Policy based Charging in Multimedia Networks

Brian Lee [1] and Donal O'Mahony[2]

[1] Ericsson Systems Expertise, Athlone, Ireland
Brian.A.Lee@ericsson.com
[2] Trinity College, Dublin 2 , Ireland
Donal.Omahony@cs.tcd.ie

Abstract. The telecommunications landscape is undergoing a period of dramatic change. A near term next generation network (NGN) is emerging characterized by a rich set of services and a dynamic and competitive marketplace where innovation and time to market will be critical success factors. Changing consumer-provider relationships, competitive pressures and new e-commerce technologies will accelerate the use of real time payment and prepayment techniques. This will increase the demand for real-time processing of charging data. Powerful, new, charging support systems will be needed to enable real time charging of these new services. Policy based management appears to offer potential solutions.

I Introduction

Current approaches to IP service charging systems development are an evolution of the traditional PSTN approach to charging. While these systems represent an advance on the state of the art they remain based on the "call data record", (CDR), paradigm and will not provide the scalability and flexibility needed for NGN charging. A more radical approach is needed which pushes elements of the charging process, e.g. rating, down into the network layer and allows for a more distributed and scalable charging system. The central theme of this paper is that the use of programmable networking technologies, specifically *active networking* and *policy management*, can provide a distributed charging system solution which meets the above demands.

Section II examines the evolving NGN and identifies the key issues characterizing its growth and identifies the main challenges it poses for charging and billing systems. Section III reviews the current charging approaches taken in the industry today and identifies why they will not provide the solutions that are needed in the emerging NGN. Section IV introduces the PEACH (Policy Execution Environment for Accounting and Charging) charging system and describes the main features of the system. Chapter V describes the some examples of the usage of PEACH. Section VI surveys the state of the art.

II the Next Generation Network

The flux in the telecom industry caused by the ongoing technological and marketplace evolution has given rise to a diverse and complex public communications network. Older technologies are gradually being replaced as and when market economics judge such updates to be viable. Newer technologies are continually being absorbed into the network. While the rate of change varies in different places, the network *is* nonetheless changing and .a next Generation Network (NGN) is emerging. Technology diversity, a service rich marketplace and fierce competition amongst a wide variety of service providers will characterize this network. The NGN will be complex in all respects. Figure 1 below presents a reference model, after [1] that places the various aspects of the NGN in context

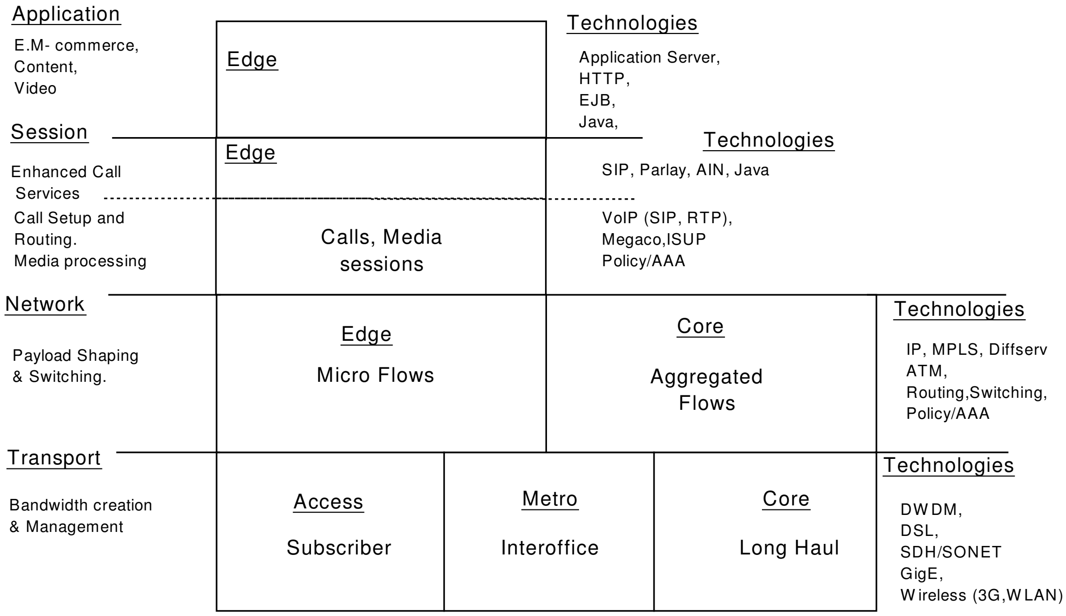

Fig. 1. NGN Reference Network

Networks and Services

The NGN in reality consists of a network of networks. Access networks enable end-users to gain entry to a service rich edge network. Access networks are based on technologies such as PSTN/ISDN, DSL, cable GSM , UMTS, wireless LAN based on 802.11 and wireless MAN based on 802.16 The emerging wireless network is sometimes termed “fourth generation” (4G) or “Beyond 3G”, [2].

The network layer offers connectivity and routing services i.e. users can transfer information to, or between, addressable entities. The NGN network layer will be a *multiservice* network based on IP. This means that the network will provide better service guarantees, or quality-of-service (QoS), to some traffic flows to meet the transmission needs of real-time traffic flows. In order to allow this model to scale it is necessary to maintain flow state only in *edge* networks and to allow the high speed *core* of the network to be state-free.

The multiservice IP network serves as the platform for the provision of a large number of value added end-user services in the session and application layers. The HTTP like semantics of SIP (Session Initiation Protocol) means that the NGN concept of a session is much broader than the usual "telephony call" meaning of the word. This allows the possibility to create a wide variety of session based services e.g. multiparty gaming while the inclusion of media, enabled by RTP (Real Time Transport Protocol), extends the service creation process even further to enable services based on a combination of media types.

Service Marketplace

As observed earlier each layer of the network offers services to layers above. The relationship between each layer is essentially client-server and the lower layers are in general unaware of the existence of upper layer services. One consequence of the layering of services and the diversity of network types is the emergence of a variety of service provider types. Examples include wireless ISP's (WISP), mobile Virtual Network Operators (MVNO's), who lease network capacity from large cellular operators and handle subscriber management and billing, and communication ASPs (CASP's) offering enhanced communication services using enhanced service API's. Competitive pressures will lead to specialisation and new business models based on wholesale and partnering relationship between service providers will emerge.

III Pricing and Billing

A wide variety of pricing schemes is likely to be applied to multimedia services in the NGN. Competition will force service providers to differentiate themselves and they will want to experiment with different pricing models. Current approaches to pricing for Internet access are predominantly based around the *flat rate* model. In an examination of pricing models and cost structures for ISP's, Leida, [3], notes that service providers lose money for the provision of basic access with flat rate charging and that these losses are exacerbated when providing multimedia services. He concludes that some form of usage based pricing may be needed to recover costs and make profits.

Current approaches to usage-based charging are based heavily based on the processing of 'call detail records' (CDR), i.e.. records of usage generated by the network. In legacy approaches to billing these records are normally extracted from the network for processing in a billing centre. There, the CDRs are 'rated' to determine the charge to be associated with the users network session. Invoices are then prepared and dispatched for payment. These approaches are evolving in modern networks in a number of ways. Firstly the volume of CDR's is growing greatly as networks become more complex and more distributed in nature e.g. it is estimated that GPRS networks may generate up to forty times as much charging data as GSM networks, [4]. Secondly the complexity of services e.g. IP based QoS, will demand that billing systems become real time based and that some services become rated in real time in order to avoid fraud. Thirdly there is a continued movement toward the use of prepaid billing, a trend that is anticipated to increase as e-commerce technologies enable more dynamic business models.

These trends are placing new demands on charging and billing systems and leading toward an ‘unbundling’ of billing systems. Mediation systems are becoming more widespread and more complex in term of functionality as aggregation and correlation of multiple charging records becomes necessary to gain a view over a single user session. Rating engines are being extracted from the billing systems and deployed nearer the network, in some cases as adjuncts to the mediation systems. At the cutting edge specialized hardware engines are being deployed in layer 4-7 stateful inspection switches to rate IP encapsulated traffic, [5]. In the NGN a number of factors will drive the evolution of charging systems including

- The need to be able to rapidly create and deploy new pricing models
- The need to process charging data in real time
- The need to deal with very large amounts of charging data
- The need to be adaptable to new services and network elements

While current trends in charging system architecture are encouraging we do not believe that these approaches will provide the solutions needed. Today’s systems are too tightly bound to the use of CDR’s and we believe that over the longer term this will prove untenable. Neither will prepaid architectures scale in their current form as the billing system must maintain state for every session and this gives rise to large volume of traffic in a multimedia, multi-layered service network.

To overcome these limitations we propose a distributed charging system in which some elements of the charging /billing system are pushed into the network layer. In particular we propose that the rating engine be collocated with the data metering function. This will greatly reduce the amount of charging data that will need to be moved around the network and will remove the need for CDR’s altogether in most cases. This represents a new architectural approach to provide the flexibility and scalability for real-time charging of complex network services. Rather than bringing the data to the processing logic we are bringing the processing logic to the data.

IV Charging Framework

The framework described here is called PEACH (Policy Execution Environment for Accounting and Charging) and is based on complementary fields of programmable networking research viz. *active networking* and *policy based management*.

Active networks are intended to allow programmability at the packet level, [6]. In addition to data, packets traversing an IP network may carry programs that can be executed in active node “execution environments”, (EE), thereby introducing new features or services into a network. An EE provides the resources and support that agents need in order to execute, communicate and migrate through the network.

Policy management is a general term that describes how the resources and services of the network are allocated and managed in order to meet the strategic and business needs of the service provider. *Policies* formulate and express these goals in context specific terms e.g. security policies, QoS policies, routing policies, traffic management and indeed charging and accounting policies. A policy is defined as an aggregation of *policy rules* [7]. A policy rule is composed of a set of conditions and a set of actions. The use of rule based policies to manage and control has been spurred recently by the work on QoS policy management in the IETF [8].

The key features of PEACH include

- An architecture for interactive session-based NGN charging.
- A language and programming model for charging policy definition
- An execution environment for policy evaluation and communication

Charging Reference Model

PEACH defines a functional model for multi-layer charging based on well defined logical nodes. See Fig. 2. The model allows PEACH to cater for charging for any multimedia service session, whether in a single layer, or bundled across multiple Layers. The multi-layer aspect is a crucial feature of the reference model as it recognizes the reality of the evolving NGN as described in section II i.e. the model allows for separate pricing models to be applied at each network layer.

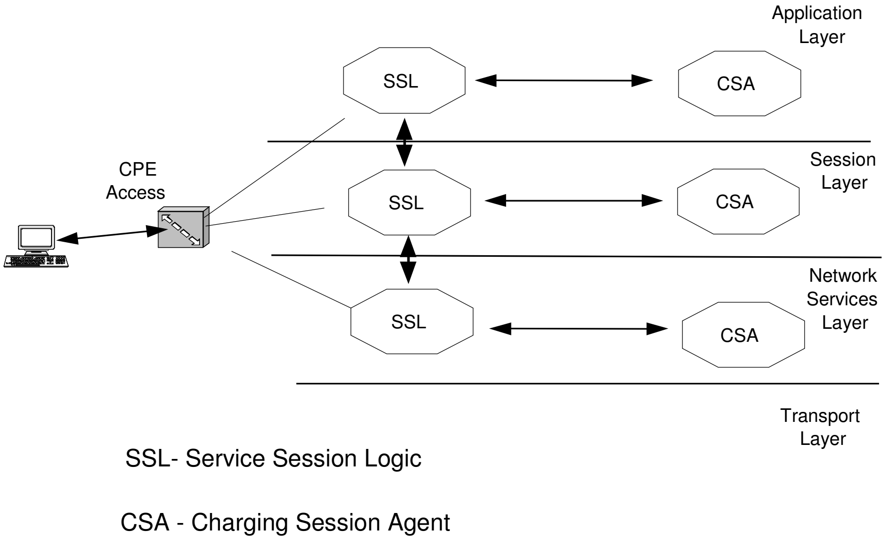

Fig. 2. Charging Reference Model

The model is predicated on the notion of a session. For example a VoIP SIP session may use a number of different RTP media sessions and the lifetime of the SIP session may vary considerably from the RTP sessions. The user-session is represented in the model by the Service Session Logic (SSL) entity that controls the set-up and removal of the session. A Charging Session Agent handles the charging for the session. This is decomposed into the three sub-nodes viz. the charge coordination point (CCP), charge analysis point (CAP) and the charge execution point (CEP). The CCP interacts the with the service logic and coordinates all the charging entities involved in the session. The CAP decides which pricing model should be applied and where in the network charging should apply. The CEP applies the actual pricing model in real time i.e. rates the call in real time.

Policy Language

PEACH provides a programming language, APPLE (Accounting Policy Programming Language), to define charging policies and charging agent task logic. APPLE provides two constructs to encapsulate program content. A *Rule* defines a specific policy for a particular charging situation. A *Module* defines active task logic for charging agents. It encapsulates state and co-ordinates with other modules to provide dynamic charging services. The charging model functional entities introduced above, (viz. CAP, CCP and CCP), are implemented in APPLE as modules. Rules represent static polices and capture business goals and pricing models. They are not standalone executable elements and must be invoked from within a module though rules may invoke other rules. Typically a charging policy is implemented as a chain of rules where the primary, or root rule, is triggered from within a module.

APPLE provides a form of *service composition* typical of programming network languages, [6], i.e. the languages provides hooks to incorporate service components which may be written in other languages into APPLE programs. APPLE distinguishes between *policy parameters* and *policy components*. Policy parameters are charging scheme data types while policy components are service components.

APPLE is a procedural language and can be regarded as a scripting language i.e. it is an interpreted language. As well as standard control flow statements (if, switch etc) APPLE provides statements for data manipulation, event handling and inter-node communication. Modules and rules place and retrieve data on a shared space and from external sources by means of the language statements

```
inp( policy parameter1, ... policy_parameterN);
outp(policy_parameter1, ...policy_parameterN);
```

Rules invoke each other by means of the call statement:-

```
call policy_rule;
```

Modules exchange information between each other by means of the statements

```
send(node-address, application, message,
policy_parameter1… policy_parameterN);

inputEvent{ event1 ->{}; ..eventN->{});
```

A message that is sent from node A is received as an event in node B. PEACH also provides support for code mobility: -

```
move(dest_url, module_name);
start(dest_url, module_name);
```

APPLE separates the act of migrating a code module from the act of activating the module. This gives greater flexibility to the charging architecture as it allows different nodes to initiate each action. APPLE also support the activation of user defined logic by means of:-

```
action(policy_component.called_method,
policy_parameter1..policy_parameterN);
```

```
Module AnalyseCharging {

  // message receiver info.
  application=" aChargingCoordinator";
  message="chargeResults";
  ccpUrl="bilbo.bagend.shire:39001/ccp";

  // do analysis
  call IpQ_getEFcsch;

  // get results and return info to caller
  inp(chargingScheme);
  send(ccpUrl, application,  message,
       chargingScheme);

  // wait for next message and process
    inputEvent{
            close-> {
                    action(Logger.write, application);
                    exit(); // clean up and die
             }
  }
} //end module
```

```
Rule IpQ_getEFcsch{

   inp(originForCharging);

   switch(originForCharging)
   {
      between(1,10) :
         chargingScheme =#flat_rate;
      between(11,20) :
         chargingScheme=#no_charge;
      between(21,100) :
         chargingScheme=#EF_time_volume;
       in (101,102,103,104):
         chargingScheme=#EF_volume;
       gt 500:
           chargingScheme=#no_charge;
        default:
         chargingScheme =#flat_rate;
   }

   call CheckToD;
   meter="ipQosMeter";
   outp(chargingScheme,meter);

} // end IpQ_AnalyseEF
```

Fig. 3. APPLE Rule and Module example

A simplified example of a module and an associated rule are shown below. In this example a Charge Analysis Point (CAP) node has received a message from a charge coordination point (CCP) requesting analysis for a service session (in this case for the use of a particular IP QoS i.e. Expedited Forwarding) It invokes a rule to determine the type of charging to apply, returns this to the caller and then waits for any further messages

Context Support

A context is a PEACH concept that defines a service category for which charging policies and modules may be specified. A PEACH node can support multiple simultaneous contexts. A context defines a family of policy parameters, policy rules and modules and policy (service) components. These elements are defined in an XML context configuration file.

A simplifying assumption in PEACH is that it is necessary only to manipulate primitive data rather than structured data. There are no entities or relationships visible in APPLE rules or modules and thus an information model of charging domain entities is not needed. The context mechanism enables the basic data type vocabulary to be defined and thereby removes the need for a complex object oriented schema

Policy components must be provided in order to retrieve policy parameter data from external sources. Policy parameters may, in principle, be drawn from arbitrary sources and the PEACH context designer has complete freedom to extract data from any suitable source. This in turn puts the onus on the context designer to provide the appropriate policy component. Each context has an associated policy protocol. The policy protocol is used to transfer data between nodes. Many contexts can use the same policy protocol. PEACH is designed to support different policy protocols. This is done in order to be able to adapt PEACH to multiple network technologies. The data objects in the policy protocol are referred to as *protocol objects*. Policy protocols must support delivery of the data to an agent or handler at the destination node.

There are a number of directions in which future development of PEACH could occur. So far we have concentrated only on the language and policy evaluation. Since PEACH is liable to be used by marketing/commercial then a graphical “user friendly” toolset is needed to facilitate rapid creation of policies. APPLE is intended for deployment on network nodes but there is no conceptual reason why it could not be deployed on handsets. However a more efficient implementation of the language would be needed. This might entail a more efficient interpreter or the APPLE could be compiled to Java or other languages. Finally since there are a number of existing approaches to policy management it could be useful to examine if APPLE could be integrated into a broader policy context e.g. DEN-ng, [7].

PEACH is implemented as a prototype in JAVA.

V Application to Multimedia Services

One of the main drivers in the design of PEACH has been the capability to enable charging for composite, or “bundled” end-user service i.e. services which may be composed of a number of network services, each of which may have different charging schemes[1]. Network multimedia services are typically bundled services e.g. SIP based VoIP. The use of PEACH to carry out charging of SIP based VoIP has been investigated - cf. Fig. 4 below:

[1] Different service providers may indeed provide them.

This is based on the use of the QSIP implementation of SIP, [9] and using the implementation of Diffserv on Linux, including the BBTP bandwidth broker infrastructure, [10]. Figure 4 show the interaction between the service session logic (SSL from Figure 2) in each layer i.e. QSIP server and bandwidth broker, and the PEACH charging nodes in each layer viz. charging control point (CCP), charging analysis point (CAP) and charging execution point, (CEP). The charging nodes have been deployed in a variety of configurations on different hosts during this case study. QSIP contains non-standard extension to SIP to enable reservation of network resources. Minor Modifications have been made to both BBTP and QSIP in order to integrate them with PEACH.

The sequence of activities is :

1. The QSIP server receives a request to set up a VoIP session and creates a CCP to handle the charging . The CCP in turn invokes the CAP to determine which type of charging to apply. In the case where real time charging is to apply the CAP instantiates a charging algorithm in the CEP. The CCP returns a “charging_origin” token to the QSIP server to be passed to the Diffserv charging via the bandwidth broker (BB).
2. The QSIP server then invokes the bandwidth broker to reserve resources in the network for the RTP session . The BB invokes creates a CCP and the charging process is repeated on the network layer. The token is passed to the charging analysis. If real-time charging is to occur then a module is instantiated in the CEP.
3. When all analysis is complete the VoIP session proceeds and charging is carried out in the CEP’s. The calculated costs are periodically output to a charge accumulation point (not shown) which could be any of a range of options from a display of charge to a real-time payment system. When the session terminates all agents are removed.

Communication between the QSIP client and the PEACH entities is by means of a ‘request-reply’ protocol called PRR (PEACH request/reply). PRR is also used for communication between the PEACH entities. PEACH is architected so that any request/reply protocol maybe used

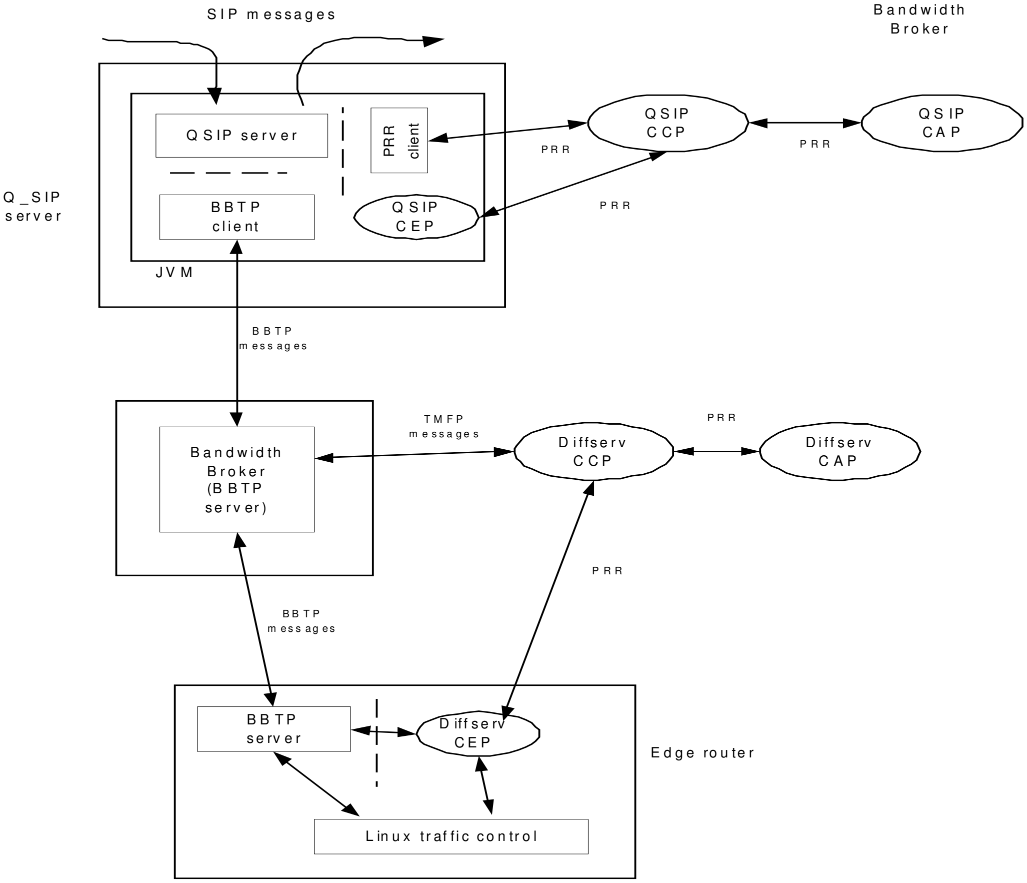

Fig. 4. QSIP VoIP charging

Charging in each layer takes places independently of each other though the choice of charging scheme is influenced by the charging_origin token. Further the scenario above illustrates a fully distributed PEACH system which is unlikely to be the case in a deployed system for efficiency reasons. PEACH is designed to allow arbitrary distribution but does no mandate it. The architecture described above can be applied to many different types of NGN service and has been implemented also for transfer of MP3 files across a Diffserv network.

VI Related Work

Travostino, [11] describes an "Active IP Accounting Co-processor Environment" (AIACE) which embeds a dedicated accounting processor in network nodes to perform real time processing of accounting data. Real time processing will allow, prepaid billing, fraud detection, per-flow accounting etc. AIACE allows accounting "plug-ins" to be dynamically loaded onto the coprocessor. Plug-ins represent specific accounting functions or applications and are akin to active applications in the active network architecture. As presented AIACE is primarily an architectural approach and no information is given on how policies are defined or how "plug-ins" are deployed and activated. It is thus difficult to gauge how flexible AIACE is for particular service scenarios. AIACE in its current form is therefore unlikely to provide the flexibility and support for a competitive NGN marketplace.

Briscoe et al. [12], describe an approach to accounting based on "active tariffs". This proposes to push all accounting and charging on to end-customers hosts. Charging is on a per packet basis and tariffs, which are active Java objects, are disseminated to end-users via multicast, a multicast group corresponding to a particular network service. PEACH is intended to provide a flexible, extensible and scalable solution to real time charging in a multi-provider NGN and while Briscoe is also concerned to provide flexibility, his major goal is to use the charging/pricing system for traffic management by means of dynamic pricing. PEACH is a more general and suitable solution for real-time NGN service charging.

Carle, [13], describes a billing architecture that is policy based at all levels and provides the architectural adaptability and policy flexibility to incorporate new technologies. However Carle's work does not provide enough to meet real time charging needs. It is primarily a description of an architecture. No detail is given on how charging schemes can be created or modified.

Bellavista et al., [14], describe a charging approach based on the use of mobile agents, specifically to meet the challenges brought about by mobility in the NGN. Their proposal uses a couple of mobile agents, a configuration agent (CA) and a gathering agent (GA). The CA moves with the user and has responsibility to configure charging related parameters but also to configure any service specific elements that may be needed e.g. media transcoders. When the session is finished a GA is sent along the path to collect charging data that are then transferred to the billing system so that charges may be calculated. This approach addresses only the issue of mobility and does not address the overall issues of charging in the NGN.

Policy management is an active area of research and a number of approaches to policy specification and enforcement have been put forward. Perhaps the best known is Ponder, [15], DEN-Ng, [7] and the IETF QoS policy model ,[8].

Ponder is a declarative language which can be used to specify a broad range of policy types. Ponder also offers facilities for policy grouping and policies can be compiled to standard computer languages such as Java by provision of a suitable compiler back-end. Although Ponder allows for formulation of a wide variety of policies it is not suitable for the problem of real time charging. Charging policies require data comparisons to be quickly and easily made, something Ponder is not designed for. Further Ponder was not designed to be stateful and though it can be event driven it does not naturally lend itself to a stateful event loop charging session controller.

DEN-ng is primarily an architectural approach to policy based network management to facilitate the definition and deployment of a range of related policies. DEN-ng defines a "policy continuum" which represents the communications network as a number of layers with more abstract layers at the upper levels, e.g. service layer, and more device specific layers at the bottom. DEN-ng recognizes the need for a variety of formalisms to express policies. In terms of the DEN-ng policy continuum APPLE is located at the system or network layer..

References

1. Clavenna S. and Heywood P., "Optical Taxonomy", lightreading.com. http://www.lightreading.com/document.asp?site=lightreading&doc_id=3780&page_number=1
2. O'Mahony D. and Doyle L, "Beyond 3G: 4G IP-Based Mobile Networks" in *Wireless IP: Building the Mobile Internet* ed. Dixit S., Arctech House, Norwood, MA 2003 Chap. 6 pp 71-86.
3. Leida. B.A," A Cost Model of Internet Service Providers: Implications for Internet Telephony and Yield Management" , M. Sc. Massachusetts Institute of Technology, Feb 1998.
4. Sur A, "Technical Tutorial: Mediation Device Requirements for GPRS", *Billing World*, vol. 7 no. 6 June 2001, pp 100-107
5. "Encharge ", www.p-cube.com/products/encharge
6. Calvert K.L. et al. "Directions in Active Network".*IEEE Network* Oct. 1998
7. Strassner J., "Policy Based Network Management- Solutions for the Next Generation", ", Morgan-Kaufmann, Amsterdam 2003
8. Moore B. et al., "Policy Core Information Model – Version 1 Specification", IETF RFC 3060 Feb. 2001, www.ietf.org
9. Salsano S. and Vetriani L., "QoS and Policy Control by means of COPS to support SIP based Applications" *IEEE Network* Mar/Apr 2002
10. "Bandwidth Broker Transfer Protocol" Revision 0.3, British Columbia Institute of Technology, Group for Advanced Information Technology, Nov. 1998
11. Travostino F., "Towards an Active IP Accounting Infrastructure",., *Proc. 3rd IEEE Conference on Open Architectures and Network Programming, OPENARCH 2000*, Tel Aviv, Israel, March 2000
12. Briscoe B. et al., "Lightweight Policing and Charging for Packet Networks,". , *Proc. IEEE 3rd Conference on Open Architectures and Network Programming, OPENARCH 2000* Tel Aviv, Israel, March 2000
13. Carle G. et al. "Policy Based Accounting", IRTF Internet Draft Feb 2002, <draft-irtf-aaaarch-pol-acct-04.txt>

14. Bellavista P, Corradi A and Vecchi S, "Mobile Agents for Usage-based Accounting in Wireless Ubiquitous Networks", WOA2002 University of Milan-Bicocca, November 2002.
15. Sloman S. and Lupu E., "Security and Management Policy Specification", IEEE *Network* March 2002