Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Control Plane Algorithms targeting Challenging Autonomic Properties in Grey Systems

E.C.L. Watts, A. Taleb-Bendiab
School of Computing and Mathematical Sciences
Liverpool John Moores University
Bryom Street, Liverpool L3 3AF, UK
E-mail: cel@watts2885.freeserve.co.uk, A.Talebbendiab@livjm.ac.uk

***Abstract*—Autonomic self-configuration is an essential mechanism underpinning agile service provision, allowing predictive or non-predictive grey system domains to transform their underlying system behaviour. However, much more work is required to improve our understanding of the control plane algorithms for these systems to prevent challenging autonomic system properties from impacting service provisioning. Current research focuses upon three key 'planes'; the knowledge plane, the control plane, and the data plane. This work proposes and presents a prototype for an autonomic self-configuration control framework, comprising of a Control Plane Control Protocol, and optional algorithmic overlay. The framework bridges the three networking planes, with the knowledge plane governance selecting the required configuration data, and the control plane controlling the negotiation and dissemination of the payload, through a generic control protocol, and an optional control plane overlay. The new framework can form part of a protective system infrastructure. A novel aspect of this framework is that it can address challenging autonomic properties through the control plan overlay. By autonomically controlling how a networked appliance responds to an external stimulus; a permanent or transitory change may ensue or self-configuration can be prevented. These framework attributes are evaluated using a prototype to demonstrate algorithms assessing actor hybridization and blocking avalanches of changes that may result in unrestrained, rapid actor hybridization.**

***Keywords*—self-configuration, framework, control plane, algorithmic overlay, avalanche and hybridization algorithms**

I. INTRODUCTION

Outsourcing day-to-day network management to new forms of autonomic approaches, techniques and toolsets bestows many benefits and challenges on systems. It represents one approach to the intimidating problem of relentlessly, rapidly and agilely managing vast quantities of heterogeneous configuration data. But configuration data dexterity means that system lifecycle state is increasingly indeterminate and uncertain, with past history often shaping future behaviour. System inter-dependency can change as a result of autonomic self-configuration therefore agile, context-sensitive control mechanisms are crucial; addressing challenging autonomic system self-configuration behaviour. In particular three significant challenging properties; system hybridization, avalanche indicators, denoting major autonomic self-configuration perturbations and avalanche dams require additional research from the perspective of an autonomic actor. Heterogeneous systems can be compared to varied soil types, with networked appliances (actors) the diverse seeds. These seeds can cross-pollinate through autonomic self-configuration resulting in actor hybridization. In contrast avalanche properties express the sudden arrival of an overwhelming number of self-configuration requests which can result in uninhibited oscillations. Control mechanisms need to identify and take account of the potential impact of these oscillations; controlling their ebb and flow through the use of 'avalanche dam' and hybridization control plane overlays. This paper presents a number of novel contributions with an autonomic self-configuration control plane architecture and framework addressing a generic requirement for controlled self-configuration provision, by combining a control protocol called HELP (High-level Extensible Learning Protocol), with an optional control plane decision-making overlay. Operating as a logical network service; overlay(s) support a range of decision-making algorithms across the configuration and feedback stages of a self-configuration. The algorithms take into account these challenging autonomic properties from the perspective of an actor. An actor may form part of a flat or hierarchical infrastructure, and their influence may be localized or extend to inter or intra system domains. The rest of the paper is organized as follows. Section II discusses the problem definition along with the research motivation. Section III explains the proposed new framework and control plane overlay and Section IV presents the prototype evaluation. Finally, Section V concludes the paper prior to discussing future work.

II. PROBLEM DEFINITION AND MOTIVIATION

Autonomic actors within system domains need suitable 'generic' control mechanisms to maintain evolving service provisioning. These actors may operate in one or multiple system domains. Control mechanisms are crucial for the future, impacting three key problem domain issues, namely (i) system infrastructure challenges such as system 'greyness', (ii) challenging autonomic properties resulting from self-configuration, and (iii) overlay algorithms. Hybrid infrastructures such as cross-plane infrastructures are increasingly favored over a legacy, flat, centralized paradigm. In future the Internet may be replaced, or co-exist with a new "Internet3" [7]. DARPA [1, 7] is funding research into "overlay architectures" and "mechanisms" to deploy and customize services in increments. Whereas [2, 6] are examining control plane support for present network interoperability and data being sent on-demand while [23] are examining a free form infrastructure. Legacy system integration is currently a particular problem; policies to safely and fittingly configure legacy and hybrid systems are needed. Active research into system of system infrastructures includes safety policy research. An aircraft "flight control system" [11] allows a controlling policy to define domain 'greyness'. A pilot can choose to push a specified operational domain boundary, meeting progressive resistance as the domain state line is exceeded. Trying to define this safe domain behaviour necessitates specifying safe and unsafe states; but domains of operation are not necessarily contiguous, can evolve over time and depend upon context thereby underlining the role of an actor in a system. Generally these architectures are proving a challenge, incurring increased levels of control traffic.

In comparison challenging properties as a result of self-configuration include demanding hybridization and avalanches of transformation. [16] considered hybridization in terms of "Bio-Inspired Communication Systems". Actor components can comprise many self-configurable values, categorized as fluctuating, or variation values. A fluctuation value represents self-configuration data changing condition according to an actors' environment. These values can work in one environment, but not in another, whereas a variation value represents self-configuration data that changes permanently within an actor and so causes an actor to mutate from its' original configuration data state. Care must be taken with these values especially if transformation requests result in unrestrained increases in control traffic and system change. Suitable 'avalanche dams' are needed to control these fluctuations; deciding whether they pass on the control traffic to other actors or absorb the fluctuations. [13, 15] categorizes avalanche dams in terms of catching, deflecting or a combination of both dams. Control mechanisms need to consider the composition of the control traffic; are the fluctuations on a micro or macro scale and do they contain conflicting or redundant request? Much research also focuses on algorithms, for instance considering hybrid reinforcement learning algorithms [20]; avalanche algorithms [13, 15] exploring control mechanisms to underpin the construction of suitable physical avalanche barrier dams; hysteresis, backlash and dead zone algorithms [3] examining potentially undesirable patterns of displacement; Paerto-based algorithms [19], [21], finding an optimal solution, sustaining knowledge-plane decision-making or considering global goals; and other algorithms [4, 12] self-organizing, load balancing, establishing contextual values, [10] pruning or [17] addressing multiple domains of science. Section III now describes the proposed new framework and control plane overlay.

III. PROPOSED DESIGN

Current research focuses upon three key 'planes'; a knowledge plane, a control plane and a data plane. This paper proposes decoupling and extending the control plane, to provide extensible support for a range of self-configuration infrastructures. It proposes an autonomic framework and overlay targeting the control plane as part of a novel new architecture, allowing peer-based heterogeneous device collaboration, and addressing a range of technological platforms. This proposed architecture is drawn as a free form architectural diagram in Fig. 1 [24, 25]. It is a situation-aware system, showing four key components including; the *knowledge plane, control plane, data plane* and the *collaborating actors/ components* Decoupling the data plane from the control and knowledge planes allows the architecture to negotiate and configure a variety of current, legacy and future protocols and control traffic such as configuration or feedback data. The framework supports *collaborating components* co-located on an *actor,* or located across one or several actors.

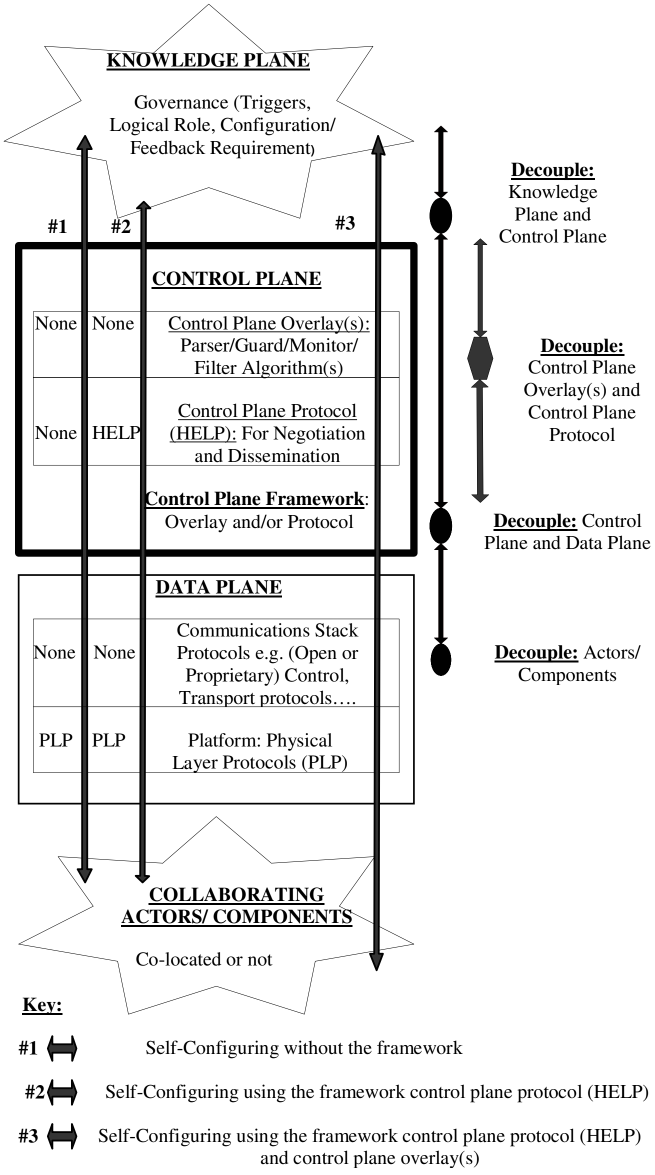

Figure 1. Control Plane Framework and Overlay Architecture

Inter-plane as well as cross-plane self-configuration is supported. Self-configuration 'needs' activate the HELP (High-level Extensible Learning Protocol)-controller service, which coordinates self-configuration processes, including the identification of the need, such as a Learner asking for specific self-configuration data, or a Tutor requesting feedback, indicating whether or not configuration data proved suitable.. The new control plane framework proposes two self-configuring components, a self-configuration control protocol called HELP (described in Section A) and an optional control plane overlay (illustrated in Section B), describing some novel algorithms targeting challenging behaviour.

A. Control Protocol for Negotiation and Dissemination

The control protocol forms a template, allowing actors to dynamically negotiate a variety of payloads using a context-sensitive array of protocol fields and negotiation stages. Protocol field and stage order is significant, allowing an actor to sift part or all of a prospective payload. Negotiations can be staged, with the protocol allowing up to three stages of negotiation. Any device can act as a logical Learner or Tutor. During a session feedback can be agreed and sent using a single stage optional overlay; this feedback may be exchanged immediately after a session completes, at a pre-agreed time or once a trigger event is met.

B. Control Plane Overlay Algorithm

Overlays provide the controlling data negotiation and dissemination strategy. A session algorithmic overlay spans every negotiation stage whereas a multi-stage algorithmic overlay spans a flexible subset of session stages. A single stage algorithmic overlay targets one particular negotiation stage. One or multiple stage overlay algorithms use control plane protocol fields to provide algorithmic inputs or outputs. This approach allows one stage and/or multiple stage algorithmic overlay(s) to target a request, a response or both message types. For instance an avalanche dam algorithm establishes an AD (Avalanche Dam) value, identifying the size of threshold (barrier) needed to block a self-configuration in an actor.

1. $SV = SR * SS$

Figure 2. Self-configuration Velocity Algorithm

Equation (1) establishes SV, the self-configuration velocity of fluctuation by multiplying SR, the required rate of self-configuration, by SS, the size of a prospective self-configuration (LearnerStateHybridizationSize (equation (3) in Fig. 4)). The term SR in the equation is typically assumed to be flexible, being set by the knowledge plane.

2. $AD = (SV/SD * 100)$

Figure 3. Avalanche Dam Algorithm

Equation (2) divides SV by SD, the self-configuration density of preceding hybridizations (LearnerStateHybridizationDensity (equation (4) in Fig.5)). This value is multiplied by 100 to establish AD, the avalanche dam value. No change takes place if an actor sets a self-configuration threshold value greater than AD. All three values, SR, SD and SS can change over time as an actor evolves

3. learnerStateHybridizationSize = sum(componentsAffected * ComponentImpact) n

Figure 4. LearnerStateHybridizationSize Algorithm

Equation (3) assesses the size of a prospective change. The granularity of these components may vary and can be categorized as varying or fluctuating depending upon if a change results in a transitory or permanent transformation in an actor. For instance, a sensor may change from a monitoring to a monitoring and analysis sensor. Whereas Equation (4) involves assessing the Learner hybridization density; it measures the size of change (summation of SS) over a period of time. The knowledge plane assigns context-specific values to a time-bounded self-configuration period.

4. learnerStateHybridizationDensity = size of change (sum SS) over a period of time (T)

Figure 5. LearnerStateHybridizationDensity Algorithm

This paper presents a framework and control plane overlay architecture, as shown in Fig. 1. The control protocol can be used on its own, or combined with one or multiple overlays, during a staged self-configuration session. The framework control plane overlays, under the knowledge plane governance can control self-configuration avalanche oscillations and actor hybridization. The control plane overlay can thereby address challenging autonomic self-configuration properties. Section 4 evaluates the framework control plane overlay agility in terms of hybridization and avalanche dam algorithms.

IV. PROTOTYPE EVALUATION

This section presents an initial evaluation of the proposed framework and its support for agile control plane overlay algorithms. A small scale system of two logical peers is used; one peer performs a role of a Tutor, and another a role of a Learner. They join a default, 'self-help' peer group. The prototype operates over a JXTA framework, with sockets streaming data between peers. An elapsed time for each negotiation is recorded, allowing assessments of scenarios, with and without the framework and control plane overlay. Avalanche dam algorithms establishing avalanche velocity by Equation (1) and an avalanche dam value by Equation (2) and hybridization algorithms determining the size of self-configuration by Equation (3) and the self-configuration density by Equation (4) are examined through a series of illustrative scenarios. To summarize these scenario results Fig. 6 depicts the comparative throughput rates graphically.

A. Knowledge Plane APIs

The knowledge plane governance APIs specify the self-configuration rating, component sizes and time period values. For instance Table 1 specifies an immediate change has a value of 10 whereas a change that can take place sometime/ anytime has a value of 2.

TABLE I. SELF-CONFIGURATION RATINGS (SR)

SR Description	**SR Value**
Now/ immediately	10
As soon as possible	5
Sometime/ anytime	2
Not now	1
Never	0

Table 2 details the component values. For this scenario Learner functionality (100%) comprises of 8 components. A Tutor may offer none, one, several or all of the requested components, so this value is calculated after a Tutor responds to a request.

TABLE II. SELF-CONFIGURATION SIZE (SS)

Learner Component	SS Value (as % of 100 %)
1	50
2	20
3	10
4	5
5	5
6	4
7	3
8	3

Table 3 assigns a value to particular time period, as dictated by the knowledge plane.

TABLE III. SELF-CONFIGURATION SIZE (T)

Time Period	T Value
Hour	SumSS/ 50
Day	SumSS/ 40
Week	SumSS/ 30
Month	SumSS/ 20
6 Month	SumSS/ 10
Year	Sum SS/ 5
None previously	Sum SS

Finally Table 4 shows threshold values controlling when a self-configuration is blocked or allowed.

TABLE IV. SELF-CONFIGURATION THRESHOLD

Threshold Dam	Threshold Dam Value
1	2000
2	1000

B. *Control Plane Overlay*

Up to 4 equations may overlay a Learner self-configuration. This control mechanism is structured as follows:

- Step 1: Knowledge plane API specifies resetting or retention of the sum of SS (self-configuration sizes). Allowing an actor to monitor self-configurations over a time period.
- Step 2; learnerStateHybridization Algorithm-Equation (3) *or* knowledge plane API specified
- Step 3; Self-configuration Velocity Algorithm-Equation (1)
- Step 4: learnerStateHybridizationDensity Algorithm-Equation (4) *or* knowledge plane API specified.
- Step 5; Avalanche Dam Algorithm-Equation (2)
- Step 6: Knowledge plane API specified Threshold Dam Value. If AD exceeds this Threshold Dam then the avalanche impetus for self-configuration succeeds and a payload is exchanged else a HELP session is blocked.

C. *Challenging Properties - Avalanche Dam Algorithms*

A Learner actor employs equations (1 and 2); using knowledge plane APIs to specify a SS (self-configuration size) of 50 and a SD (self-configuration density) of 20. The SR requires an immediate change resulting in a value of 10 and the Threshold Dam is set to 2000.

- Step 3: SV (500) = (SR (10)* SS (50))
- Step 5: AD (2500) = (SV (500) /SD(20) * 100)
- Step 6: Help Clarification session completes and a payload is exchanged.

D. *Challenging Properties – Avalanche and Hybridization Algorithms*

A Learner actor employs all four algorithms as part of its control plane overlay. One component is affected; assessed as 50% of the value of the whole actor. The SR requires an immediate change resulting in a value of 10, the Time period requirement is a self-configuration within a month and the Threshold Dam is 2000.

- Step 2: SS (50) = sum (1*50)
- Step 3: SV (500) = SR (10) * SS (50)
- Step 4: SD (20) = sum SS (50) / (T) (50/20)
- Step 5; AD (2500) = (SV (500) /SD(20) * 100)
- Step 6: Help Clarification session completes and a payload is exchanged.

E. *Challenging Properties – Cumulative Avalanches*

A Learner actor employs all four algorithms as part of its control plane overlay. Two components are affected; assessed as contributing 70% of the value of the whole actor. The SR requires a self-configuration as soon as possible resulting in a value of 5, the Time period requirement is a self-configuration within a week and the Threshold Dam is set to 1000.

- Step 2: SS (70) = sum((1 *50) + (1*20))
- Step 3: SV (350) = SR (5) * SS (70)
- Step 4: SD (30) = sumSS (70 +(Section D) 50 = 120) / (T) (120/30=4)
- Step 5: AD (1166.7) = (SV (350)/SD(30) * 100)
- Step 6: Help Clarification session completes and a payload is exchanged.

F. *Challenging Properties –Cumulative Self-Configuring Blocked*

A Learner actor employs all four algorithms as part of its control plane overlay. Six components are affected; assessed as 79% of the value of the whole actor. The SR requires a self-configuration sometime, anytime resulting in a value of 2, the Time period requirement is a self-configuration within a week and the Threshold Dam is lowered to 1000.

- Step 2: SS (93) = sum((1*50) + (1*20) + (1*10) + (2*5) + (1* 3))
- Step 3: SV (186) = SR (2) * SS (93)
- Step 4: SD (30) = sumSS (93 +(Section E) 120 = 213) / (T)(213/30 = 7.1)
- Step 5: AD (620) = (SV (186) /SD(30) * 100)
- Step 6: Help Clarification session is blocked.

G. Challenging Properties –Cumulative sumSS Reset

The scenario is the same as above except for the addition of Step 1 to reset the cumulative total of sumSS to 0.

- Step 1: sumSS=0
- Step 2: SS (93) = sum((1*50) + (1*20) + (1*10) + (2*5) + (1* 3))
- Step 3: SV (186) = SR (2) * SS (93)
- Step 4: SD (30) = sumSS (93) / (T)(93/30 = 3.1)
- Step 5: AD (620) = (SV (186) /SD(30) * 100)
- Step 6: Help Clarification session is blocked.

H. Comparative Graph

The graph in Fig. 6 shows comparative values for a HELP Clarification session with no control plane overlay, followed by values for scenarios depicting the control plane overlays for Sections C, D, E, F and G. Each scenario represents the mean value of 10 runs, with a successful session transmitting a payload of 27639 bytes and exchanging 29180 bytes in total whereas a blocked session resulted in 1072 bytes. The mean values for Sections (C, D and E) and Sections (F and G) are similar, implying that a HELP session with no overlay exchanges a successful payload more quickly than a HELP session with a control plane overlay but fails to block unwanted payload exchanges. The overlay allows this level of control, under the dictate of a knowledge plane API.

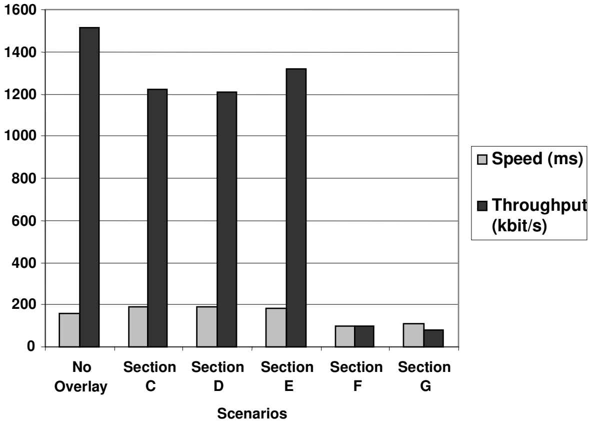

Figure 6. Control Plane Overlay Assessment Graph

A subsequent single test run based upon the overlay algorithms in Section G, but with a time (T) value of 6 months allows a payload exchange to proceed. For these tests the time (T) value is of particular note. It can be assigned a value by the knowledge plane API, as in this case, or can be derived from for instance a clock. This value can thus be tuned to both context and configuration data subtleties. This section evaluated the challenging properties of hybridization [20] and avalanche dams [13, 15]; increasingly prevalent ‘generic’ attributes of actors, using the new framework and optional control plane overlay as a potential approach to this problem. [1, 7] highlighted a growth in the complexity of supporting infrastructures and there is a growing requirement for subtle control mechanisms to allow an actor to develop overtime. The framework control plane overlay successfully allowed and blocked self-configuration payload exchanges. Further scenario work is needed to build on these initial results.

V. CONCLUSION AND FUTURE WORK

Autonomic self-configuration provides an approach to the challenging nature of complex, timely and agile configuration data provisioning in grey systems. Yet this approach is creating a ‘generic’ dilemma for service providers, incurring associated challenging properties such as unrestrained actor hybridization and uncontrolled avalanches of change. [1] is funding research into overlay architectures and mechanisms, operating across flat or hierarchical networks, which do not impinge on an underlying infrastructure whereas [7] discussed ongoing research work into a future “Internet3”, consisting of separate control and management domains. [17] described a wide range of topical algorithmic challenges including determining when parameters of local interactions evolve, the way macroscopic properties transition and how global properties relate to local properties. These challenges need further research.

This paper proposes a new architecture, as shown in Fig. 1. It is a situation intentional system, allowing knowledge plane governance of self-configurations through a knowledge plane API and control plane framework and overlay. A crucial attribute of this architecture is that it allows support for overlayed control mechanisms addressing challenging self-configuring behaviours such as actor hybridization and avalanches of change. An actor can thus overlay a range of algorithms using the framework and optional control plane overlay. The prototype allows an actor to assume the logical role of either a Tutor or a Learner; a role independent of an actor’s functional role. These roles are dictated by its governance (knowledge plane) as are the choice of algorithms. The prototype demonstrates the application of our algorithms, targeting these particular challenging autonomic behaviours.

Future work could include establishing further hybridization and avalanche characteristics from the perspective of system governance; determining whether or not a system domain experiences a ‘clean’ or ‘dirty’ type avalanche. A ‘clean’ avalanche ripples across all the actors in a domain leaving a similar trace behind whereas a ‘dirty’ avalanche results in partial or unpredictable actor self-configuration. Additional algorithms could gauge functional differences within a system, allowing system governors to deliberately introduce new actors into a domain to produce hybrid actors for a particular context. These algorithms would complement future work, discerning the interplay and inter-dependence between autonomic actors and the associated oscillation patterns.

Also system domains may contain metastable states, containing two or more substates. A metastable state can fluctuate significantly from domain to domain and so remains a challenging research arena. Hysteresis control plane overlay algorithms provide one potential approach to challenging autonomic properties in grey systems with [20] discussing policy hysteresis in conjunction with their hybrid reinforcement learning. Hysteresis and backlash actions require further research as these challenging attributes can cause an adverse effect or jarring reaction as a consequence of an autonomic self-configuration. Autonomic control mechanisms need to consider time lags, data-dependence and multi-state input. They also need to consider dead zone behaviour, resulting in an actor being unable to measure activity for a period of time.

Self-regulation/governance principles are beyond the scope of this paper but these principles are being addressed by other work [14]. In future for instance, the framework may be extended to react proactively to sensed or received information, such as the detection or prediction of an avalanche.

Finally knowledge plane applications providing static or runtime Monte Carlo Analysis within an actor could prove beneficial. Collated feedback could seed a beta-PERT distribution, to add a weighting factor to future configuration data fluctuation and variation values as well as the associated hybrid and avalanche algorithms.

[1] Amir, Y., Awerbuch, B., (principal investigators) Stanton, J., (co-principal investigators), DARPA/ITO grants, "A Cost-Benefit Approach to Fault Tolerant Communication and Information Access", *CNDS Grants – Tolerant Networks, Center for Networking and Distributed Systems*, Computer Science Department, John Hopkins University, Baltimore, http://www.dsn.jhu.edu/funding/tolerant_networks/, May 2000- September 2003

[2] Ahlgres, B., Eggert, L., Ohlman, B., Schieder, A., "Ambient Networks: Bridging Heterogeneous Network Domain", *16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC),* Berlin, Germany, September 11-14th, 2005

[3] Arpaci-Dusseau, A.C., Arpaci-Dusseau, R.H., "Information and Control in Gray-Box Systems", *ACM Symposium on Operating System Principles Proceedings of the eighteenth ACM Symposium on Operating System Principles*, 2001, pp. 43-56, ISBN: 1-58113-389-8

[4] Castro, M., Drischel, P., Hu, Y.C., Rowstron, A., "Exploiting Network Proximity in Peer-to-Peer Overlay Networks", 2002, *Technical Report MSR-TR-2002-82*, Microsoft Research, research.microsoft.com/~antr/PAST/location.pdf,

[5] Deval, M., Khosravi, H., Muralidhar, R., Ahmed, S., Bakshi, S., Yavatkar, R., "Distributed Control Plane Architecure for Network Elements", *Intel Technology Journal,* Volume 07, Issue 04, Published, November 14, 2003, ISSN 1535-864X

[6] Doshi, Bharat, JHU/APL "Response to RFI SN07-12 Assurable Global Networking", February 5, 2007, http://csc-ballston.dmeid.org/darpa/meetings/presentations/jQfxH8aC/JHUAPL.pdf, 09/07/07

[7] Durresi, A., "Designing the Future Internet", Key Note Speech, *AINA07, 21st International Conference on Advanced Information Networking and Applications Workshops/Symposium*, Niagara Falls, Ontario, Canada, 21-23 May 2007, IEEE Computer Society

[8] Ghavamzadeh, M., Mahadevan, S., Makar, R., "Hierarchical multi-agent reinforcement learning", (4/4/6), *Autonomic Agents and Multi-Agent Systems*, Volume 13, Number 2, September 2006, Springer, pp.197-229, ISSN: 1387-2532

[9] Gibson, T., Colonel, "Control Plane", DARPA Advanced Technology Office, 12/05/2003, http://www.arpa.mil/sto/solicitations/ControlPlane/gibson_brief.pdf, 09/07/07

[10] Greenwald, A., Boyan, J., "Bidding Algorithms for Simultaneous Auctions: A Case Study", *Autonomic Agents and Multi-Agent Systems*, Volume 10, Number 1, January 2005, Springer, pp. 67-89, ISSN: 1387-2532

[11] Hall-May, M., "Ensuring Safety of Systems of Systems A Policy-Based Approach", Setp. 2007, Phd Thesis, University of York

[12] Huebscher, M., McCann, J., "An Adaptive Middleware Framework for Context-Aware Applications", *Personal and Ubiquitous Computing*, Volume 10, Issue 1, January 2006, pp. 12-20, ISSN: 1617-4909

[13] Kuntz, M.C., Perkovic, O., Dahmen, K.A., Roberts, B.W., Sethna, J.P., Hysteresis, Avalanches and Noise, Computing in Science an Engineering, pp. 73-81, Volume 1, Issue 4, July 1999, ISSN:1521-9615

[14] Lamb, D., Randles, M., Taleb-Bendiab, A., "Monitoring Autonomic Networks through Signatures of Emergence", In Proceedings of the 6th International IEEE Conference and Workshop on the Engineering of autonomic and Autonomous Systems 2009 (EASe2009), San Francisco, USA, pp. 56-65.

[15] Leeds University, "The design of avalanche protection dams. Recent practical and theoretical developments", 1-DRAFT, SATSIE project deliverable, Report-The Design of Avalanche Protection Dams, EU Contract No EVG1-CT2002-00059, July 25, 2006, http://www.leeds.ac.uk/satsie/docs/satsie_d14.pdf

[16] Miorandi D., Yamamoto L., Dini P., "Service Evolution in Bio-Inspired Communication Systems", pp. 51-60, International Transactions on Systems Science and Applications Journal, (ITSSA), Volume 2, Number 1, September 2006, cn.cs.unibas.ch/people/ly/doc/soas2006-myd.pdf

[17] Papadimitriou C., "The Algorithmic Lens: How the Computational Perspective is Transforming the Sciences", *Guest Lecture at Liverpool University,* 4 March 2009

[18] Smidl, V., Prikryl, J., "From Bayesian Decision-Makers to Bayesian Agents", Edited by Czap, H., Unland, R., Branki, C., Tianfield, H., *Self-Organization and Autonomic Informatics* (I), pp.62-76, ISBN, 1-58603-577-0, ISSN 0922-6389, IOS Press

[19] Soh, L., Tsatsoulis, C., "A Real-Time Negotiation Model and A Multi-Agent Sensor Network Implementation", *Autonomic Agents and Multi-Agent Systems*, Volume 11, Number 3, November 2005, Springer, pp. 251-271, ISSN: 1387-2532

[20] Tesauro, G., Jong, N.K., Das, R., Bennani, M.N., "A Hybrid Reinforcement Learning Approach to Autonomic Resource Allocation", *The 5th IEEE International Conference on Autonomic Computing (ICAC06),* Dublin, Ireland, June 2006

[21] Thomas, R.W., Friend, D.H., DaSilva, L.A., MacKenzine, A.B., "Cognitive Networks: Adaptation and Learning to Achieve End-to-End Performance Objectives", *IEEE Communications Magazine*, Volume 44, Number 12, December 2006, pp. 51-57, ISSN: 0163-6804

[22] Verbeeck, K., Nowe, A., Parent, J., Tuyls, K., (Exploring selfish reinforcement learning in repeated games with stochastic rewards, 10/11/6), pp. 239-269, Autonomic Agents and Multi-Agent Systems, Volume 14, Number 3, June 2007, Springer, ISSN: 1387-2532

[23] Vicente, J., Rungra, S., Ding, G., Krishnaswamy, D., Chan, W., Miao, K., OverMesh: "Network-Centric Computing", Volume 45, Number 2, February 2007, *IEEE Communications Magazine*, pp. 126-133, ISSN: 0163-6804

[24] Watts , E.C.L., "A Self-Configuration Framework enabling an Overlay Control Plane for Autonomic Systems", February 2008, Phd Thesis, Liverpool John Moores University

[25] Watts, E.C.L., Merabti, M., Taleb-Bendiab, A., "An Extensible Self-Configuration Framework and Control Plane Overlay To Support Agile System Infrastructure", Volume 7, May 2009, Communications of SIWN, pp. 142-148