Conceptualisation of an application of adaptive synthetic socioeconomic agents for intelligent network control

Doug Legge, Atta Badii
The Intelligent Media Systems & Services Research Centre
University of Reading
(www.imss.reading.ac.uk)
(d.j.s.legge, atta.badii)@reading.ac.uk

Abstract**—** The deployment of Quality of Service (QoS) techniques involves careful analysis of area including: those business requirements; corporate strategy; and technical implementation process, which can lead to conflict or contradiction between those goals of various user groups involved in that policy definition. In addition long-term change management provides a challenge as these implementations typically require a high-skill set and experience level, which expose organisations to effects such as "hyperthymestria" [1] and "The Seven Sins of Memory", defined by Schacter and discussed further within this paper.

It is proposed that, given the information embedded within the packets of IP traffic, an opportunity exists to augment the traffic management with a machine-learning agent-based mechanism.

This paper describes the process by which current policies are defined and that research required to support the development of an application which enables adaptive intelligent Quality of Service controls to augment or replace those policy-based mechanisms currently in use.

Index Terms**—Quality of Service, multi-agent, ontology.**

I. INTRODUCTION

Within Service Orientated Architecture (SOA) network topologies that implement Quality of Service (QoS) mechanisms for communication, there exists an issue with the human-oriented requirements analysis and ongoing policy change management procedures to ensure acceptable user perception of the application(s) classified. This is in part due to that initial "application requirements analysis", which can be complicated and involve many differing user-groups, each with distinct and differing options, perceptions and goal orientated requirements (design by committee). The provision of an ongoing change management function to support business or network changes means many corporate, particularly Small-to-Medium-Enterprise (SME) businesses face real challenges regards resource, skill-set, business priorities and timeframe.

It has been well documented within the network (e.g. IETF, IEEE) and vendor community (e.g. Cisco Systems, Juniper Networks) network metrics and sensitivities required to define appropriate QoS policies, but it is the goal of the enterprise to interpret these various best-practises, in such a way that they can support the "business requirements" of the applications being deployed. Therefore the application requirements analysis must tie those technical-criteria and business-goals into a defined policy.

This typically necessitates a business analyst, questioning groups of collaborative parties, possibly including: an application development team; user group; management group; network team; technical support group and technical management group, to establish the properties of the application or applications under development, characteristically influenced by:

- Protocol sensitivities (e.g. voice compared with ftp)
- Application sensitivity (e.g. real-time ticketing system, see table A.1.1. at appendix A)
- Concurrent user access
- Infrastructure (e.g. the abilities of the supporting LAN, WAN & client > server environment)
- Topology (e.g. the geographic, political & corporate distribution of the intended user group)
- Commercial business priority (e.g. willingness to support priority of application <u>*x*</u> over application <u>*y*</u>) and any associated surcharge or penalty

From this requirements analysis a policy can be proposed, to implement mechanisms by which to classify, mark and "rate limit" traffic. Classification is the process of identifying traffic and categorising that traffic into different classes such that prioritisation can be allocated. Typical traffic descriptors include: Class of Service (CoS); incoming interface; IP precedence; Differentiated Services Code Point (DSCP); source or destination address; application; and Multiprotocol Label Switching (MPLS) Experimental Bits (EXP). After the traffic has been defined and classified, it is accessible to QoS mechanisms for treatment. Classification should take place at the network edge as this reduces the transfer of packets which may only be dropped later due to their low classification marking value. Layer 2, Class of Service (CoS) mappings can be implemented, however a disadvantage of CoS markings is that frames lose their CoS markings when transiting a non-802.1Q or non-802.1P link, including any non-Ethernet WAN link. Therefore a more ubiquitous permanent marking such as IP DSCP, should be used for network transit. In a Cisco network device environment this would take the form:

Classification

```
class-map name
        match traffic type
```

description :an example of which could be match protocol http url "*important*"
!

Marking/Policing

```
policy-map name
        class name
                set mark
```

description :an example of which could be ip dscp af21
!

Rate Limit

```
service-policy name (applied at an interface level)
        set constraint
```

description :an example of which could be setting the bandwidth available, or the queuing mechanism to be applied, such as: fair-queue or LLQ
!

Assured Forwarding, defined in RFC 2597 [2], delivers a Per Hop behaviour (PHB) for applications that require a better reliability than the best-effort service, which identifies four classes of service, and within each class, three drop precedence's, detailed in Table A.1.3 at appendix A.

The analysis may require a review of all existing business applications, such as: finance; sales; billing; customer service; email; call-logging systems, which were previously mapped into class-groups based on the business and network audit of their requirements, sensitivities and operational business criticality. This is due to best practise guidelines that recommend that no more than 3 applications are mapped to the mission-critical, and transactional service classes. The greater the bandwidth saturation of the prioritised classes the greater the impact on those applications allocated with a lower priority classification. The resulting application requirements proforma (see table A.1.2 at appendix A) must then obtain executive endorsement, including an indication of any supporting changes or impacts (e.g. application *x* may have to be demoted to allow sufficient resource for this new delivery). A full policy to support a business scenario above is outside the scope of this paper, however a typical policy structure, based on Cisco Internetwork Operating System (IOS) syntax, defined for applications in use within the IMSS department at the University of Reading might show:

```
ip cef
!
class-map voip-rtp
        match protocol rtp audio
class-map http-blackboard
        match protocol http url "*blackboard*"
class-map http-reading.ac.uk
        match protocol http url "*reading.ac.uk*"
class-map match-any NetMeet
        match protocol rtp payload-type 4
        match protocol rtp payload-type 34
!
policy-map imss-list
!
  class voip-rtp
        set ip dscp EF
  class NetMeet
        set ip dscp AF41
  class http-blackboard
        set ip dscp AF 21
  class http-reading.ac.uk
        set ip dscp AF23
  class class-default
        set ip DSCP default
!
interface Fastethernet 0/0
!
  service-policy input imss-list end
!
```

What this policy, called imss-list, specifies is:

- Match the protocol rtp audio and mark with DSCP Expedited Forwarding (EF)
- Match the protocol NetMeet (rtp payload-type 4 and payload-type 34) and mark with Assured Forwarding (AF) 41
- Match any http traffic that contains the URL string "blackboard" anywhere in the URL and mark with Assured Forwarding (AF) 21
- Match any http traffic that contains the URL string "reading.ac.uk" anywhere in the URL and mark with Assured Forwarding (AF) 23
- Match all other traffic and mark with DSCP default which is DSCP BE (Best Effort)

These application sensitivities, class-groups and associated addressing, once captured and signed off by the relevant management, are mapped into policies and applied to the interfaces of the networking equipments deployed in an infrastructure, further detailed in section 3. However as previously highlighted, following deployment adjustments can be made which effectively break those deployed policies; an example of this would be the change of the application server IP address used within the prioritisation policy. In addition, over a period of time, applications usage can change: more users attempt to concurrently connect; the application is enhanced to support additional features not covered (e.g. by port address) within that initial policy; or in fact the application sensitivities might have been incorrectly identified within the initial requirements analysis and all of these areas must be captured and considered to ensure the intended SLA can be effectively delivered.

This research aims to develop an agent-based society, where agents gain an understanding of that network on which the applications are delivered their individual application sensitivities and how well the network supports these sensitivities, and where these agents negotiate with other, or with application-specific or peering-agents, to achieve a suitable *classification* setting, such that the application can match both an acceptable user perception, termed by the author's as Quality of Experience (QoE) or any higher level Service Level Agreement's (SLA) defined within the business or with third-party service providers (e.g. network provider). This is the correct and "best-effort" prioritisation, where best-effort is the ability to deliver the traffic for which an agent is negotiating within the constraints of its classification sensitivities and SLA, *but no better*, as this is resource that other agents must bid for as part of their outcomes.

This research will investigate adaptive learning techniques, including, but not exclusive to: neural; Genetic; and Bayesian networks; and consider adaptations of Swarm Intelligence models. Whilst the prospect of autonomous packet-based agents, for example: Reynold flocking boid's or Dorigo ant-hill colony algorithm, with outcome-based tasking, flocking throughout the network gaining that knowledge required to deliver a positive outcome, is fascinating, the current accepted view that Swarm Intelligence relies on dumb agents that select the most attractive free task, and that they do not typically negotiate for tasks, or in the coordination of these tasks as demonstrated by [3], which would appear to be a constraint requiring further investigation.

It is therefore proposed that this research focus on the delivery of an Application Specific Interface Card (ASIC) hardware based machine learning mechanism, which provides QoS based, not on human definition but actual application specific requirements against real-time network resource availability, and that this is then extended to provide inter-agent negotiation through peering arrangements of available resources. A major focus of the research will be that of "reuse", where reuse is defined as an activity that focuses on the recognition of commonalities of systems within and across research domains, e.g. should there be a requirement to provide the agents with an understanding of delay's occurring within the network, then an existing mechanism, such as an RSVP PATH message, is investigated for suitability or adaptation prior to any proprietary implementation.

II. Problem domain

End-users, of an application or system, require a positive Quality of Experience (QoE) in their interaction with technologies. Whether through: an operationally efficient system; ease of use, intuitiveness; interface design, functionality or aesthetics; or a new technology, such as; the quality of a High Definition (HD) TV service. In the field of data communications our focus as service and communications' providers is typically on the delivery of such services, within a set of Service Level Agreements (SLAs). These SLAs require consideration in a number of areas: from the applications or services being delivered; the infrastructure over which these services will be delivered; through interaction with other operators or applications, which may contend the delivery network; the skill and experience of the network support staff; right up to the selection of Customer Premises Equipments (CPE) including Personal Computers (PCs), any required user training, and ongoing support. All of these items can not only affect the delivery of this positive experience, but likewise have an impact on the operational expenditure (OPEX) of the provider, should it be deemed viable to deploy more or bigger supporting infrastructures to achieve this required experience. One of the biggest issues faced, not just by Communications Providers (CPs) but by enterprise IT departments, is the management of user contention, between the applications and services being delivered and that bandwidth available, particularly given the trend towards converged (data & voice) networks. The continued deployment of QoS mechanisms, introduces a number of reflections:

1. Prioritization mechanisms essentially provide "managed unfairness". It could be said that they, "rob Peter to pay Paul", that is they allocate applications a percentage of the available bandwidth, or apply queuing mechanisms, which during peak-demand periods, the service of such designated applications can negatively impact those other applications, allocated to lower priority queues or which have restricted bandwidth access imposed.

2. Whilst current QoS mechanisms capture the fact that applications do not typically require all 100% of the bandwidth allocated, 100% of the time, which allows available resource to be accessed by applications of a lower priority. These current QoS mechanisms are prone to ongoing change management issues due to their dependence on identifying the application(s) being classified by metrics such as: source/destination IP address; and IP port number. That is, should these identifiers change, as they typically do in enterprise environments and the change management procedures in place do not capture and incorporate these requirements within the classification policy, then the policy will break down.

3. The deployment of QoS mechanisms, whilst being done to manage the requirement for further bandwidth, can still result in escalating OPEX for organisations through the requirement to supply relevant skill-sets and mechanisms to visualise, monitor, trend, analyse, troubleshoot, and resolve issues. The delivery of a true end-to-end (E2E) management process, for most SME organizations is still:
    a. complex to design & configure
        i. ensure all system areas are incorporated (from ISO layer 1-7; client; server; supporting Infrastructure)
        ii. ensure all user groups perception of the solution are the same
        iii. multi-vendor environments requiring distinct language & even revision specific skill-set (e.g. IOS, JUNOS, v11.x, v12.x)
    b. complex to manage
        i. ongoing (change) management of deployed

policies and associated documentation
c. complex to visualize
   i. those policies deployed
   ii. define technical & management reports of sufficient depth
d. can be extremely expensive to purchase and maintain
   i. requires dedicated, trained personnel to ensure full ROI

The author is aware from his own experience, both as a technologist and as a team manager, that whilst it can be easy to define a human as an expert, through certification for instance, it becomes increasingly difficult for humans to retain that initial expertise through memory alone. We rely on memory to perform an astonishing variety of tasks in our everyday lives and this loss can be attributed to a number of conditions in a work environment:

a. Advancement in career typically results in a promotion away from that initial area of expertise, resulting in advancement in those new areas, but a loss of proficiency in those original skills. This is typically true of engineers who are promoted into management positions who must then lead other engineers in the delivery. The more time an engineer spends in this role, the more they find their hands-on skills diminish. A lot of knowledge engineers gather over the course of their tenure is held privately, this can result in issues regards a full and complete analysis of implemented policies should that engineer be unavailable.
b. All engineers' will reach a time in their life when earlier instinctive knowledge is lost. Marshall [1] highlights that irrelevant or out-of-date information needs to be pruned from the memory, because if humans remembered every detail, a condition known as "hyperthymestria", then this would result in mass confusion. Schacter in his book "The Seven Sins of Memory", which was the basis for Marshall's article in the New Scientist, details research into the ways humans forget and why, which can be summarised as (the sins):
   i. Omission (forgetting) - which includes; *Transience*, a gradual loss of memory in both short and long term memory; *Absent-mindedness*, a breakdown between attention and memory, either at the encoding or retrieval stage; and *Blocking*, a failure to retrieve or access deeply encoded information [4]
   ii. Distortion in contrast is responsible for; *Misattribution*, forgetting the source or assigning the memory to the wrong source of the information; *Suggestibility*, the implantation of memories; *Bias*, misremembering or influence of memories as a result of current knowledge, emotions or beliefs.
   iii. Intrusion responsible for *Persistence*, memories that should be forgotten but cannot be forgotten characteristically those resulting from strong emotional experiences.

Therefore, if machine learning was applied, not just to those tasks appropriate to the configuration and control of traffic (e.g. Cisco CCNA/CCNP/CCIE, Juniper JNCIA/S/E or similar), but to the business and technical applications analysis and classification, then whilst those skills remained relevant, memory loss would not be a factor in the system forgetting its goal orientated tasks.

Therefore this research looks to provide consistent and ongoing management of complex networks; which typically require Quality of Service (QoS) policy application, including, but not exclusive to:

- requirements analysis
- application classification and prioritisation
- policy definition & consistent management
- resource allocation
- network access control mechanisms
- topology, delay, learning and ontology engines to support automated inference
- negotiation, reputation & repudiation
- human visualisation reporting

III. PROPOSAL

In a traditional network environment, the policy rules, once defined from the business & technical analysis stage, would be applied to a hardware device such as a router depicted in Figure 3.1, these policies are now exposed to those issues presented within this paper, and for which the authors' now define their proposed mediation.

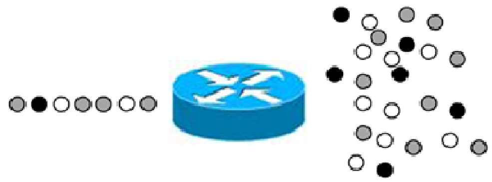

Figure 3.1. Typical Hardware Implementation

It should be pointed out that the *agents-unit*, used in this paper are adaptations of an "A Reflexive Autopoietic Unity" (Figure 3.2), presented by [5] in their paper "Organizations as Self-organizing and Sustaining Systems: A Complex and Autopoietic Systems Perspective". The "Autopoietic Unity" was originally presented by [6] in their book a "The Tree of Knowledge", and here they make the distinction, highlighted by Goldspink to the author [7], that an Autopoietic Unity demonstrated in Figure 3.2, is self-reproducing. However the author's do not contend that the agent-unit, based in-silico, has or even requires self-reproduction of its components, purely that the representation as shown in The Tree of Knowledge allows for a simple yet aesthetic representation, but this adaption by the authors' to demonstrate their "learning-process" has been done in such a way as to ensure no confusion with the original representation or any biological inference.

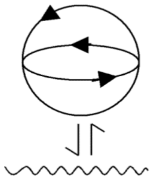

Figure 3.2. Autopoietic Unity [4] from [5].

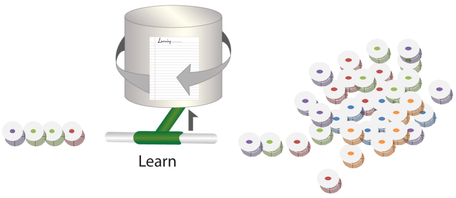

Figure 3.3. Agent Performs Supervised Learning

To overcome those issues identified, the adaptive synthetic socioeconomic agent would initially acquire data from the network, represented by the network cable and tap. The arrow represents that the agent now completes a comparison of that manual policy statement reasoning against its own knowledge base, indicated by the symbol: . At this stage the explicit IF *x* THEN *y* actions based on that manually designated service-policy, represented by: , are completed and the agents uses internal processes (**Learn**): Egocentric representation - what is my local environment (e.g am I a router or switch and what interfaces do I have?) and; Topographic representation - what other devices are in the environment (routing map) to establish that manual outcome against its proposed outcome, thus establishing an experience of the positive/negative outcomes of this matching. A *Prediction System* evaluates the proposed outcome, this outcome then being compared to that actual episodic outcome, such that an error can be established, from here the system adapts to develop its outcome. A *Meta-prediction system*, looks at how well the system is learning, not the prediction of the outcomes of proposed actions, but its progress along the prediction error curve. To ensure that any service impact during this learning process is limited it should be possible to mirror (port-mirror) that inbound traffic such that the router processes *live* traffic, whilst the agent processes a *copy* of that same data.

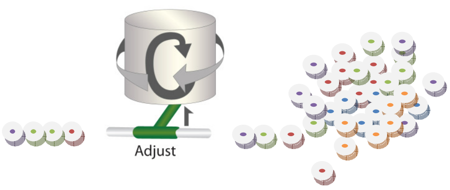

Figure 3.4. Agent Performs Unsupervised Learning

In Figure 3.4 the agent is now (**Adjust**)ing internal metric tables, represented by the internal arrow, such that it can remark, for example: the Differentiated Services Code Point, of those packets to be transmitted in a way that is believed will be beneficial to that traffic for which it is responsible. It performs implicit (**Reason**)ing, which could be weighted such, that whilst in a training mode, the learning and outcome was of more importance that the delivery of an outcome within any perceived budget (e.g. round-trip-time). In addition the agent should be updating topographic and ontological representations and potentially completing human visualisation (e.g. setting snmp traps and outputting for management reporting). However at this stage the agent is working in isolation to other agents and still completes a positive/negative outcome matching against that manual policy definition.

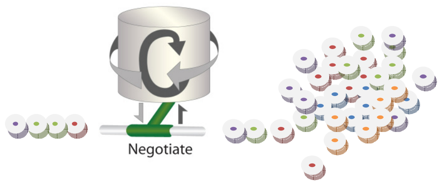

Figure 3.5. Agent Performs Reinforcement Learning

Figure's 3.5 & 3.6 show negotiation taking place between agents (**Negotiate**), such that resource allocation can be transacted, agreements made and following the transaction a reputation (positive or negative) achieve, allowing (**Grow**)th.

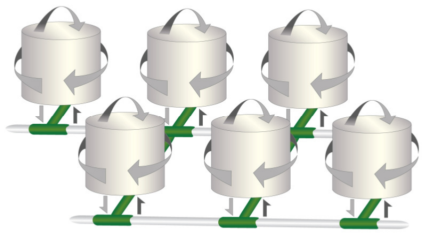

Figure 3.6. Inter-agent Negotiation, Reputation and Repudiation

The agents now display an Inference, the ability to connect existing knowledge, within the knowledge store, to create new knowledge through the chaining of existing rules or patterns [1], allowing a classification or ability to use data, to create knowledge, and not just be a digital filing cabinet. Finally the system reaches a state of *Formal Operations* [8] displaying an emergent behavior [9]. Emergence is the property in which the interaction of the parts creates some greater property of the whole which cannot be fully explained from the measured cause-and-effect behaviours of the components. The system is able to think logically about potential events or abstract ideas, can handle errors or mismatches and can build further on its semantic dictionary, constructing a library of "what-if" patterns for yet-to-happen (anticipated) events. This formal operating state, defined by Piaget [10], see Figure A.3.1.at appendix A, and which according to research by Bradmetz, few humans achieve, draws parallels with the work of

Artificial Life, which analyses and creates lifelike organisms in digital or robotic form. This Constructionist approach to education, where learning takes place inside the learner rather than being imposed by the teacher [11] aligns with the authors' goal of the system starting to self-organise.

IV. FUTURE WORK

Having established the framework on which the research will be based the authors present the following as further research milestones:

1. Experimentation and Simulation – This next paper will detail the work undertaken to present that observed traffic data to an artificial intelligence, machine-learning, application. In particular the authors are keen to demonstrate that any hypothesis of "optimization" should be based on an analysis and comparison of that performance which existed a priori and any associated proof be based on well formed and accepted methodologies. To this end the selection of the "learning-function" will be supported by appropriate statistical modeling to ensure that any experiments::
   - conform to Shewhart's rule [12] to preserve all the evidence in the original data. This is important should the dataset, which [13] refers to as the *frame*, be used across experimental iterations, or multiple testing software' or new elements within that data are found in later experiments which can then be tested to ensure any comparison made to previous results is supportable
   - an indication of what randomisation was selected
   - an indication of what the results refer to and the robustness of the conclusions
   - how the authors propose to use these conclusions to support further experimentation

This paper will then progress to describe those mechanisms required to support that proposed experimental & simulation environment, shown in Figure A.4.1 at appendix A, which includes:

- Generation – the creation of that data which will be input to the "learning mechanism"..
- Capture – how that network traffic data generated can be captured such that it can be analysed to ensure that it is exactly as sent (generated).
- Benchmark – benchmarked regards that performance provided by the system at that time, regards: load; transaction rate; pre-QoS; post-QoS; QoS policy applied; learning-mechanism; etc.
- Learning-mechanism – the learning types: neural; Bayesian; Kohonen; Hopfield; Boltzmann etc that allow for an optimum outcome
- Meta-prediction – how well is system the learning? As the system analyses patterns, triggers' or recall of other patterns are invoked. The time required for each iteration, and the number of iterations per pattern before the system to reaches an optimum (local or global) outcome, such as a point in the error curve, which can be measured and analysed. An additional reflection being what didn't work, as well as what did work. Later research papers will then progress those areas such as:

- Agent language – the language required for the agents to communicate
- Agent protocols – the elements which the agents adopt to ensure the global outcome (e.g. normalization) can be achieved over any local outcome requirements. This should include: non-repudiation; reputation and consider that social aspect of the agent society., that is those elements critical to ensure proper interaction to enable that "global" outcome (e.g. normalisation of the network environment). Whether the agents have, require or will achieve characteristics, and what those characteristics are. Could there be the equivalent of spectrum disorder (e.g. an autistic agent, which demonstrates those well known human autistic traits as [adapted from 14]:
   a. insistence on sameness; resistance to change
   b. difficulty in expressing needs
   c. difficulty in communicating with others
   d. unresponsive to normal teaching methods
   e. sustained odd behaviour
   f. obsessive attachment to objects (resource)
   g. apparent over-sensitivity or under-sensitivity
   h. noticeable over-activity or extreme under-activity
   i. non-responsive to cues

- Experimentation control – those controls and verification metrics suitable to support the results of the simulation as being correct and consistent
- Representation lexicon – those characters adopted for the representation of the work

Figure A.4.2. at appendix A, demonstrates a proposed distribution and agent type which is summarized in Table 4.1 below, and further detailed at A.4.1., Appendix A:

Agent Type	**Description**
Type 1 agent	Agent, typically software based on a PC/laptop device, resides in an internal Autonomous Agent Domain (iAAD). Communicates with Type 3 or 4 agent its location potentially via mechanisms such as its DHCP registration. Has local mechanism that records traffic sent & received and the delays seen in these transmissions.
Type 2 agent	Agent, typically software based on a telephony device, resides in an internal Autonomous Agent Domain (iAAD). Communicates with Type 3 or 4 agent its location potentially via mechanisms such as its DHCP registration. Has local mechanism that records traffic sent & received and the delays seen in these transmissions.

Type 3 agent	Agent, typically software based on a layer 2 (switch or access point) device, resides in an internal Autonomous Agent Domain (iAAD). Has local mechanism that records traffic sent & received and the delays seen in these transmissions. Type3 agents communicate full or local summary topology maps to Type 4 agents. Type 3 agents would typically have policing controls which allow the NBAR recognition of Scavenger services and take appropriate action (e.g. drop), of value where no Type 2 agents exist. During initial research the default mechanism is to pass all data traffic marked as COS/TOS 0 and all voice traffic as COS/TOS 5, this is to allow focus on the learning and optimisation with Type 4 devices, however further research will be focused on providing agent negotiation for Type 3 agents. It is feasible that a switch could perform all the duties of a layer 4 router. It is yet to be defined whether such a switch performing this function would be counted as a Type 4 agent device.
Type 4 agent	Agent, ASIC hardware based on a layer 3 (router) device, resides in both an internal Autonomous Agent Domain (iAAD) and if connected to a Communications Provider Type 5 agent an external Autonomous Agent Domain (eAAD). Type 4 agents (single or High Availability (HA) pair) are at the core of this current research. The main components of a Type 4 agent are proposed as: Topology engine: for internal mapping local interfaces and of the agent-based network, by collating all Type 1, 2, 3 and other Type 4 agent locations, potentially via existing mechanisms, such as: those devices DHCP registration; spanning-tree; or routing updates. This engine also collates all Type 1, 2, 3, 4 traffic sent/received/delays transmission reports such that internal E2E can be recorded. In addition it should be defined whether Type 4 agents collate summary information from locally attached Type 5 device(s) which provide an E2E baseline on which negotiations can be based, or Type 4 devices create a tunnel through the Type 5 devices for this purpose. Type 4 agents are the main focus of the initial research and it is currently still to be proved via simulation whether the agent performing their own proprietary communications and topology map building is optimal, or whether this topology is built by participating/listening to local routing protocol update information (e.g. OSPF, IGRP, iBGP, etc updates).
Type 5 agent	Agent, ASIC hardware based on a layer 3 (router) device, resides in both an internal Autonomous Agent Domain (iAAD) for Intra-service communications and if connected to a CPE Type 4 agent, or other Communications Providers Peering Partners an external Autonomous Agent Domain (eAAD). It is proposed that Type 5 agents are Communication/Service Provider agent devices which are able to either: a) consolidate, summarise and distribute topology maps, not only of their own internal topology and delay, but can consolidate and summarise that detail of their attached customer base b) or allow the construction of Type 4 agent tunnels for this purpose.
Type 6 agent	Agent, ASIC hardware or software based on a server device, or per application software agent install, resides in an internal Autonomous Agent Domain (iAAD). It is proposed that Type 6 agents are responsible for: a) the administration of E2E agent negotiation on behalf of the applications services installed b) they are to record and summarise data relevant to delay resulting from excessive internal processing

Table 4.1. Agent Architecture Summary

V. CONCLUSION

The authors conclude that this paper presents a cohesive programme of research intended to support, initially enterprise organisations in the requirements specification, deployment and ongoing service management of quality of service policies, but that eventually this methodology could prove scalable enough to support service provider of Internet environments.
Architecture Summary

VI. APPENDIX A

The supporting figures, diagrams and tables for this paper, can be found at: <http://www.imss.rdg.ac.uk/Publications/DLeggeABadii2009AppendixA.pdf>,
whilst the original un-summarised paper can be found at: http://www.imss.rdg.ac.uk/Publications/DLeggeABadii2009Full.pdf.

VII. REFERENCE

[1] Marshall, J., (2008), Unforgettable, New Scientist, 16th January 2008, page 30

[2] Heinanen, J., Baker, F. Weiss, W., Wroclawski, J., (1999) Request for Comments: 2597, Assured Forwarding PHB, IETF Network Working Group, June 1999, http://www.ietf.org/rfc/rfc2597.txt, (Accessed 17th February 2009).

[3] del Acebo, E., and de la Rosa, P., (2008), Bar Systems: A Class of Swarm Intelligence Optimization Algorithms, AISB Conference, Aberdeen (April 2008).

[4] Wallisch, P., The Seven Sins of Memory, (2007), [Internet], http://lascap.de/The_Seven_Sins_Of_Memory.pdf, (Accessed 20th February 2008, not currently online).

[5] Goldspink, C., (2008), Conversation relating from his presentation: Agent Cognitive Capabilities and Orders of emergence: Critical Thresholds relevant to the Simulation of Social Behaviours, AISB '08, University of Aberdeen (April 2008).

[6] Humberto R. Maturana and Francisco J. Varela (1987), The Tree of Knowledge, reprinted by arrangement with Shambhala Publications, Inc., Boston, www.shambhala.com.

[7] Goldspink, C., & Kay R., Organizations as Self-organizing and Sustaining Systems: A Complex and Autopoietic Systems Perspective, International Graduate School of Management, University of South Australia, joint impact, email from author [5] & meeting held in Guilford, October 2008.

[8] Eysenck, M., (2001), A2 Psychology: Key Topics, London: Psychology Press

[9] Highsmith, J., A., III, (1999), Adaptive Software Development: A Collaborative Approach to Managing Complex Systems, Dorset House Publishing, New York.

[10] Wagner, K., Van, Background and Key Concepts of Piaget's Theory, Stages of Cognitive Development, http://psychology.about.com/od/piagetstheory/a/keyconcepts.htm, (Accessed 17th February 2009).

[11] Challoner, J., (2002), Artificial Intelligence, London: Doring Kindersly

[12] Shewhart, W. A., Economic Control of Quality of Manufactured Product, http://en.wikipedia.org/wiki/Shewhart, (Accessed 8th September 2008).

[13] Deming, E., W. (1975) On Probability As a Basis For Action. *The American Statistician,* Volume 29, No.4., http://www.stat.osu.edu/~jas/stat600601/articles/article1.pdf (Accessed 19th February 2009).

[14] Characteristics of Autism, (2008), [Internet], Autism Society of America, http://www.autism-society.org/site/PageServer?JServSessionIdr012=sqyplynxu1.app24a&pagename=about_whatis_char, (Accessed 8th September 2008).