans charged at a premium to the consumer in exchange for better real-time/interactive traffic characteristics. They can establish links with various real-time/interactive services providers and charge them to prioritise traffic to their services and not to competitors. They can also simply use the product as a value added feature as part of their standard plans to entice more customers to join them.

3) Real-time/interactive services providers: Real-time/interactive services providers also stand to gain from ANGEL in a number of ways. As numbers of ANGEL users with broadband links now capable of delivering better real-time/interactive traffic characteristics begins to increase, the potential market for the services providers also increases, which translates into more customers. ANGEL also allows the possibility of services providers establishing ties with ANGEL enabled ISPs to only prioritise traffic to their service locations, thus further increasing the likelihood of more customers using their services.

4) Broadband Networking Equipment Manufacturers: Equipment manufacturers stand to gain from this technology as they can build CPE capable of interfacing with the ANGEL system. This means selling higher numbers of units to people connected to ISPs that support ANGEL, and encourages relationships between ANGEL enabled ISPs and the manufacturers as preferred product suppliers for new broadband users.

B. ANGEL Deployment Scenarios

The ANGEL system has been designed to be deployed in four main situations:

- Standard ANGEL enabled home site connecting to a fully ANGEL enabled ISP
- Intelligent ANGEL enabled home site connecting to a non-ANGEL enabled ISP
- Standard/fully ANGEL enabled enterprise connecting to a fully ANGEL enabled ISP
- Fully ANGEL enabled enterprise connecting to a non-ANGEL enabled ISP

The network traffic being generated in each of the scenarios will be assumed to be a heterogeneous mix of both realtime/interactive and non realtime/interactive flows. Applications such as interactive online multiplayer games and voice over IP make up the main realtime/interactive applications, and web, email and peer-to-peer (p2p) file sharing represent the main non realtime/interactive applications. It should also be noted that even in case of a single host being used, it is common that multiple applications are run at the same time e.g. p2p downloads/uploads are running while the user is playing games. This justifies the need for ANGEL even in situations where there is only one client side host being used to access the Internet.

In the following subsections, we will describe the 4 different scenarios in greater depth and the most pertinent permutations that exist within each scenario.

1) Standard ANGEL enabled home site connecting to a fully ANGEL enabled ISP: Figure 1 illustrates the typical usage scenario for the ANGEL system. An ISP network provides broadband access to customers via technologies such as ADSL, cable and/or wireless. The typical customer would be a small business, single home user or a small group of home users e.g. a family. This type of customer typically has a few hosts (often only one) connected to a LAN, all sharing a single broadband connection with a single public IP address. IP addresses from one of the private non-routable ranges are used in conjunction with network address translation (NAT) to allow multiple hosts attached to the client side network to access the Internet via the single public IP address. The technical knowledge of this user subset tends to be very limited, and ongoing manual network (re)configuration/maintenance to preserve quality of service for various applications in use at a given time is an unreasonable expectation to place on such customers.

In this scenario we assume that the ANGEL system is running in the ISP domain, monitoring all traffic

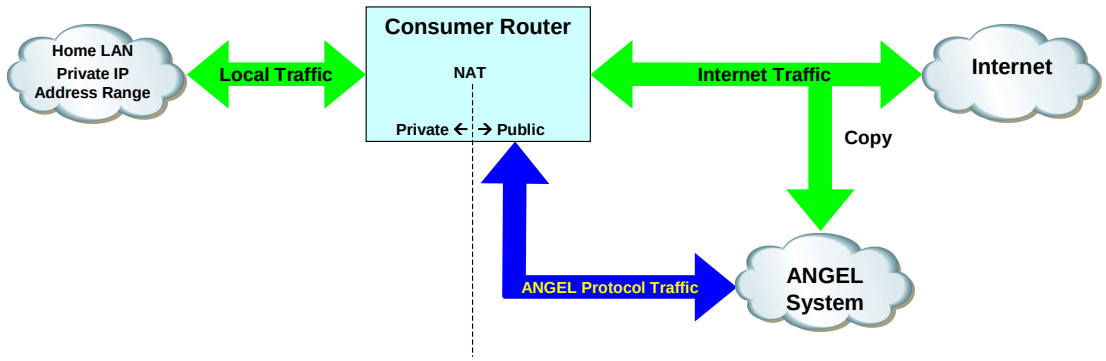

Fig. 1. Standard ANGEL enabled home site connecting to a fully ANGEL enabled ISP

originating from and terminating at the customer. When a user starts an interactive application, the ANGEL system will classify the realtime/interactive traffic and signal this information to the CPE (for upstream prioritisation) and possibly the client's default gateway router (for downstream prioritisation). The CPE (and possibly the default gateway router) will then be configured such that the classified realtime/interactive traffic will get priority over all non realtime/interactive traffic currently traversing the CPE/ISP link. The CPE will be responsible for correctly dealing with the presence of NAT, including correct translation of rules received from the ISP ANGEL system into prioritisation rules for the device's priority queuing subsystem. The ANGEL system remains completely transparent to the user during this process, as all commands will be generated by the ISP side ANGEL system and sent to the CPE, without any human intervention required. In addition to this automatic configuration, technically savvy customers way wish to create static prioritisation rules in the CPE device for applications/hosts known to produce realtime/interactive traffic e.g. a VoIP ATA, which will be catered for.

2) Intelligent ANGEL enabled home site connecting to a non-ANGEL enabled ISP: Figure 2 illustrates the usage of ANGEL in a home environment with a connection to a non ANGEL enabled ISP. The typical user and client side deployment situation is the same as the previous scenario. The technical knowledge of such customers would tend to be at an intermediate level, as this scenario requires them to configure and maintain more ANGEL functionality than in scenario 1. The CPE required for this scenario is also going to be more expensive than the standard CPE used in scenario 1 as a result of the additional functionality and complexity of the device. This entails the user has to make a conscious decision to follow the more expensive and technically difficult path of running an ANGEL system on their own.

In this scenario we assume that there is no ANGEL system running in the ISP domain, and the CPE is responsible for monitoring all traffic travelling over the CPE/ISP link. When a host on the local LAN starts an interactive application, the intelligent CPE will classify the realtime/interactive traffic and internally reconfigure itself to provide upstream prioritisation to the flow. It may in addition possibly signal the client's default gateway router using an existing QoS signalling mechanism to obtain downstream prioritisation for the same flow. If an explicit QoS signalling scheme is not supported by the ISP, a packet marking scheme such as DiffServ could be employed by the intelligent CPE instead.

This scenario has a number of advantages over the previous scenrio for the end user, in that no ANGEL signalling traffic is required on the network, and security/privacy issues are almost completely nulled. However, the increased cost of the CPE, additional management complexity and need to periodically update classification rules detracts from this scenario's end user appeal. Broadband networking equipment manufacturers stand to gain from the potentially higher prices of intelligent CPE devices for customers wishing to use ANGEL in non ANGEL enabled environments.

3) Standard/fully ANGEL enabled enterprise connecting to a fully ANGEL enabled ISP: Figure 3 illustrates the usage of ANGEL in an enterprise environment. An ISP network provides Internet access to customers via technologies such as ADSL, cable, wireless, leased lines, etc.. The typical customer would be medium to large enterprises, providing Internet access to hundreds and possibly thousands of employees. This type of customer typically has hundreds of hosts connected to a LAN, all sharing a single Internet connection. These customers are often large enough to have their own public IP subnets,

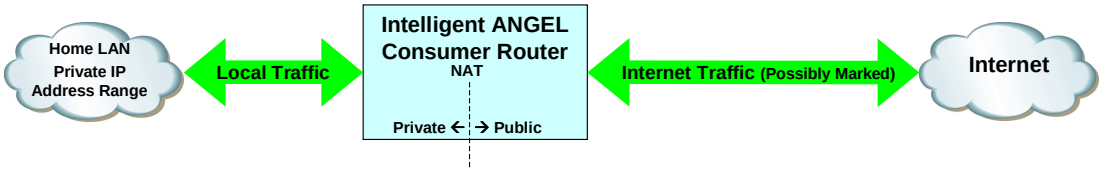

Fig. 2. Intelligent ANGEL enabled home site connecting to a non-ANGEL enabled ISP

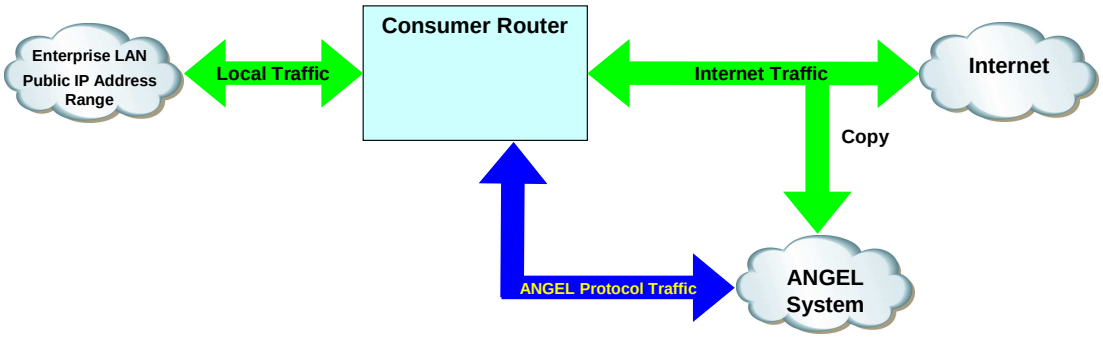

Fig. 3. Standard ANGEL enabled enterprise connecting to a fully ANGEL enabled ISP

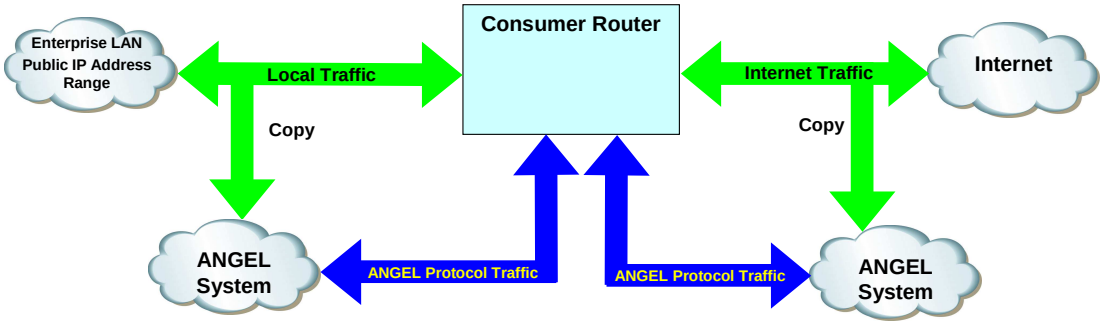

Fig. 4. Fully ANGEL enabled enterprise connecting to a fully ANGEL enabled ISP

which allows every host on their internal LAN to have its own public IP address, thus eliminating the need for NAT. The technical knowledge of such customers tends to be very complete, owing to the fact they have IT and/or network infrastructure departments managing their core network equipment. This in turn makes the possibility of a customer running their own internal ANGEL system feasible, as illustrated in Figure 4. This may be useful if the ISP is only monitoring some types of realtime/interactive traffic, or perhaps the ISP is unable to monitor some traffic flows because of security issues.

In this scenario we assume that the ANGEL system is running in the ISP domain, monitoring all traffic originating from and terminating at the customer. When a host starts an interactive application, the ANGEL system will classify the reatlime/interactive traffic and signal this information to the CPE (for upstream prioritisation) and possibly the client's default gateway router (for downstream prioritisation). The CPE (and possibly the default gateway router) will then be configured such that the classified realtime/interactive traffic will get priority over all non realtime/interactive traffic currently traversing the CPE/ISP link. In addition to this, the possibility exists for the customer to run their own private ANGEL system, which would be monitoring all traffic originating from and travelling to the customer's LAN from a client side network tap. The CPE would receive rules from both ANGEL systems, and where rules overlap, would be responsible for determining a precedence to decide which ANGEL system's rules are applied.

As with the first scenario, the ANGEL system remains completely transparent to the end user, negating the need for any human intervention. In addition to this automatic configuration, network administrators should be able to create static prioritisation rules in the CPE device for applications/hosts known to produce realtime/interactive

traffic e.g. a VoIP PABX, which will be catered for.

4) Fully ANGEL enabled enterprise connecting to a non-ANGEL enabled ISP: Figure 5 illustrates the usage of ANGEL in an enterprise environment with a connection to a non ANGEL enabled ISP. The typical customer and client side deployment situation is the same as the previous scenario, as is the assumed technical knowledge of such customers. This scenario is similar in principle to that shown in Figure 4, except that CPE configuration signalling is only received from the private ANGEL system.

In this scenario we assume that there is no ANGEL system running in the ISP domain, and the private ANGEL system is responsible for monitoring all traffic originating from and travelling over the CPE/ISP link. When a host starts an interactive application, the private ANGEL system will classify the realtime/interactive traffic and signal this information to the CPE (for upstream prioritisation) and possibly the client's default gateway router (for downstream prioritisation). The CPE (and possibly the default gateway router) will then be configured such that the classified realtime/interactive traffic will get priority over all non realtime/interactive traffic currently traversing the CPE/ISP link. The signalling between the private ANGEL system and ISP network would probably make use of an existing QoS signalling or packet marking mechanism e.g. RSVP, DiffServ, etc.. In the event that the enterprise is using private IP address space and NAT as in Figure 5, the CPE will be responsible for correctly dealing with the presence of NAT, including correct translation of rules received from the private ANGEL system into prioritisation rules for the device's priority queuing subsystem. The private ANGEL system would also possibly need to proxy ISP signalling through the CPE so that rule conditions could be rewritten by the CPE according to the NAT tables within the CPE.

IV. Requirements

This section specifies the requirements for the ANGEL system and the different building blocks. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

In the following we use the term user as a synonym for customer.

A. ANGEL System

1) Deployment: It MUST be possible to implement the ANGEL system in the deployment scenarios described in Section III without requiring significant changes to currently existing network infrastructures. Furthermore any additional functionality required by the ANGEL system in existing network components SHOULD be minimal.

2) Access Technology Independence: The ANGEL system MUST work with different network access technologies such as (A)DSL, cable modems or 802.11 WLAN. In order to cope with specific requirements for different access technologies it MAY be necessary for the ANGEL system to identify what access technology certain CPEs are using.

3) Scalability: The ANGEL system architecture MUST be scalable to support a very large number of subscribed users (tens of thousands) and a large number of simultaneously online users (several thousand).

If an administrative domain spans across a wide geographical area it MUST be possible to run multiple independent ANGEL systems in different locations of this network.

4) Failover: The ANGEL system SHOULD have failover mechanisms for all components to provide a reliable and predictable service.

5) Transparency: The ANGEL system MUST be transparent for existing end host network applications. This means no end host application needs to be aware of the ANGEL system and no changes are required to existing end host applications.

Furthermore, the ANGEL system MUST be transparent to end users. From a users viewpoint, no additional knowledge or configuration is required to benefit from the ANGEL system.

6) Non ANGEL enabled CPEs: The ANGEL system MUST ignore Customer Premises Equipment that is not ANGEL enabled and MUST NOT send any protocol messages to it. If the ANGEL system is used to provide a value-added service to some users the ANGEL system MUST NOT send any protocol messages to users that are not subscribed to that service.

B. ANGEL CPE

1) Priority Queuing: The ANGEL CPE MUST support priority queuing and MUST support at least two different priorities. It MAY support more than two priorities.

The ANGEL CPE MUST support dynamic updating and addition of prioritisation rules.

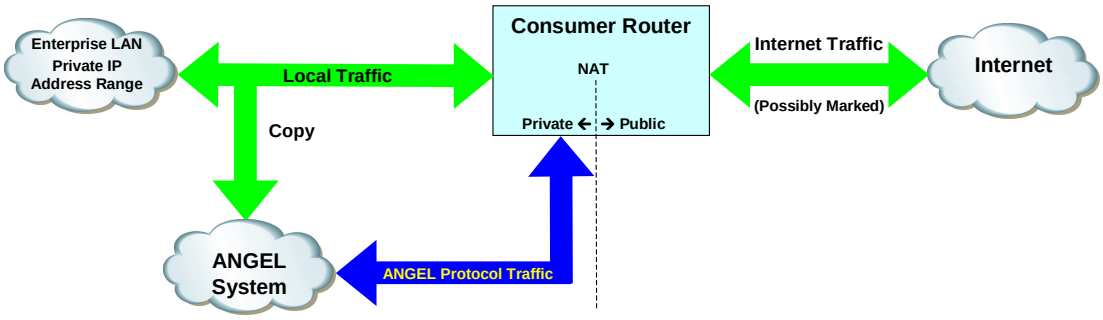

Fig. 5. Fully ANGEL enabled enterprise connecting to a non-ANGEL enabled ISP

The ANGEL system MUST provide means to detect rules that have not been used for some time, where use of a rule means that one or more packets have matched the rule. If the CPE has a per-rule packet counter or stores the timestamp of the last packet matching a rule, this information can be used to detect rules that have become inactive.

The ANGEL CPE MAY support packet marking (setting the Diffserv Code Point) to signal QoS classes further upstream.

2) User Management: The ANGEL CPE MUST have a configuration option that allows users to disable the ANGEL protocol.

The ANGEL CPE SHOULD support rules managed by the user. These user-defined rules MUST NOT be replaced or changed by rules generated by the ANGEL system and MUST NOT be timed out.

3) ANGEL Protocol: The ANGEL CPE MUST be capable of discovering if an ISP network is ANGEL enabled. If a network is determined not to be ANGEL enabled, the CPE device MUST NOT send any ANGEL messages.

The ANGEL CPE SHOULD send ANGEL protocol messages to the ANGEL client manager with high priority to minimise message loss and delay.

C. ANGEL Client Manager

1) Scalability and Failover: The ANGEL client manager MUST be scalable to a large number of simultaneously active ANGEL-enabled users (several thousand). The ANGEL client manager MUST implement failover mechanisms.

2) ANGEL Protocol: The ANGEL client manager SHOULD send ANGEL protocol messages to the CPE with high priority to minimise potential message loss and delay. Note that this requires prioritisation in the ISP domain at least on the access router.

D. ANGEL Flow Meter

1) Scalability and Failover: The ANGEL flow meter MUST support failover mechanisms. In case one flow meter fails, another flow meter MUST be able to take over within 30 seconds without the loss of any current flow data.

2) Flow Detection and Flow Timeout: The flow meter MUST be able to differentiate network traffic flows based on source and destination IP address. For transport protocols with port numbers the flow meter MUST also be able to use source and destination port for further differentiation. The flow meter MAY use further packet fields or meta information for flow differentiation.

The flow meter MUST implement timeouts to detect the end of a flow. The timeout value MUST be configurable. Flow meters MAY implement other techniques to detect the end of flows such as ending TCP flows after a teardown handshake is observed.

3) Flow Information Export: The flow meter MUST export interim flow records in regular intervals (push mode). The export interval MUST be configurable. Flow meters MAY be able to export data if requested (pull mode).

The ANGEL flow meter SHOULD support short term batching of flow records to minimise network overhead. However, the time a flow record is delayed MUST be small enough to not significantly delay enabling QoS for real-time flows.

4) Flow Attributes: The type and number of flow attributes or features that are computed for each flow and exported to the classifier SHOULD be configurable.

5) Performance: The flow meter MUST be able to handle 10Mbps and SHOULD be able to handle 100Mbps line rate traffic. The flow meter MAY support 1Gbps line rate traffic. The ANGEL flow meter MUST support metering of thousands of concurrent flows. The ANGEL flow meter MAY support IP address filtering

to ignore the packets of non ANGEL-enabled users for enhanced performance.

E. ANGEL Classifier

1) Accuracy and Classification Time: The ANGEL classifier MUST provide reasonably high accuracy when differentiating real-time from non real-time flows ($\geq$95 percent). Besides outputting the predicted class and priority for each flow, the ANGEL classifier SHOULD also compute a measure of how correct the prediction is.

The classifier MUST provide the required accuracy a short time after the flow has started and maintain the accuracy during the flow's lifetime. The time in which an accurate prediction is needed depends on the application e.g. for games this may be longer as joining a game server usually takes some time but for Voice over IP (VoIP) it may be shorter.

2) Performance: The ANGEL classifier MUST be fast enough to classify thousands of flows per second. The ANGEL classifier MAY support IP address filtering to ignore any flow information for non ANGEL-enabled CPEs that has been generated by flow meters incapable of filtering.

It SHOULD NOT take an unreasonably long time to update/change the classifier. For example, some machine learning algorithms have very long training times that would prevent reacting quickly to changes in the network traffic such as new applications.

3) Flexibility: The ANGEL classifier MUST be flexible regarding the classification technique and features used. The ANGEL classifier MAY use different techniques and or features simultaneously for different flows. The ANGEL classifier MAY also use different classification techniques and different features simultaneously to determine the class of a single flow.

The ANGEL classifier SHOULD be updatable without interrupting current operation. As new real-time applications are developed it becomes necessary to update the classifier with information required to accurately classify them. Similarly applications may need to be removed if their profiles have changed. The classifier SHOULD record statistics on what applications are used in the network.

4) Data Export: The ANGEL classifier MUST keep some state per flow. At the very least it MUST keep the previous class prediction to detect changes. It MUST NOT export information to the ANGEL client manager where the class prediction has not changed.

The ANGEL classifier MAY keep more previous class predictions, for example if that information can be used to increase classification accuracy.

The ANGEL classifier SHOULD support short term batching of flow predictions to minimise network overhead. However the time a flow prediction is delayed MUST be small enough to not significantly delay enabling QoS for real-time flows.

F. ANGEL QoS Signaling Protocol

The ANGEL signalling protocol is responsible for signalling the QoS rules to the CPE and optionally the ISP access router.

1) Based on Standard Protocol: The ANGEL protocol MUST run over IPv4 and IPv6. It MUST run over one the standard IETF transport protocols: UDP, TCP or SCTP. It SHOULD be based on one of the standard IETF protocols such as SNMP.

2) Flexibility and Extensibility: The ANGEL protocol MUST be flexible and extensible to allow adding new methods and new data fields. It MUST be designed in a way that there can be different versions of the protocol.

The ANGEL protocol MUST convey QoS rules in a generic way so they can be translated into different native representations used on different CPE devices.

3) Efficiency: The ANGEL protocol MUST be simple to allow for easy implementation in current and next generation CPEs and to minimise the resources (CPU, RAM) needed on the CPE.

Because the ANGEL protocol runs over the low bandwidth CPE ISP link, it MUST be very efficient in terms of bandwidth. In the direction from the CPE to the ISP especially, protocol overhead MUST be very low because the capacity can be very small, for example on asymmetric links such as ADSL.

4) Reliability: The protocol MUST have mechanisms to ensure reliable message transfer such as acknowledgments, retransmissions and timeouts. However, because many ANGEL messages require very timely delivery the maximum number of retransmissions SHOULD be fairly small. The number of retransmissions and the time between retransmissions MUST be configurable.

5) Security Considerations: The ANGEL protocol REQUIRES mutual authentication between ANGEL server and CPE device. The ANGEL CPE MUST be able to authenticate rule updates sent by the ANGEL server. The ANGEL server MUST be able to authenticate a registering CPE. If IP address spoofing can be prevented IP addresses provide sufficient authentication. However, if IP address spoofing can not be prevented the ANGEL protocol is vulnerable to insertion attacks.

Rule update messages SHOULD NOT be sent to a wrong CPE. This could happen for example if a CPE disconnects and a different CPE connects obtaining the same IP address and there are still rule updates for the old CPE in the ANGEL system. However, because the ANGEL system only sends rule updates to the CPE for new flows or if the class prediction changes shortly after these activities are detected rule update messages will not stay longer in the system than it takes for a CPE to connect.

The ANGEL protocol MUST NOT introduce new possibilities for Denial of Service (DoS) attacks. A new effective DoS attack can only be achieved if there is some packet multiplication effect i.e. for each packet the attacker sends multiple packets arrive at the target. An attacker could cause an ANGEL server to send rule updates to a victim by sending packets into the network with the victims IP address as source. If the victim CPE is ANGEL enabled these packets would then trigger rule update messages that are sent to the victim. However, in the worst and very unlikely case there would only be one rule update message for each packet of the attacker. Packet multiplication is impossible and the rule update messages are very small so the number of bytes does not significantly increase either. If the ISP network is secured against DoS attacks (e.g. if rate limiting is used) the ANGEL system and ANGEL enabled CPEs would also be protected.

G. ANGEL Internal Communication Protocols

Any internal communication between different ANGEL components MUST be secured against attacks. However, we can assume that no eavesdropping/wiretapping is possible because all components are run inside the ISP premises which are protected by physical security. We can also assume that authentication can be provided solely based on IP addresses by allocating address ranges to ANGEL components that are excluded from use by user traffic and preventing IP address spoofing (e.g. by proper ingress filtering). In the extreme case all ANGEL internal communication could be run over a separate internal network. Note that this means a provider that is distributed across different geographical locations and is using another provider to carry the traffic between these locations MUST run a separate ANGEL system at each location.

V. ANGEL System Architecture

Given the provided scenarios, assumptions and requirements of the network infrastructure under which we expect ANGEL to operate, we have designed an architecture for the ANGEL system. In this section we outline this architecture, the means by which the requirements are met, and the operations of each seperate component of the ANGEL system.

The ANGEL architecture is summarised in Figure 6. In simple terms we can consider the case of a single consumer premises device (router/modem) interacting with the ANGEL ISP side components in the network using a form of a feedback control loop. In this instance a copy of all traffic that is generated within the home network is monitored by the ANGEL ISP Side Components which will then classify real-time flows and provide prioritisation rules back to the consumer premises router via means of the ANGEL protocol.

The router device is then responsible for interpreting these rules in order to provide the best QoS possible given available resources. These rules will be continuously updated based on user generated network traffic and therefore prioritised service will only be provided for current real-time flows. Further we expect the router to be able to handle ANGEL protocol messages from more than one ANGEL Monitoring System, allowing the user to provide their own traffic classification system and rules in addition to that provided by the ISP.

We also recognise the scenario whereby routers owned by the ISP can also be ANGEL enabled to prioritise traffic flows. In this instance we consider likely routers to include the downlink router to the customer premises and the ISP Internet gateway router as locations where traffic prioritisation may be beneficial to ISP operators. In this instance these routers will also be running the ANGEL CPE Management module to allow communications with the ANGEL ISP Components deployed in the network.

The ANGEL ISP Side has a number of requirements that must be considered when in the context of the design architecture:

- A CPE Router must be able to communicate with multiple ANGEL ISP Side components - perhaps a user wishes to configure the server side components within their own network
- The system components must be able to be deployed in a modular scenario such that the system is scalable to support large ISP networks. This involves consideration of issues such as traffic volume, failover redundancy and ease of repair of damaged components.
- The ANGEL system component modules must be able to be deployed either as a whole - or a subsection of components - on a single hardware platform

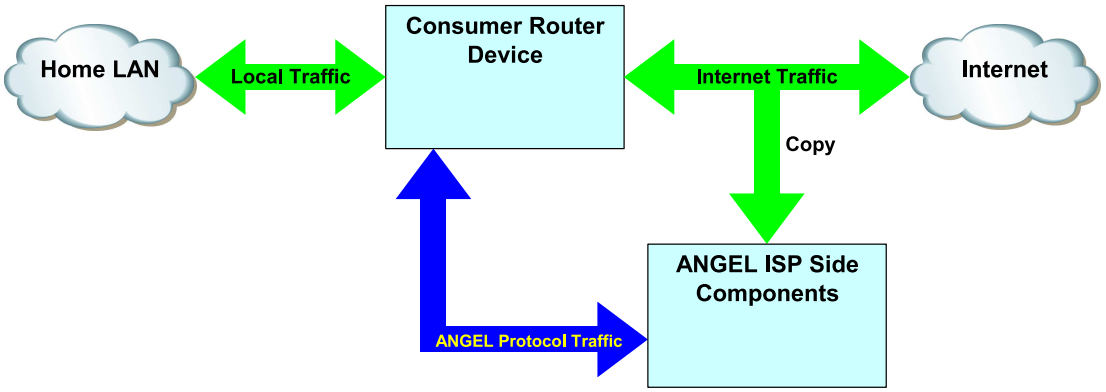

Fig. 6. ANGEL System Block Diagram

to reduce costs and requirements for smaller network configurations.

The ANGEL ISP side components are shown in Figure 7 and consist of five distinct blocks. It is anticipated that these blocks can be implemented as a single system that operates on one physical platform, or distributed amongst multiple physical platforms in a distributed fashion. Further, scalability issues may require a single block be distributed amongst multiple platforms to provide not only improved service but also failsafe operation.

A summary of the system design shows that apart from the ANGEL Database - which contains system wide configuration information and the ANGEL Configuration Manager - through which the system can be configured, the architecture is implemented as a waterfall type model. In this case each block of the system architecture performs some processing and then passes information down to the next block in the chain. As described in further detail in the following sections, no block is required to pass information back up the chain to preceeding blocks in the data processing path.

The three primary blocks consist of:

- **ANGEL Flow Meter -** Extracts packet statistics for each network flow that is tapped within the network and passes the feature sets to the Flow Classifier.
- **ANGEL Flow Classifier -** Analyses the packet statistics and classifies traffic into different classes based on the estimated application being used. Signals the Client Manager when a flow state changes so that this information can be used to develop prioritisation rules to send to ANGEL enabled routers.
- **ANGEL Client Manager -** Manages communications with the connected ANGEL enabled routers. Generates traffic prioritisation rules and communicates these rules to the routers for them to implement.

The physical architecture allows for more than one Flow Meter to be deployed within the ANGEL system but requires a single Flow Classifier and a single Client Manager Module. These requirements allow the system to properly maintain state. Scalability and failover can be managed through the use of a clustered system to implement the Flow Classifier and Client Manager modules, or the running of two parallel ANGEL Monitoring Systems within a network. Further details for failover support within ANGEL are covered in a later section.

A. *ANGEL DataBase*

The ANGEL Database server is accessible by all other modules within the ANGEL Monitoring System. This design feature allows us to place system configuration information within the Database, and allows individual modules to query the database for any information required to properly function within the entire ANGEL System.

The database is also responsible for managing details of all currently registered ANGEL-enabled customers, in particular the IP addresses being managed by their modem/router devices. The database can be used by various modules within the ANGEL Monitoring System to filter traffic from being processed and by the Client Manager to properly communicate with the registered customers.

The database can be implemented within the same hardware platform as other ANGEL Monitoring System components or as an external database which is accessible by these components. We expect that existing database tools can be used to minimise development effort.

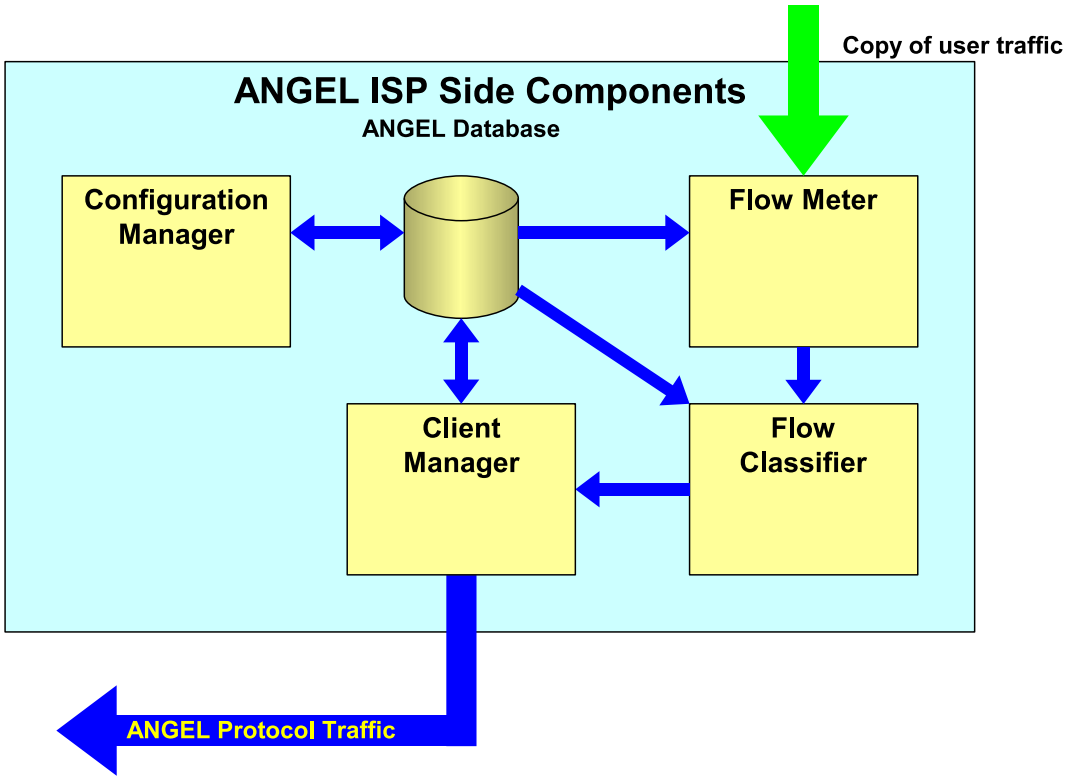

Fig. 7. ANGEL ISP Side Components Block Diagram

B. ANGEL Flow Meter

An ANGEL System can contain numerous ANGEL Flow Meters. The number and configuration of each Flow Meter is dependent on the network design and configuration, the number of current or potential ANGEL users, and the ammount of network traffic flowing through the network at Flow Meter locations.

The primary tasks of a Flow Meter are:

- Monitor a copy of network traffic at a particular location
- Filter the traffic into individual network flows
- Extract the packet statistics for each flow on a periodic basis
- Forward the extracted per-flow data to the ANGEL Flow Classifier

Key design restrictions for all aspects of the ANGEL Flow Meter are:

- Two or more network connections, one to communicate with the Flow Classifier and ANGEL Database, the other(s) to receive a copy of all network traffic from the tap(s)
- Configured with (either locally or from the ANGEL Database):
  - IP Address of the ANGEL Database
  - IP Address of the ANGEL Flow Classifier to communicate the calculated feature sets to
  - IP network connection details of ANGEL Customers
  - Flow timeout values
- Optionally filter captured packets to only consider current ANGEL enabled users - packets that match the active list of IP addresses to be monitored are kept while other packets are immediately discarded. This is particularly useful for a small proportion of ANGEL users amongst the network user base as less Flow Meters and hardware can be deployed. In situations where a large proportion of users are ANGEL enabled, it may be more efficient to process all traffic including non-ANGEL enabled users.
- Packet statistics are always communicated to a single Flow Classifier
- Statistics for a single packet must not span multiple packets when communicating with the Flow Clas-

sifier
- Packet statistics for multiple packets/flows can be sent in a single packet to the Flow Classifier
- If a currently active flow becomes inactive for the configured timeout period, resources used to hold the flow state and feature information will be released and the Flow Meter will signal the Flow Classifier that the flow is terminated.

1) Flow Meter Components: The sub-module architecture of the Flow Meter Module is highlighted in Figure 8. Processing is predominately performed in a linear fashion with a Control Module configuring the processing modules based on configuration information in the external ANGEL Database. Each module has the following tasks:

- **Network Tap** - Capture traffic from the network port. This could be implemented using platform independent tools such as libpcap [2] or via direct (OS dependent) means (eg. BPF filters on BSD based systems) [3]. The module should capture all network traffic and pass captured IP packets to the next module in the chain.
- **Filter** - This module is optional. If present the purpose of this module is to filter out and discard packets for which ANGEL monitoring and analysis is not enabled. The external ANGEL Database stores the IP addresses of all currently registered ANGEL CPE devices, these IP addresses are compared against source and destination IP addresses in captured packets to determine if the packet should be filtered or not.
- **Packet Classifier** - Sort the captured packets into distinct network flows. Flows are determined by matching IP Address/Port Number/Transport Protocol tuples. Packets should be tagged for which flow they belong to and passed to the Feature Extractor module.
- **Feature Extractor** - Extract individual packet statistics on a per-flow basis and regularly communicate results to the ANGEL Flow Classifier. Manages timeouts on flow information to release resources used to store flow state information.
- **Control Module** - Regulary read the external ANGEL Database for current configuration information and use it to reprogram the processing modules to perform their duties. In particular inform the (optional) Filter of the IP addresses to filter and the Feature Extractor of the feature set to calculate and the contact details of the Flow Classifier.

C. ANGEL Flow Classifier

An ANGEL System may only consist of a single Flow Classifier. Scalability of the Flow Classifier to handle larger processing loads should be handled either through the implementation of multiple parallel ANGEL Monitoring Systems or through the use of load balancing within a cluster of machines.

The primary tasks of the Flow Classifier are:

- Classify flows based on the statistics provided by any Flow Meters within the system
- If a new flow is classified as requiring prioritisation or changes its prioritisation level, signal the Client Manager with this information
- Purge stored flow information when that flow terminates

For the Flow Classifier to function properly, it must have access to the previous flow classification - and possibly the recent classification history - for comparison against the current classification. It would be possible to have multiple Flow Classifiers within a single ANGEL System as long as the above condition is met. Our design calls for the Flow Classifier to be implemented as a single unit to simplify implementation and to make the implementation of the Flow Classifier transparent to the Flow Meters. Should a system be designed with multiple Classifiers, the flow classification history must be common amongst all Classifiers.

The key design features of the Flow Classifier are:

- Only one Flow Classifier for an ANGEL Monitoring System
- Configured with (either locally or from the ANGEL Database):
  - IP Address of the ANGEL Database
  - IP Address of the ANGEL Client Manager to communicate with
  - IP network connection details of ANGEL Customers
  - Flow timeout values
- Optionally filter flows to only consider current ANGEL enabled users - flows that match the active list of IP addresses to be monitored are analysed while other flows are immediately discarded. This can be useful to minimise the load on the Flow Classifier if it has not been over-provisioned. In situations where the Flow Classifier has been adequetly dimensioned to support the entire network, it may be more efficient to process all flows including those of non-ANGEL enabled users

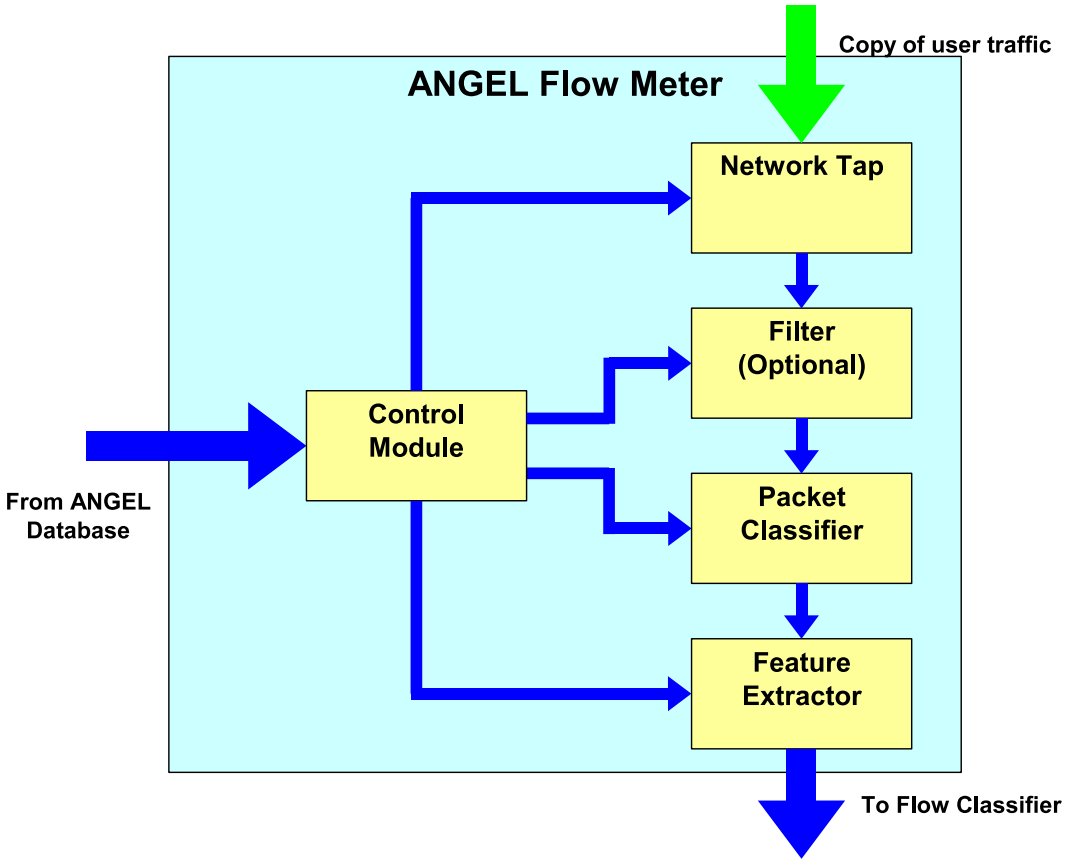

Fig. 8. ANGEL Flow Meter Sub-System Block Diagram

- Store state information for currently active flows such that a a change in state can be detected
- If any of the following classifications are made, signal the ANGEL Client Manager with the IP address of the client, the flow identification details (eg. IP Address/Port Number/Protocol tuples), the flow classification and the prioritisation level
  - A new flow is categorised as real-time
  - A previously non-realtime flow is categorised as realtime
  - An existing real-time flow is classified as non-realtime
- Flow termination is determined and signalled by a Flow Meter, if this occurs, all resources maintaining state for that flow are released. The Client Manager need not be notified as any prioritisation rules in the router devices will eventually timeout.

1) Flow Classifier Components: The sub-module architecture of the Flow Classifier Module is highlighted in Figure 9. Processing is predominantly performed in a linear fashion with a Control Module configuring the processing modules based on configuration information in the external ANGEL Database and an internal Flow Database storing Flow Classification state. Each module has the following tasks:

- **Filter** - This module is optional. If present the purpose of this module is to filter out and discard flows for which ANGEL monitoring and analysis is not enabled. The external ANGEL Database stores the IP addresses managed by all ANGEL connected CPE devices, these IP addresses are compared against source and destination IP addresses in provided flows to determine if the flow should be filtered or not.
- **Classifier** - Classify the flows based on the provided

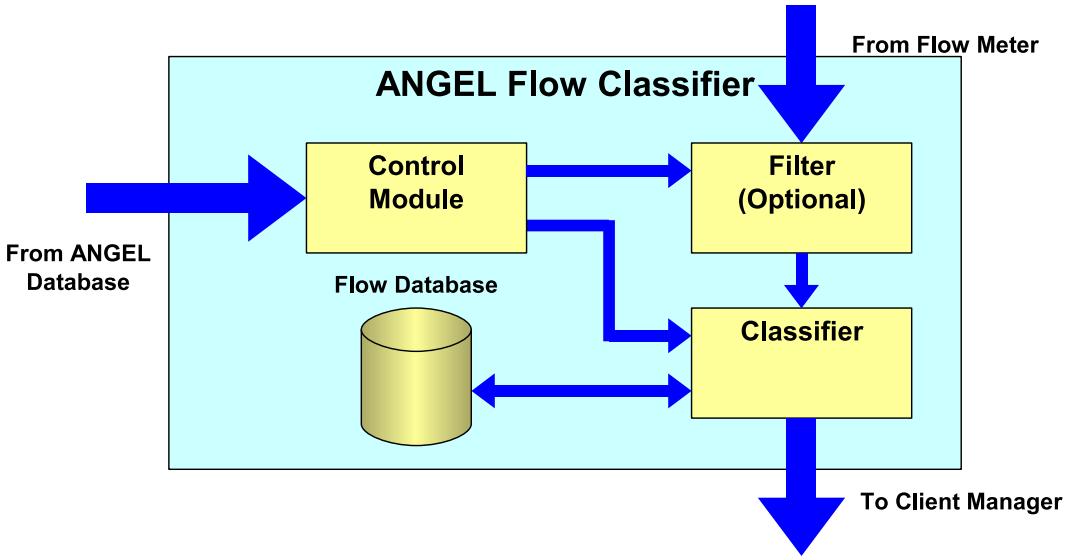

Fig. 9. ANGEL Flow Classifier Sub-System Block Diagram

per-flow statistics and state information stored for the given flow in the Internal Flow Database. Update the Flow Database as required due to changes in classification and/or determination that a flow has ceased. As required forward changes of flow state to the Client Manager for communication of priority levels to connected ANGEL Routers. The design of the Classifier is deliberately left open to allow for future advances in flow classification techniques to be applied in the ANGEL System.

- **Flow Database** - Stores state information on all classified flows. This information is used in conjunction with new feature sets to properly classify and prioritise flows.
- **Control Module** - Regulary read the external ANGEL Database for current configuration information and use it to reprogram the processing modules to perform their task. In particular inform the (optional) Filter of the IP addresses to filter and the Classifier of the contact details of the Client Manager.

D. ANGEL Client Manager

An ANGEL System may only consist of a single Client Manager. Scalability of the Client Manager to handle increased customers should be handled either through the implementation of multiple parallel ANGEL Monitoring Systems or through the use of load balancing within a cluster of machines.

The primary tasks of the Client Manager are:

- Accept connections to the ANGEL System from ANGEL enabled consumer routers
- Validate ANGEL capable routers for access to the ANGEL System
- Maintain the contents of the ANGEL Database - adding new consumer router details when the Customer Premises router registers with the ANGEL System and removing entries when routers leave the system
- Signal all appropriate routers upon detection of flow classification change by the Flow Classifiers

The Client Manager is responsible for maintaining the state of currently connected users within the ANGEL Database as well as communicating flow prioritisation information (determined by the Flow Classifer) to any routers that may prioritise these flows.

Upon receiving flow prioritisation information, the Client Manager should extract router details from the database and ensure that the rule has reached the corresponding router.

The key design features of the Client Manager are:

- Configured with (either locally or from the ANGEL Database):
  - IP Address of the ANGEL Database

- Manage communications with any connected ANGEL enabled routers
- Optionally support validation of ANGEL enabled routers if ANGEL is to be provided as a paid feature by the ISP.
- Remove users/routers from the ANGEL database when the router leaves the system or changes configuration to disable ANGEL.
- Communicate ANGEL prioritisation information to consumer routers (and ISP routers if appropriate) using a reliable protocol to ensure that the information reaches its destination.
- Must filter flows to only communicate prioritisation information to current ANGEL enabled users - no ANGEL packets should be sent to non ANGEL-capable devices or routers which are configured with ANGEL disabled.

1) Client Manager Components: The sub-module architecture of the Client Manager Module is highlighted in Figure 10. Data is independently processed by two seperate modules. A smaller timeout module is used to trigger events that indicate a consumer premises router has disengaged from the ANGEL System. The other two modules have the following tasks:

- **CPE Registration** - Manage the registration process of individual Consumer Premises Router devices. The module should manage the proper registration procedure with the individual routers and update user details in the external ANGEL Database. A re-registration by the routers SHOULD occur within the configured timeout period. The Timeout Module is used to trigger this period and to determine when to remove user details from the ANGEL Database.
- **Prioritisation Information Forwarder** - Forward any flow prioritisation information received from the Flow Classifier to the respective ANGEL enabled router devices. Contact information for the router devices (eg. IP Address, Keys) are obtained from the external ANGEL Database. The module MUST await acknowledgement of reception of the flow prioritisation information and possibly retransmit.

E. ANGEL Configuration Manager

The ANGEL Configuration Manager is a system whereby configuration settings of the ANGEL Monitoring System can be modified. The module is a front-end allowing direct access to change details in the ANGEL Database and other ANGEL configuration settings.

F. ANGEL Customer Premises Router

An ANGEL System will involve the use of numerous ANGEL enabled Consumer Premises Router devices - ideally the broadband access router (ADSL Modem) employed by the customer. These devices typically have minimal processing power and are designed for mass reproduction. To this extent, the implementation of ANGEL on these devices must be kept simple. This has already been touched on previously in explaining why the majority of the effort in traffic flow classification should be performed by the ISP or elsewhere in the network, leaving the router itself to simply accept ANGEL prioritisation rule updates and to implement these rules within the router to provide QoS for the specified flows.

The structure of components within the Consumer Premises Router is expanded in Figure 11. We envisage the majority of the existing functionality provided by such equipment to remain unchanged. The router would require the addition of one further module - ANGEL Protocol Manager - to manage communications between the ANGEL System and the Consumer Premises Router. These communications would involve the use of the ANGEL Protocol (see Section VI) to communicate traffic prioritisation information to the router. The Protocol Manager Module will be responsible for parsing and verifying these messages before using any available QoS functionality on the router to provide improved QoS for the specified flows.

It is anticipated that existing Consumer Broadband Access Technology has the capability to implement priority queueing, and that if not currently available these forms of QoS functionality will be common in next generation consumer premises equipment. Already UbiComm have released their StreamEngine technology [4] which attempts to detect real-time flows and then implement priority queing on these flows. This indicates that the capability to implement stream prioritisation has already been developed. The goal of ANGEL is to standardise the process of communicating prioritisation information to the router.

The primary tasks of the ANGEL Consumer Premises Router are:

- Initially communicate with the ANGEL Client Manager and register its IP address details
- Re-register with the ANGEL Client Manager at configured intervals to ensure continued reception of ANGEL flow prioritisation information
- Process information sent by the ANGEL Client Manager containing flows to prioritise and their

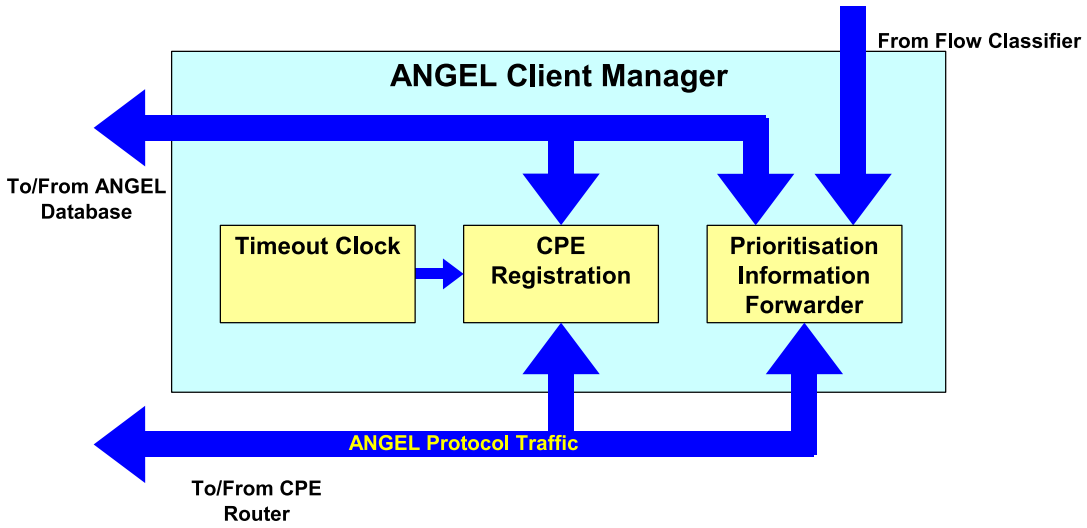

Fig. 10. ANGEL Client Manager Sub-System Block Diagram

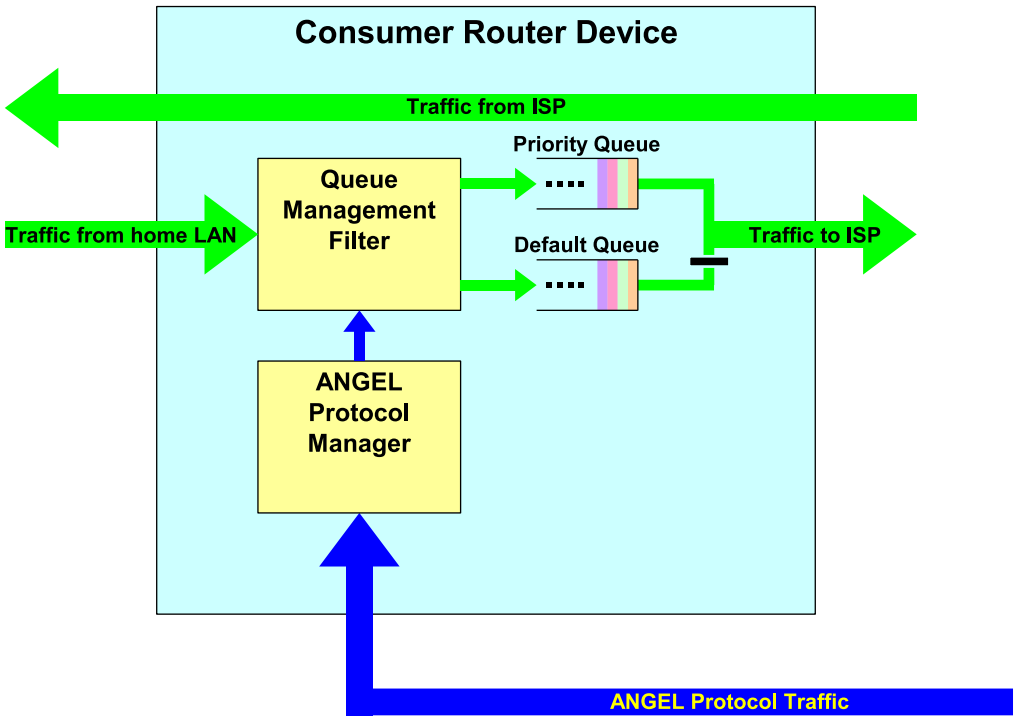

Fig. 11. ANGEL Consumer Premises Router Block Diagram

prioritisation levels and to provide the best level of support possible on the device to support these requests
- Validate that any prioritisation requests come from configured ANGEL Systems only
- Remove flow prioritisation of flows when no packets for a prioritised flow have been witnessed within a timeout period

The key aspect to any implementation on the Consumer Premises Router is to minimise processing and memory requirements. As per previous discussions, the majority of the flow detection and classification should be performed externally to the ANGEL enabled routers.

The key design features are:

- Configurable whether to use ANGEL or not
- Configurable addresses of ANGEL Systems to register with - more than one ANGEL System can provide scope for the end user to provide their own flow classification in conjunction with an ISP based system
- Configurable use of insecure (signed only) or secure (encrypted) communications with the ANGEL System
- Manage communications with any configured ANGEL Systems
- Access to any features on the underlying device to manage Quality-of-Service (not all the following may be available on any given ANGEL based system):
  - Priority Queueing
  - Multiple Virtual Circuits (in the case of ATM based ADSL devices)
  - Link-layer packet interleaving
- Direct conversion of ANGEL provided prioritisation information to the underlying flow prioritisation features on the device

G. Scalability and Failover Support

Scalability and support for system failure has been considered in the design of the ANGEL Architecture and in particular in the possible implementation of each individual module. Obviously a choice to implement all modules on one physical platform does not provide any scalability or hardware failure protection. In this section we discuss how these features are supported within the different modules of the ANGEL Monitoring System, in particular the three primary modules (Flow Meter, Flow Classifier and Client Manager) where these issues are most important.

ANGEL Database

We expect the ANGEL Database to be built using existing database tools. We acknowledge that there exist many database systems that provide scalable and failsafe database operation that can be chosen to meet the specific needs of any particular ANGEL Monitoring System implementation.

ANGEL Flow Meter

In the case of the Flow Meter, the base architecture allows for scalability through the use of multiple distributed Flow Meters amongst the network being monitored. In this case scalability is managed by distributing as many Flow Meters as is required to manage the network traffic being generated by the end users and located wherever necessary to most efficiently process that traffic.

This implementation does not however allow for continued operation in the event of failure of a Flow Meter. To allow for failover implementation, we allow for a group of machines to be implemented as an ANGEL Flow Meter Cluster, a cluster can consist of one or more individual machines working together to perform the task of the Flow Meter.

If the Flow Meter is to be implemented as a cluster, then the design details are:

- CARP [5] (or a similar protocol) must be used to handle cluster management. One machine in the cluster is to be elected as the master while the remaining machines are labelled as slaves.
- All machines in the cluster receive all network traffic as captured at the network tap.
- All machines in the cluster caculate flow features as configured.
- Only the currently elected master communicates flow features to the ANGEL Flow Classifier
- In the case of failure of the master machine, a slave is elected to be the new master and resumes forwarding of information to the Flow Classifier

An ANGEL Flow Meter Cluster does not share the processing load amongst many machines, the presence of extra machines is purely to allow system operation to continue if failure occurs. This design choice was made to ensure that all packets from a particular network flow are handled by the same machine within the same Flow Meter Cluster.

While the design allows for many machines within a Flow Meter Cluster, we anticipate that a cluster will consist of no more than two machines - when one machine fails the backup will continue processing traffic until the primary machine can be replaced.

ANGEL Flow Classifier

The base ANGEL Architecture requires that flow state be maintained across all Flow Classifiers in the system. As such, the Classifier can be implemented in either of the following ways:

- As a standalone machine performing the Flow Classification Task
- As a virtual machine implemented as a cluster providing load balancing and failover capabilities with all machines in the cluster and sharing a virtual shared process system
- As multiple Flow Classifiers within the system. Each Flow Meter would need to be configured with all the Flow Classifiers to which they can communicate with and flow classification history must be shared amongst all the Flow Classifiers

Scalability and failover operation can be managed through the use of either of the last two options. The third option is more complex and difficult to manage. If we implement scalability through the use of a clustered Flow Classifier, then the design details are:

- The Classifier must act as a virtual machine with one publically visible IP address
- Use existing tools to manage the cluster and to balance incoming flow features from the Flow Meters - possibly a cluster capable OS
- A local shared database amongst the machines in the cluster containing state information on all current flows, allowing any machine in the cluster to process a feature set on any given flow
- All machines within a cluster must be configured to calculate the same feature sets

In the case of a Flow Classifier Cluster, processing load is distributed amongst all active machines in the cluster, making full use of the available CPU resources. The shared local database ensures that any machine in the cluster can process any feature set from any flow. We expect that the use of existing tools to provide cluster management and load sharing capabilities should ease the implementation details of a clustered Flow Classifier.

ANGEL Client Manager

The base architecture requires a single Client Manager to be deployed within an ANGEL system. As per the Flow Classifier, scalability and failover operation can be managed through the use of a virtual machine implementing a clustered Client Manager.

The Client Manager keeps track of currently connected customers through the use of the external database, upon receiving notifications from the Flow Classifier, the CLient Manager needs to consult the database and forward appropriate ANGEL messages to registered user devices.

VI. Protocol Details

In this section we first describe a number of use cases that require information exchange between different ANGEL components. For each use case we then develop the necessary protocol operations (if any). We only describe the different message types and the information that is sent in each message but we do not describe a particular encoding of the protocol messages. A full protocol specification including encoding is left for a future document.

A. Use Cases

1) ANGEL CPE Connects: The ANGEL enabled CPE device connects to the ISP network e.g. when it is booted. As soon as IP connectivity is established the ANGEL CPE performs a Registration message exchange with the ANGEL client manager in the ISP domain to register itself as an active ANGEL enabled CPE.

2) ANGEL CPE Disconnects: The ANGEL CPE disconnects e.g. when it is turned off. No protocol messages are sent in this case. Flow state in the flow meters and flow classifiers will time out after no packets from the CPE are observed within a flow timeout. Session state monitored by the ANGEL client manager will time out when no re-registration is performed within the session timeout. All rule state in the CPE device will be removed if the CPE is powered down.

3) New Flow Starts: An end host behind an ANGEL enabled CPE starts a new application flow. The flow meter will detect the new flow, create a flow entry and start computing flow attributes. Flow Data messages are then sent to the flow classifier. The flow classifier will predict the class of the flow based on the flow attributes and store the information. If the flow is classified as real-time the flow classifier will send a Class Update message to the Client Manager. The Client Manager will then send a Rule Update message to the CPE. The CPE acknowledges a received rule update with a Rule Update Acknowledgment. If the flow is classified as non-realtime no information is sent from the flow classifier to the ANGEL client manager.

During a flow's lifetime the flow meter will export flow attributes in regular time intervals to the flow classifier and each time the classifier will re-classify the flow. If the predicted class is different from the previous prediction the flow classifier will send a Class Update

message to the ANGEL client manager which will send a Rule Update message to the CPE device.

4) Flow Stops: An end host behind an ANGEL enabled CPE stops an application flow. The flow meter will detect the end of the flow after no further packets are observed within a flow timeout. A Flow Data message is sent to the flow classifier and flow state within the flow meter is deleted. The message includes an indication that the flow has ended and this is the final Flow Data message. The flow classifier will usually delete any state for this flow and will not send further Flow Class Updates to the ANGEL client manager. However, if necessary for e.g. auditing purposes the flow classifier might use the final flow information to classify the flow again and store the final classification in its database. Regardless of the predicted class no information will be passed on to the ANGEL server.

5) Flow Class Change: An end host behind an ANGEL enabled CPE has changed the class of a flow. This can happen for example if one application sends both realtime and non-realtime flows or if an application has quit and a new application sends traffic from the same port and to the same destination. In this case the flow classifier will detect the class change and send a Class Update message to the ANGEL client manager. This will trigger the Client Manager to send a Rule Update message to the CPE device.

6) ANGEL Components Start/Stop: An ANGEL component in the ISP domain is started or stopped. When an ANGEL component is started it sends a Configuration Request message to the ANGEL DB. It then receives a Configuration Response message that contains all necessary configuration needed during the operation.

As long as a component is running it keeps updating its configuration in regular time intervals.

If an ANGEL component is stopped or crashes no messages are sent (see Section V on redundancy/failover of components).

B. Protocol Operations

1) Registration: Before an ANGEL System will begin sending information to an ANGEL enabled router, it must first register with the ANGEL system. The purpose of this stage is to ensure that only authenticated and authorised routers have access to the ANGEL service and flow prioritisation information is only generated and delivered to routers that can make use of it.

In order to perform this, the ANGEL router must be configured with the IP addresses of the ANGEL client managers that it needs to contact. This information can be obtained in a number of ways:

1) The IP addresses can be manually programmed into the router
2) The IP addresses can be provided to the router as part of its automatic IP address discovery (e.g. DHCP, PPP)
3) A mixture of the above two techniques, with some IP addresses manually configured (e.g. the address of an internal ANGEL System) and some automatically obtained (e.g. an ISP-based ANGEL System).

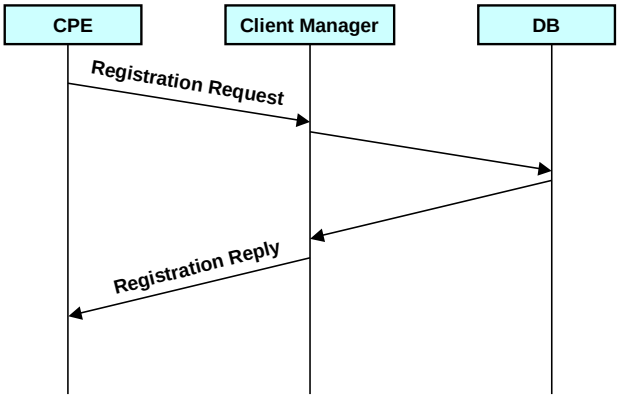

Fig. 12. Simple Registration

Once the IP addresses are obtained, the ANGEL CPE registers with all provided ANGEL Servers. In a network which is secure (no IP address spoofing, no eavesdropping) registration can be done with two messages (see Figure 12). The ANGEL router sends a Registration Request message to the ANGEL server. If the ANGEL CPE can be authenticated and authorised the ANGEL client manager creates a new session entry in the ANGEL database and returns a positive Registration Response message containing all necessary session information (e.g. session timeout). If the the client cannot be authenticated and authorised, or the ANGEL client manager can not provide service to the router for other reasons (e.g. user limit reached) a negative Registration Response is sent back to the router containing the reason why access could not be granted.

In a non-secure network the registration process needs to be more complicated. If IP spoofing cannot be prevented and the ANGEL protocol is based on UDP, an attacker can send fake registration messages with spoofed source IP addresses. This would result in the ANGEL system sending rule updates to CPEs that never registered and that possibly are not even ANGEL en-

abled. Even worse this attack would create fake sessions in the ANGEL systems. Because each session takes some resources a denial of service attack could be mounted not only against CPE devices but also against the ANGEL system itself.

Furthermore, if IP spoofing cannot be prevented, Rule Update messages MUST contain an authenticator. Otherwise an attacker could send fake Rule Updates to CPE devices and insert wrong prioritisation rules into CPE devices or mount a denial of service attack on CPE devices. If eavesdropping is not possible, a simple cookie (cleartext key) is sufficient for authentication but otherwise proper cryptographic authentication is required.

In networks with shared medium (e.g. WLAN) where eavesdropping is possible. Rule update messages would need to encrypted if privacy is of concern. For example, if the rules contain port numbers an attacker could possibly identify the network applications used as many application have well-known port numbers. However, we believe that the case where an attacker cannot obtain the port numbers from the original traffic directly but could exploit Rule Updates to learn the information is unlikely. If traffic from the ANGEL CPE is unencrypted or encryption is only used above the transport layer, port numbers could be obtained directly from the traffic. Rule updates do not need to be secured in this case. If the ANGEL CPE has a secure tunnel into the ISP domain that encrypts transport layer information Rule Updates would also traverse the tunnel and therefore be secured automatically. Only if the CPE is using an encrypted tunnel and the Rule Updates could not be sent across the tunnel encryption would be required.

The secure registration is shown in Figure 13. The ANGEL CPE initially sends a Registration Request to the ANGEL client manager. The client manager responds with a Challenge message. The CPE responds to the challenge with a Challenge Response message and finally the ANGEL client manager sends a Registration Response to the CPE. The Challenge/Response messages are used for verifying that the Registration Request did indeed originate from the CPE and to achieve mutual cryptographic authentication between both hosts. As in the simple registration the Registration Response is either positive including all necessary session parameters or negative containing the reason why ANGEL access is denied to the requesting CPE.

Normally, registration attempts would only fail if the ANGEL client manager denies access to the ANGEL CPE (e.g. CPE cannot be authenticated). However, certain other conditions may cause the registration to fail,

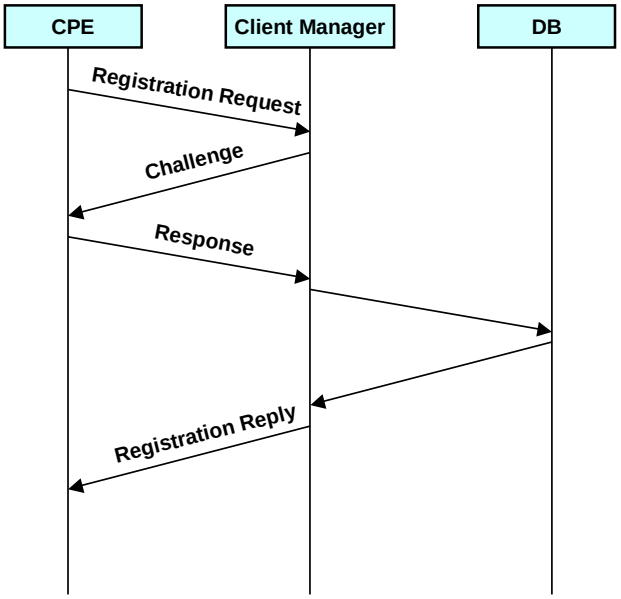

Fig. 13. Secure Registration

such as:

- Monitoring System is currently down.
- Network link is currently down.
- Monitoring System is too busy to respond immediately.
- Packets part of the registration request get lost.

If registration is not successful the ANGEL CPE will retry registration after some retry timeout for a limited number of times. If all of these attempts fail, then the router should wait for the specified inter-registration timeout period before attempting to register again. This behaviour is controlled by the following configuration in the ANGEL router:

1) Registration timeout
2) Number of registration retries.
3) Timeout until next registration attempt

After successful registration a valid ANGEL session is established. The duration of the session is controlled by the ANGEL client manager which sends a session timeout value in the final Registration Response message. An ANGEL CPE need not re-register with the ANGEL System until shortly before the specified timeout period has passed, following which the entire registration procedure should be repeated. An ANGEL client manager will remove any sessions where the timeout has been expired. This timeout allows the removal of an ANGEL router from a client manager's database of connected devices should the router be disconnected from the network or reconfigured to not use ANGEL.

The Registration has four different messages: Registration Request, Challenge, Challenge Response, Registration Response. Challenge and Challenge Response are only needed if cryptographic authentication and/or encryption are required or if a connectionless transport protocol is used and IP address spoofing cannot be prevented.

Registration Request

- Version: Version of the protocol.
- Preferred security level (optional): ANGEL router chosen security levels.
  - No security
  - Cookie authentication (negotiate 64bit cookie (random number) which is later included in rule update messages).
  - Crypto authentication (negotiate shared secret (session key) which is later used to compute signature for rule updates. messages e.g. using MD5, SHA1 algorithms).
  - Crypto authentication and encryption (besides computing signatures rule update messages are also encrypted with the shared secret using an encryption algorithm).
- Preferred security algorithms and parameters (optional): ANGEL router lists all preferred algorithms and parameters depending on the security level e.g. preferred signature and encryption algorithm.
- ANGEL router IP address: ANGEL client manager needs this to determine where to send prioritisation information.
- ANGEL router subnet mask: ANGEL client manager uses this in conjunction with the IP address to determine which IP addresses are routed by this router and therefore which network flows are relevant for classification purposes.
- ANGEL router gateway address (optional): If the ISP wishes to deploy ANGEL to prioritise flows on the downlink, then this can be used to determine which downlink router is responsible for flows to the specified client.
- Random data (optional): If the ANGEL router wants to authenticate the ANGEL server it generates some random data.

Challenge

- version: Version number of the protocol.
- Preferred security level (optional): ANGEL client managers preferred security levels.
- Preferred security algorithms and parameters (optional): ANGEL client manager lists all preferred algorithms and parameters depending on the security level and algorithms and parameters preferred by the ANGEL router.
- ANGEL client manager certificate including public key: The client manager sends its certificate to the router, which uses it to verify the signed data.
- Random data signed: ANGEL router random data signed by the ANGEL client manager.
- Request client authentication flag (optional): ANGEL client manager can request client authentication.
- Random data (optional): If client authentication is required ANGEL client manager generates some random data.

Challenge Response

- Version: Version of the protocol.
- Security level: Selected security level for the session.
- Security algorithms and parameters: Selected authentication and/or encryption algorithms and parameters for the session.
- ANGEL router certificate including public key (optional): If client authentication was requested the ANGEL router must include its certificate.
- Random data signed (optional): If client authentication was requested the ANGEL router must sign the random data sent by the client manager.

Registration Response

- Version: Version of the protocol.
- Session key (optional): Session key to be used to secure later rule update messages. This is either a session key used for signing and/or encrypting the messages or a random number that will be send in plaintext.
- Session timeout: Maximum duration of the session. The ANGEL router must reregister before the timeout if it wants to continue the session.
- Rule timeout: Default rule timeout for all rules send during the session. If no packets have matched a rule for longer than the timeout the ANGEL router must remove the rule.
- Error code (OK, error): This fields tells the ANGEL router if the registration was successful or not
- Reason (optional): If the registration is rejected because of any error, the client manager can provide

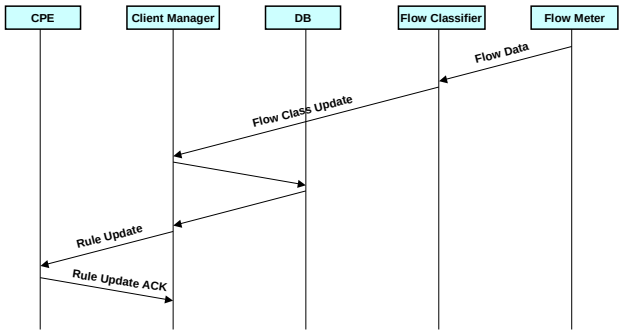

Fig. 14. Rule Update

more detailed information about the problem.

2) Rule Update: Once registration is complete communication between ANGEL client manager and ANGEL CPE predominantly consists of updating the prioritisation information. This information is required to be transmitted over the bottleneck downlink. Furthermore, acknowledgments must travel across the even more bandwidth limited uplink back to the ISP. However, a timely delivery of prioritisation rules is required to setup QoS for realtime flows as quickly and reliably as possible. Rule Updates and Rule Update Acknowledgments are sent with highest priority (if possible) to avoid message loss and unnecessary delays.

Rule updates with acknowledgments are more suitable to ensure quick rule updates and limit the amount of messages in the network. Therefore the ANGEL CPE sends a Rule Update Acknowledgment message back to the ANGEL client manager for each authenticated rule update it receives.

The ANGEL client manager has two configuration options for rule updates:

- Rule update timeout
- Rule update retries

After each Rule Update message sent, the ANGEL client manager for an acknowledgment received within the rule update response timeout. If no acknowledgment is received the ANGEL client manager will resend the Rule Update message. The ANGEL client manager will resend a Rule Update message only for the configured number of retries. To match rule updates and rule update acknowledgments the ANGEL client manager puts a sequence number in the Rule Update message and the ANGEL router returns the sequence number in the acknowledgment. Figure 14 shows the rule update process including flow data and flow status update messages.

An ANGEL router may be communicating with more than one ANGEL System, of which some may be communicating in plaintext mode while others are using cryptographic authentication mode. During the registration a security mode has been agreed between ANGEL client manager and CPE. The security mode as well as all necessary parameters (e.g. session key) are stored for each session together with an ANGEL System identifier (e.g. ANGEL client manager IP would be simplest - would require to have a cluster of servers which looks like one IP). Necessary parameters such as session key are negotiated during the Registration phase.

If a Rule Update is received the ANGEL CPE looks up the security mode and parameters for the ANGEL system based on the identifier (e.g. ANGEL server IP). Then it (possibly decrypts and) verifies the authentication. The rule will only be updated on the CPE if the message can be authenticated. Otherwise the message is silently ignored (no negative acknowledgment is sent to the ANGEL server).

Rule Update

- Version: Version of the protocol.
- List: List of rules to be installed.
  - Flow key: Fields necessary for installing filter rules on the ANGEL router e.g. IP addresses, ports, protocol.
  - Action (optional): Action that ANGEL router has to execute when packets match the rule. In the first version of the protocol the only action is 'prioritise' but other actions could be added e.g. 'block'.
  - Priority (optional): If more than two priorities exist in the ANGEL system, the ANGEL client manager has to specify the priority of each rule.
  - Timeout (optional): ANGEL client manager can specify a timeout for the rule which overrules the default timeout sent to the ANGEL router during registration.
- Signature/Cookie (optional): Cookie or a signature is inserted into the message depending on the security level negotiated during registration.

Rule Update Acknowledgment

- Version: Version of the protocol.
- Signature/Cookie (optional): Copy of the signature/cookie from the rule update message.

3) Flow Data: All Flow Meters export the current statistics for each flow at regular time intervals. We assume that the ISP domain network is either: appropriately dimensioned so that there is no message loss and

therefore no need for acknowledgments/retransmissions, or inherently lossy and therefore a reliable transport protocol is required.

Flow Information Data

- Version: Version of the protocol.
- List: List of per-flow information where a number of statistics are assigned to each flow
  - Flow key: Fields that define how packets were grouped into flows e.g. IP addresses, ports, protocol.
  - Flow attributes: List of per-flow packet statistics that have been metered since the last data transfer

4) Flow Class Update: If the predicted class for a flow has changed or the flow is new, the Flow Classifier sends the flow information, including the current class, to the ANGEL client manager. We assume that the ISP domain network is either: appropriately dimensioned so that there is no message loss and therefore no need for acknowledgments/retransmissions, or inherently lossy and therefore a reliable transport protocol is required.

Flow Class Update

- Version: Version of the protocol.
- List: List of flow information where a traffic class has been assigned to each flow.
  - Flow key: Fields necessary for installing filter rules on the ANGEL router e.g. IP addresses, ports, protocol.
  - Class: Predicted class of the flow e.g. realtime, non-realtime.
  - Probability (optional): Probability how sure the classifier is about its prediction. In case the ANGEL classifier is not very sure (low probability) the ANGEL client manager could choose not to install rules.

ACKNOWLEDGMENT

This work was supported from 2005 to early 2007 by the Smart Internet Technology Cooperative Research Centre, http://www.smartinternet.com.au.

REFERENCES

[1] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," IETF, RFC 2119, Mar. 1997, http://www.ietf.org/rfc/rfc2119.txt.

[2] Lawrence Berkeley National Laboratory, "TCPDUMP/LIBPCAP Public Repository," January 2006, http://www.tcpdump.org/.

[3] S. McCanne and V. Jacobson, "The bsd packet filter: A new architecture for user-level packet capture," in *1993 Winter USENIX Technical Conference*, January 1993, ftp://ftp.ee.lbl.gov/papers/bpf-usenix93.ps.Z.

[4] U. Inc., "Solving Performance Problems with Interactive Applications in a Broadband Environment using StreamEngine Technology," Ubicom Inc., Tech. Rep. 20041031, October 2004, http://ubicom.com/pdfs/whitepapers/StreamEngine-WP-20041031.pdf.

[5] The OpenBSD Project, "PF: Firewall Redundancy with CARP and pfsync," February 2006, http://www.openbsd.org/faq/pf/carp.html.