

An adaptive systems perspective on network calculus, with applications to autonomic control

Simon Dobson

Systems Research Group,
School of Computer Science and Informatics,
University College of Dublin,
Belfield, Dublin 4 IE,
Ireland
E-mail: simon.dobson@ucd.ie

Abstract: Autonomic communications systems must demonstrate behaviour that remains correct under a range of environmental conditions. In order to gain confidence that a system will behave as intended, it is advantageous to have a formal description of the expected behaviour that can be analysed and tested for compliance with different stimuli. We present an adaptive systems perspective on a recently-proposed formal analytic model of network performance. We suggest how this can be used to study the adaptive behaviour of systems, and to ensure that their adaptations maintain desired properties.

Keywords: autonomic communications; context-sensitive adaptation; network calculus.

Reference to this paper should be made as follows: Dobson, S. (2008) 'An adaptive systems perspective on network calculus, with applications to autonomic control', *Int. J. Autonomous and Adaptive Communications Systems*, Vol. 1, No. 3, pp.332–341.

Biographical note: Simon Dobson's research centres around adaptive pervasive computing and novel programming techniques, supported by an active portfolio of grants from Irish and European sources. He has published over 60 internationally peer-reviewed journal and conference papers, sits on several editorial boards and programme committees, and is programme Co-chair for IEEE ICAC 2008. He is a national director for ERCIM and serves on a number of national and European strategy panels. He holds a BSc and DPhil in computer science, is a Chartered Engineer and a Member of the BCS, ACM and IEEE.

1 Introduction

Autonomic communications offer enormous potential for offering more robust, reliable, responsive and performant systems. By allowing low-level details to adapt to changing circumstances, autonomic systems can potentially offer quality of service guarantees across a range of situations: ironically, such adaptive systems may be less variable to users and developers than traditional systems, as adaptations smooth-out variations.

Implementing this sort of adaptation poses major challenges for designers. Three critical problems stand out: identifying adaptations that can be performed on a system and their consequences for its behaviour; selecting the adaptation to apply under given circumstances and verifying that the chosen adaptive strategy meets the system's external

constraints under changing circumstances. The first problem has been widely addressed in the literature through a wide range of innovative techniques (see Dobson et al., 2006 for a recent survey). The second is also widely studied in the area of pervasive computing, and involves determining a situation from a range of sensed context (Coutaz and Rey, 2002) using uncertain reasoning and similar techniques. The third, verification problem remains significantly under-studied. Addressing it effectively requires developing models of complete systems that can be explored and studied to check compliance.

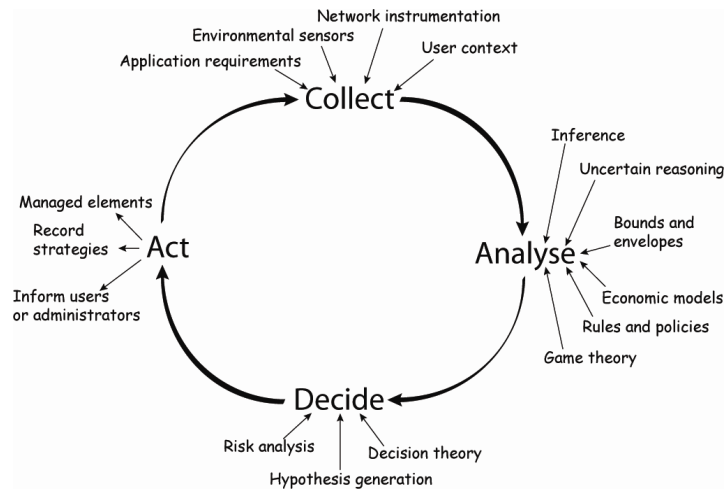
One recently-proposed model that is gaining traction is network calculus (Le Boudec and Thiran, 2001) which allows concepts from systems theory to be applied to the definition and analysis of communications systems. Network calculus allows bounds to be placed in several important elements of network performance, allows the impact of design decisions to be studied and provides a formal underpinning for implementation techniques such as IntServ and DiffServ.

Our purpose in this article is consider network calculus from the perspective of adaptive systems. That is to say, we are interested in expressing a space of possible adaptive behaviours and wish to study the selection and impact of these adaptations within the framework of network calculus. We present no new mathematics in this article, reserving the details for future work: our contribution here is to sketch, in a relatively informal way, an approach to modelling adaptive systems that may offer a promising avenue for analysing autonomic control.

Section 2 reviews the core challenges of autonomic systems and their formal modelling. Section 3 presents a description of a simple adaptive system and relates its adaptations to a model of its environment. Section 4 concludes with some directions for future work.

2 The core challenges of autonomic systems

Autonomic systems arose in the field of enterprise computing, through an initiative by IBM to reduce the total costs of ownership for enterprise systems (Kephart and Chess, 2003). The ideas have transitioned to communications (Smirnov, 2004) and it is now common to consider both computational and communications aspects of systems from an autonomic perspective. The goals of such autonomic systems are to provide a closed-loop control paradigm (Figure 1) in which the system can use monitoring of its environment and its own behaviour to provide degrees of self-management, self-optimisation and other so-called ‘self-***’ properties. A large number of approaches to autonomic control have been described in the literature, ranging from practically-focused and minimally intrusive techniques like IBM’s Autonomic Computing Toolkit (Jacob et al., 2004) to approaches inspired by biological and physical systems.

Figure 1 The autonomic control loop

Source: Dobson et al. (2006).

Despite this diversity of work, we believe that a number of core challenges remain. These centre on the ability of researchers and practitioners both to address large-scale systems and to offer guarantees to providers and consumers that the techniques being deployed deliver the desired results. At the risk of over-simplification, these challenges give rise to four key drivers for the current work:

Comprehension. First and foremost, autonomic systems must be described and analysed at sufficient depth to give confidence in their overall behaviour. This does not necessarily rule-out systems whose internal mechanisms are some-what opaque, as is sometimes the case with neural networks and genetic algorithms. It does, however, imply that such techniques must be characterised appropriately as part of a larger whole.

Compositionality. Modern systems are too complex to construct monolithically. Complexity must come from the composition of simpler components, whereby the properties of the components, and of the composition operators them-selves, allows us to derive the properties of the composite system.

Openness. The range of techniques being researched suggests very strongly that no single approach will emerge as dominant in the near future. This further suggests that we encourage diversity and focus on interoperability between sub-systems which may adopt different control structures.

Verification. The above issues lead inexorably to the problem of verification: how do we know that a particular system or technique will display the characteristics we desire for it under the expected range of conditions (and possibly beyond)? Given the weakness – and frequent infeasibility – of testing enterprise systems, compositional formal models and their analysis may offer an additional level of confidence.

Therefore, we need a form of mathematics that can be used to state and analyse different autonomic control strategies.

3 Towards a closed-form description of adaptive behaviour

What form might such a mathematical description take? We believe that we can adapt many of the techniques common in other branches of science and engineering, and in the rest of this article we suggest one possible approach.

3.1 A very brief introduction to network calculus

In this section, we present an extremely brief introduction to the core concepts of network calculus: readers are referred to (Le Boudec and Thiran, 2001) for details and proofs.

A network in the calculus is composed of individual network elements, each of which is described in terms of its *input function* (or *input flow*) $R(t)$ defining the sum of bytes received by the element at time t , and its *output function* $R^*(t)$ defining the number of bytes output at time t . Both functions are wide-sense increasing, so $R(t) > R(s)$ iff $t > s$ (and similarly for R^*); it is also clear by definition that $R^*(t) \geq R(t)$ for all t . By convention $R(t) = R^*(t) = 0$ for $t \leq 0$.

For a lossless system, the *backlog* at time t – the number of bits held within the system – is given by $R(t) - R^*(t)$ (which must always be non-negative by definition). The *virtual delay* at time t – the delay experienced by a bit arriving at time t , assuming all bits ahead of it are serviced first – is given by

$$d(t) = \inf\{\tau \geq 0 : R(t) \leq R^*(t + \tau)\}.$$

The input flow at an element is said to conform to an *arrival curve* $\alpha(t)$ if, for all $t \geq 0$ and $s \leq t$,

$$R(t) - R(s) \leq \alpha(t - s).$$

The canonical example of an arrival curve is the *leaky bucket* curve

$$\gamma_{r,b}(t) = rt + b.$$

The interpretation of $\gamma_{r,b}$ is that, it allows traffic to arrive at an average rate of r bytes per second over the long term, but also allows bursts of up to b bytes to arrive instantaneously.

Dually, output flow of an element is said to offer a *service curve* $\beta(t)$ if, for all $t \geq 0$ there exists an $s \leq t$ such that

$$R^*(t) - R(s) \geq \beta(t - s).$$

The canonical example is the *rate latency* curve

$$\beta_{R,T}(t) = R \cdot \max(t - T, 0)$$

which serves traffic at a rate R with a latency of T seconds.

Given these functions, network calculus allows elements to be composed and manipulated using *convolution* and *deconvolution* operators. The convolution of a function f by a function g is another function given by

$$(f \otimes g)(t) = \inf_{0 \leq s \leq t} [f(t-s) + g(s)]$$

and $(f \otimes g)(t) = 0$ if $t < 0$. Dually, the deconvolution of f by g is given by

$$(f \otimes g)(t) = \sup_{s \geq 0} [f(t+s) - g(s)].$$

The final fragment of the theory we need concerns the bounds that can be placed on the behaviour of elements given their arrival and service curves. The *backlog bound* states that, if a flow constrained by an arrival curve α traverses a system offering a service curve β , then the backlog satisfies

$$R(t) - R^*(t) = \sup_{s \geq 0} [\alpha(s) - \beta(s)].$$

The *output flow* of such a system is constrained by an arrival curve $\alpha^* = \alpha \otimes \beta$. Moreover, we can *concatenate* two services together since, if we know that their service curves are β_1 and β_2 , the service curve of the combined system (where the output of the first is used as the input of the second) is given by the convolution $\beta_1 \otimes \beta_2$. In particular, if the service curves are rate latency curves, then

$$\beta_{R_1, T_1} \otimes \beta_{R_2, T_2} = \beta_{\min(R_1, R_2), T_1 + T_2}$$

(i.e. a flow offering the smaller of the two component rates with the sum of the component latencies).

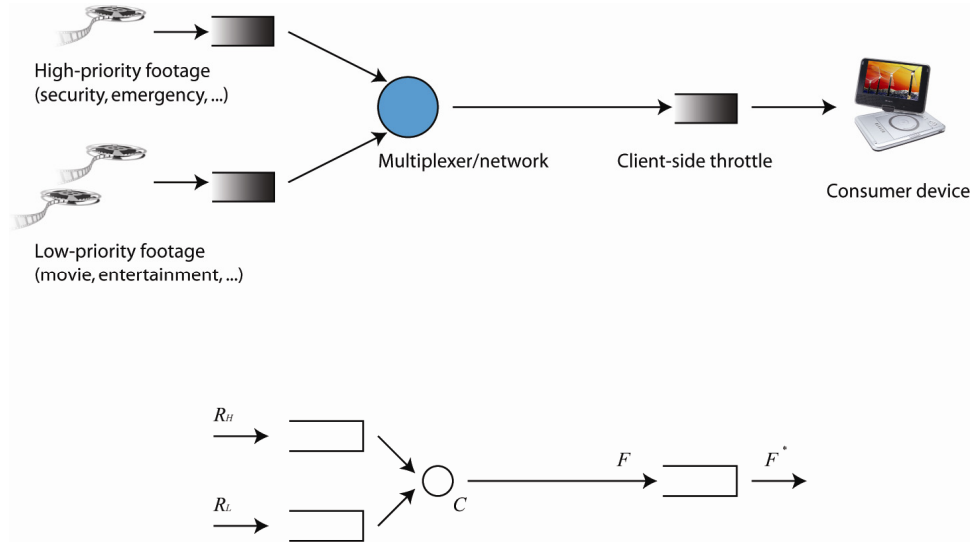
3.2 A systems perspective

The intention of network calculus is to provide performance analysis for systems composed of elements for which the arrival and service curves are known. The system can then be analysed in a number of ways.

For the purposes of illustration, let us consider the simple example network shown in Figure 2 (adapted from (Le Boudec and Thiran, 2001, chapter 1)). This network consists of two elements and four flows:

- 1 a high-priority flow R_H
- 2 a low-priority flow R_L
- 3 a multiplexer between these two flows that delivers traffic at a constant rate C , resulting in a combined flow that transmits high-priority traffic whenever available in preference to the low-priority traffic
- 4 a traffic shaper that delivers a flow to the endpoint.

Figure 2 A simple media-delivery network and its formal model (see online version for colours)



Such a network is a canonical example of a media network in which we constrain the amount of traffic to be delivered to the consumer.

We can analyse this network to determine its performance characteristics. If the low-priority traffic is transmitted in packets of maximum size l_{\max}^L , and if the high-priority traffic arrives at a rate $r < C$, then it can be shown that the multiplexer delivers the high-priority traffic with a service curve of

$$\beta_{C, \frac{l_{\max}^L}{C}}$$

whilst the low-priority traffic receives a service curve

$$\beta_{C-r, \frac{b}{C-r}}$$

If the greedy traffic shaper offers a service curve $\beta_{U, T}$ (where U is the rate of service provided to the endpoint, then during periods when high-priority traffic is available the output flow F^* is given by

$$F^* = R_H \otimes \beta_{\min(C, U), \frac{l_{\max}^L}{C} + T}$$

Although this sounds very abstract, the specification of arrival curves is central to the definition of internet integrated services (IntServ) (Clark and Shenker, 1994) whilst separating flows with different priorities is central to differentiated services (DiffServ; Blake et al., 1998). (Other network architectures provide similar mechanisms, for example, Razzaque, Nixon and Dobson (2006)). Therefore, we can map from the

high-level description to a suite of lower-level implementation techniques whilst remaining confident – to a degree – in the validity of the analysis.

3.3 An adaptive systems perspective

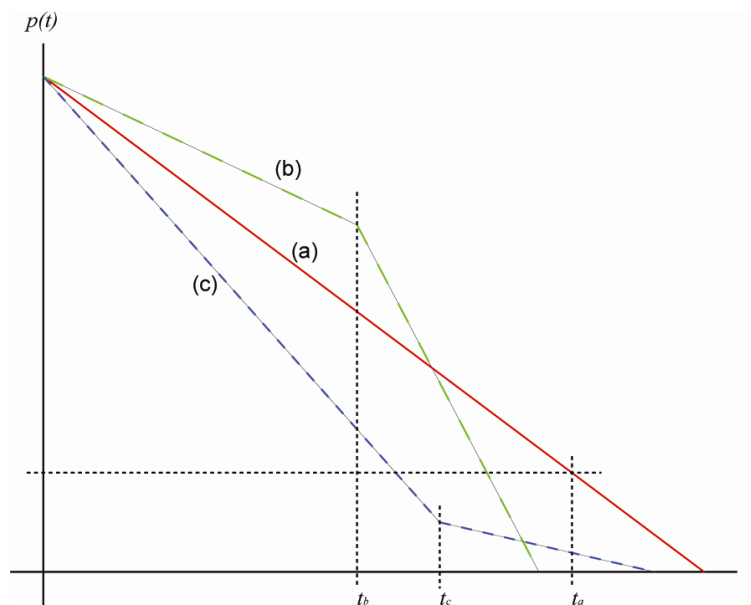
This analysis gives us a steady-state model for the system. What does an *adaptive* system add to this framework?

In an adaptive system, we may want to apply a number of adaptations to adjust the way in which the system behaves. For example, suppose we are in a power-limited environment where a function $p(t)$ models the power remaining at time t . As the power reserves fall we may wish to throttle the bandwidth made available to the endpoint in order to conserve power. In Figure 3, if power consumption follows line (a) we might intervene at time t_a when the power falls below some critical value. This would have the effect of reducing the value of U above.

What control actions could we take? We cannot simply reduce the flow, since this would imply an infinite buffer in the traffic shaper. Instead we could reduce C to remain below U , but this constrains $U \geq r$ to keep up with the arriving high-priority traffic. In this case, we could apply further control to reduce r , or perhaps introduce a lossy component either before or after the multiplexer.

The point here is not the specific actions that one might take, but that the actions, and their consequences, can be modelled relative to the constraints applied to the system by its environment. Put another way, the control actions form a function from p_c to some adjustment of the parameters to the other components of the system. We cannot decide from this much information which particular actions might be desirable – is frame loss acceptable? is it preferable to a reduction in quality? – but we can set out the space of possible actions and use other information to make this decision.

Figure 3 Different function structures cause different responses (see online version for colours)



However, the value of p is not the only variable we might consider. Suppose our goal is to keep at least some video traffic as long as possible. As well as depending on the value of p , we might also depend on the rate at which it is falling, dp/dt : a steeply-falling power reserve should cause a more radical reduction in bandwidth than a gently-falling one (Figure 3, curve (b)). Again, we can select amongst several possible control actions, but the choice we make may be different since dp/dt is now a component in our deliberations and causes us to intervene earlier.

We can go a stage further and place constraints on the way in which we vary control parameters with observations. One might, for example, require that the control actions are, or lead to changes that are, in some sense ‘smooth’ and not involving radical instantaneous change. This has the important characteristic that small errors in observation lead to small errors in control; however, it does not always seem achievable, since some control actions will result in large changes: emergency shutdown is the obvious example. Nonetheless, a goal of ‘smooth design’ can help to make such *critical points* more apparent to encourage care in their specification.

Closing the autonomic loop means that the effect of any control action will be made visible to the control system, and it is this feedback that is the distinctive feature of autonomic systems. Radically, reducing U and the consequent control applied to the network, should cause future observations of power at $p(t + \Delta t)$ to be less than they would have been had no action been taken, as indeed should observations of $(dp/dt)(t + \Delta t)$. In Figure 3, in curve (c) we might intervene at time t_c and expect a reduction in the rate of power consumption. Of course, this is a *hypothesis*, not a fact: it may be that the power rate does not decline, in which case a further control action may be required. This further complicates the decision process, but the core point remains the relationship between observations, actions and their consequences.

Such loops are the core concern of classical control theory, of course, and one might wonder whether anything is contributed by this approach. Control theory has typically not performed well under conditions in which the observations being made, and the control actions to be taken, vary over time. Normally, this is not a problem, but with autonomic systems we might want to be able to introduce new network context, sensors and actuators without extensive recalculation of the control system. This has the dual effect of introducing additional parameter curves into a running system and of turning some of the parameter curves into parameter surfaces, so we might, for example, decide on a control action based on a partial derivative $\partial p/\partial t$. Whilst many of the concerns remain the same – if dp/dt falls as expected we must ensure we do not precipitately undo the control action that caused it to do so – a more compositional and symbolic or semantic approach to control seems desirable. In fact the approach we are proposing seems to be a hybrid of control theory with AI planning: it allows discontinuous actions, dynamic composition and explicit hypothesis testing, whilst seeking to retain closed-form descriptions.

An alternative view of this is to observe that we are establishing a system state space within which we navigate by means of an available portfolio of control actions: establishing a dynamical system, in fact. We conjecture that treating autonomic control in this way will provide stronger, more verifiable guarantees by allowing us to model the ‘envelope’ of system behaviour in a tractable manner – although this remains, for the present, merely a conjecture.

4 Conclusions

The success of autonomic control relies as much upon our ability to predict and verify the behaviour of a network as it does upon our ability to engineer any particular outcome. Without reliable prediction and verification, such techniques will not be accepted in practice.

Network calculus offers an analytic model of network performance, able to model practical systems such as ATM, DiffServ and IntServ as well as systems with lossy channels and data discarding. In this article we have attempted a sketch of an *adaptive* network calculus in which environmental changes can be captured and used to adjust parameters within an underlying network description, which may in turn propagate into control decisions. The approach potentially allows for both design and verification, although substantial work needs to be done to develop these ideas more fully and formally.

An important component of this approach is that, it is largely agnostic with regard to the control actions taken. This leaves developers free to use any of the wide range of techniques being explored in the autonomic communications literature: the only requirements are that approaches can be formally characterised, and that their control decisions can be externalised.

The approach is also agnostic with regard to network architecture. However, we believe that some form of knowledge plane architecture (Clark et al., 2003; Dobson, 2005) provides the most appropriate substrate for this and other approaches to autonomic control, since it provides a clear separation of concerns between inference regarding adaptation and concrete control actions. These architectural ideas deserve further study: Particularly, we are interested in the interaction between formal models of this style and autonomic network management architectures like FOCAL (Strassner, 2007).

Acknowledgements

This work is partially supported by Science Foundation Ireland under grant numbers 05/RFP/CMS0062 “Towards a Semantics of Pervasive Computing” and 03/CE2/I3031 “Lero – the Irish Software Engineering Research Centre.”

References

- Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and Weiss, W. (1998) *An Architecture for Differentiated Services*. RFC 2475.
- Clark, D. and Shenker, S. (1994) *Integrated Services in the Internet Architecture: An Overview*. RFC 1633.
- Clark, D., Partridge, C., Ramming, J.C. and Wroclawski, J. (2003) ‘A knowledge plane for the internet’, Paper presented in the Proceedings of the 2003 *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM’03)* (pp.3–10). New York, NY: ACM Press.
- Coutaz, J. and Rey, G. (2002) ‘Foundations for a theory of contextors’, in C. Kolski and J. Vanderdonckt (Eds), *Computer-aided Design of User Interfaces* (Vol. 3, pp.13–34). Valenciennes, France: Kluwer.

- Dobson, S. (2005) 'Putting meaning into the network: some semantic issues for the design of autonomic communications systems', in M. Smirnov (Ed.), Paper presented in the Proceedings of the *1st IFIP Workshop on Autonomic Communications, volume 3457 of LNCS* (pp.207–216). Berlin, Germany: Springer Verlag.
- Dobson, S., Denazis, S., Fernández, A., Gäti, D., Gelenbe, E., Massacci, F., Nixon, P., Saffre, F., Schmidt, N. and Zambonelli, F. (2006) 'A survey of autonomic communications', *ACM Transactions on Autonomous and Adaptive Systems*, Vol. 1, pp.223–259.
- Jacob, B., LanyonHogg, R., Nadgir, D.K. and Yassin, A.F. (2004) *A Practical Guide the IBM Autonomic Computing Toolkit*. London, UK: IBM RedBooks. IBM.
- Kephart, J. and Chess, D. (2003) 'The vision of autonomic computing', *IEEE Computer*, Vol. 36, pp.41–52.
- Le Boudec, J.Y. and Thiran, P. (2001) 'Network calculus: a theory of deterministic queuing systems for the internet', *Volume 2050 of Lecture Notes in Computer Science*. Berlin, Germany: SpringerVerlag.
- Razzaque, M.A., Nixon, P. and Dobson, S. (2006) 'Demonstrating the feasibility of an autonomic communications-targeted crosslayer architecture', Paper presented in the Proceedings of the *14th International Conference on Advanced Computing and Communications*.
- Smirnov, M. (2004) 'Autonomic communication: research agenda for a new communications paradigm', *Technical Report, Fraunhofer FOKUS*.
- Strassner, J. (2007) 'The role of autonomic networking in cognitive networks', in Q. Mahmoud (Ed.), *Cognitive Networks* (pp.23–52). New York, NY: John Wiley.