Putting meaning into the network: some semantic issues for the design of autonomic communications systems

Simon Dobson

Department of Computer Science, University College, Dublin IE
simon.dobson@ucd.ie

Abstract. Traditional network abstractions follow a layered model in which a sub-system interacts with other network components through very narrow interfaces. We content that this model is weak both in providing clear models of end-to-end properties and allowing adaptation to the more abstract properties of systems. We propose instead a graph-centric, contextual abstract model in which sub-systems can relate to other components at a wide range of semantic levels. We explore the implications such a model would have for network technology, applications and users, and identify some of the major research challenges it poses.

1 Introduction

Networks are complex creations driven by equally complex software stacks, so a critical design goal is to minimise complexity for both network and application developers. The traditional approach uses some form of layering, allowing concerns to be separated behind narrow syntactic and semantic interfaces. This allows individual layers to be modified, extended and replaced without affecting the other layers, and crucially allows networks to evolve without dramatically affecting applications.

Despite its success, however, this trend may be criticised as providing too narrow an interpretation of the information that can usefully be made use of at a particular layer of abstraction in a complex software system. By reducing the information available to a minimum in the interests of simplicity, it is possible that some opportunities for optimisation are lost. In particular, given the rise of pervasive computing systems, we would contend that **contextual information of vital use in adapting the behaviour of a network to its use and environment is being neglected**, and that this acts as a brake on the creation of self-managing, self-adaptive autonomic communication systems.

This paper appears as Simon Dobson. Putting meaning into the network: some semantic issues for the design of autonomic communications systems. In Mikhail Smirnov, editor, *Proceedings of the 1st IFIP Workshop on Autonomic Communications*, LNCS. Springer Verlag. To appear.

In this paper we make the case for modifying (and perhaps eventually reversing) the trend towards layering, and advocate instead *increasing* the amount of information available to a network sub-system about the content it is carrying and the context in which that content will be used. We argue that this view of the network as an equal partner in interactions – rather than as a simple packet-carrier – is an appropriate reaction to the desire for autonomic communication systems that facilitates a range of optimisations currently difficult to accomplish in a scalable fashion. We explore the impact that a flatter model has on networks, applications and users, in order both to determine whether such a model has attractions as a research target and, if so, what research challenges it poses.

Section 2 re-visits some of the strengths and weaknesses the current layered approach to network abstractions. Section 3 proposes a graph-based model of information in which network sub-systems exists as sub-graphs rather than layers, which is then analysed in section 4 to determine its impact on some important network-level, application-level and user-level concerns. Section 5 concludes with some suggestions for further exploration.

2 Layered network abstractions

Ever since the initial design of TCP/IP there has been a desire among network architects (or at least those involved with internet protocols) to focus on the raw performance of the network. Other architectures that stressed differentiated levels of service (such as X.25) have largely been rejected in favour of the simplicity of the internet model with its single class of packets to be routed efficiently. (For an excellent overview of the history of this process see [1, chapter 5].)

While the single-service packet architecture has proved to be fantastically successful, it is now clear that additional qualities beyond bandwidth are required to support the increasingly wide range of applications for which TCP/IP (and other) networks are being deployed. A good example is the provision of *isochronous* media, in which the temporal properties of delivery are at least as important as the data itself. Indeed, many applications using isochronous media would prefer to drop individual data packets rather than compromise the overall temporal characteristics of the data stream.

Current technical solutions to these issues typically use one of two approaches (or both in conjunction). Firstly, new protocols may be developed that allow the additional characteristics of media to be expressed and supported. The Resource Reservation Protocol (RSVP)[2] is a good example of this. Secondly, networks may be dedicated to particular traffic types such as (for example) video, with the network being dimensioned to ensure delivery. Combination networks are increasingly common, good examples being the MBONE[3] and Voice-Over-IP (VOIP) overlay networks.

The use of protocols and overlay networks is in many ways conditioned by the traditional view of networking embodied in the OSI seven-layer model[4] (figure 1) – physical, data, network, transport, session, presentation, application.

(One acronym people sometimes use to remember the layers is "People Design Networks To Send Packets Accurately", which is also a good statement of the traditional view of networks we are criticising!) As with most layered models communications are only allowed between adjacent layers, making it difficult to support "end to end" statements[5]. Although the OSI model's significance is as a conceptual tool rather than a guide to implementing a real network, its notion of layers has become very prevalent.

A key notion in layered architectures is that each layer only "understands" as much about the content it is transporting as its interface allows to be expressed. This is typically very little information expressed at a very low level of abstraction. In this paper, by contrast, we are making the case for leveraging meaning from as wide a range of sources as possible. A key point is that such knowledge is *not* strictly hierarchical: knowledge may potentially have an impact across the spectrum of concerns, and does not follow a strict layered structure.

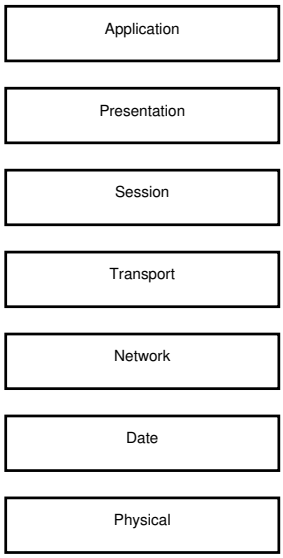

Fig. 1. The OSI reference model

One might argue that layering allows translation of concerns from one domain to another, so layer boundaries provide points at which (for example) session information is translated into transport requests. This means that layer n provides a service abstraction to which higher-level concerns (from layers $n+1$, $n+2$ *et cetera*) are mapped. This can result in semantic "squeezing", in the sense that two *different* high-level concerns that happen to map to the *same* layer-n concerns will then be indistinguishable to layers $n-1$ and so on. This limits the ability of lower layers to react to higher-level concerns.

Conversely, suppose a concern at layer n could be used to inform the adaptation of layer $n-2$. We may understand this adaptation by saying that layer $n-2$ responds to certain properties at layer n. However, the concern is mediated by layer $n-1$, and so must be presented to layer $n-2$ in layer $n-1$'s terms. Moreover, for the system to behave predictably this translation must be exact: layer $n-2$ must adapt *as if* it responded to layer n, although it in fact responds to layer $n-1$.

In a traditional network, layering is used to *simplify* the way in which concerns interact. In a more complex network, however, layering may *complicate* interactions by introducing translations that are not completely meaning-preserving. One might speculate about a data layer that can account for application preferences in terms of jitter and frame dropping – but which is unable to do so because the intermediate layers do not present these concerns in a coherent form suitable for decision-making.

3 Meaning in the network

Our concern, then, is that a layered architecture may not provide sufficiently rich access to available information. In traditional networking this is not a problem, and information can be provided implicitly through the development of new protocols *et cetera*. The question we now turn to is whether this remains true as we move towards more self-managing network architectures operating in richer information spaces.

3.1 The limits of layered access to information

TCP/IP is a general-purpose protocol, specified independently of any particular content. While the same is also true for RSVP, the latter's main purpose is to introduce additional "metadata" into the network – in this case the qualities of service required by isochronous media. At a transport level this takes the form of new message types, new processing rules and so forth; at an application level these manifest themselves in the improved delivery of time-dependent content. Clearly there is a close semantic link between these different levels: one may however question both whether hard-wiring these decisions in protocols is the more appropriate way to proceed, and whether the link can be maintained as the system evolves given the information that is explicitly available for decision-making.

It is important to recognise the distinction between *content* and *use* when discussing networks. The former represents the inherent information being represented and transferred; the latter represents a user task or collection of requirements involving that content. A network protocol is typically more concerned with use than with content: one may transfer the a song with RSVP when it is being played but with HTTP or FTP when it is being archived or mirrored. In the second case, even isochronous media do not place isochrony requirements on the network, and it is desirable to avoid the additional overheads involved. It is not clear that the user or application level should make (or indeed will be able to make) this choice accurately.

In fact there is generally more layering taking place than is apparent in the OSI (or any other) model. Consider, for example, the common practice of securing a data stream using SSL. At one level it is possible to combine SSL with RSVP, by running the SSL packets over the resource-managed connection; at another level, however, this changes the behaviour of the data stream by adding a new layer of processing for encryption and decryption of which RSVP may not be aware and which it may not account for in setting its connection parameters.

Although layering allows individual sub-systems to be changed with minimal disruption, the operative word is *minimal* disruption – which is not the same as *no* disruption. Adapting a system by changing a layer (or component within a layer) may have "knock-on" effects elsewhere in the system. A good example is switching from one encryption system to another, stronger version as content

changes, which impacts the transport parameters. Layering can therefore give a false sense of stability in an adaptive system.

More generally, there is a certain arbitrariness in discussions about using networks, protocols and router-based routing decisions in managing networked data streams. It is easy to construct scenarios in which the optimal transport policy – frame size, latency, bandwidth, sensitivity to dropped data, isochrony requirements, security and confidentiality, *et cetera* – depends critically on a wide variety of user-, application- and task-level details. This sensitivity to *context* is a familiar feature of pervasive computing systems: it may be a critical component of next-generation networks too.

3.2 Meaning and context

Context, in its most general form, may be taken as **the environment in which a system operates understood symbolically**. The concept is used extensively in pervasive computing, where the goal is to make computing devices more adaptive to their users' tasks and requirements as these change over the course in interaction.

A key observation about context is that it is non-hierarchical. One will often encounter links between facts at different conceptual levels that – if captured – illuminate facets of the environment that can prove extremely useful in reasoning and processing. More importantly, context-aware computing makes clear the subtle relationship that exists between users and information. It is not the case that a single piece of information will always be used in the same way; nor is it the case that information has constant relevance to users, or will always be presented at full fidelity, or must always be accessible, and so forth. Pervasive computing is fundamentally concerned with delivering the correct service to the correct user at the correct place and time, and in the correct format for the environment[6]: it is the use of richly interconnected information, and not necessarily constant connectivity or quality of service, that constitutes its chief strength.

Viewing a network as a component of a context-aware system leads immediately to questions of how the network can adapt to the wider context, and how it can be used to inform other components about that context. One may thus see the network – as with most other components of a pervasive computing system – as both a producer and consumer of context.

This leads us to a position contrary to the layered approach of section 2. Instead of viewing a network system as constructed of layers, what happens if we view the system as a *network itself* in which no structure is unduly privileged? This is show schematically in figure 2. The important difference between this view and the layered view is that information can flow directly between different parts of the system – the sub-systems are *sub-graphs* rather than layers.

The significance of this change is two-fold. Firstly, it "flattens" the space of information which a network can access. It reflects a more holistic, and to some extent more trusting, view of information, allowing components to access any information they can usefully process rather than forcing them to use narrow

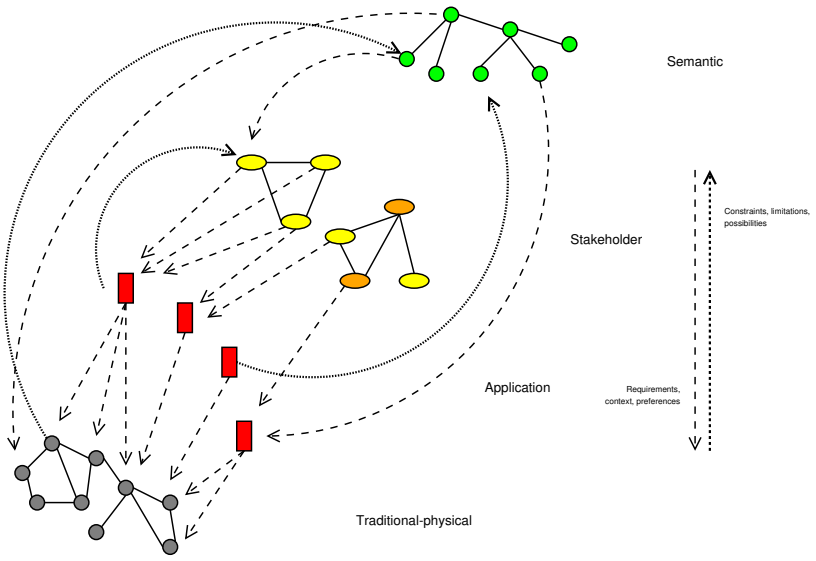

Fig. 2. Network as embedding

syntactic and semantic interfaces. Secondly, it provides a richly-linked and extensible framework within which to represent *all* the information pertaining to a system's context and configuration in a semantically well-founded manner (for example using RDF[7] to represent assertions about facts). One may then deploy a number of advanced software techniques over this information, the output of which affect the control plane of the underlying network.

As an example, consider the case of an application requirement that a particular data stream be secure against casual observation. The standard solution may be to armour the connection using SSL or similar. However, if a particular network is know to be secure – for example because it is a "hardened" land-line, or a wireless channel that is encrypted with sufficient security automatically at the air interface – then additional armour may be considered unnecessary (and may actually be harmful for security). In a layered model the decision to deploy SSL would almost certainly be made in the top levers, which might not have access to network-layer information about the intrinsic security of a channel; the contextual model potentially allows a decision lower down the stack, accessing the semantic requirements from the higher levels.

The recent interest in the network's "knowledge plane" is a step in the direction we are advocating. The knowledge plane is intended as a single locus for representing and reasoning about higher levels of knowledge in the network, running knowledge-based applications. The important point, however, is not so much the architectural detail but rather to ensure that the management system

has access to information at *all* semantic levels within the network – and this might be argued to negate the advantages of plane- or layer-based architectures.

4 Analysis

Conceptual models are important only insofar as they provide insight into design, analysis, teaching or some other mode of understanding. We will therefore explore the model of section 3 above with a view to seeing how a more contextual view of information flow impacts important concerns in autonomic communications, and use these observations to derive technical research questions needing further study.

4.1 Network concerns

When considering the traditional concerns of networking, the context in which the network is being used may seem too abstract. However, a network is primarily a system for facilitating *human* communication and activities, so the context in which these activities occur is actually of fundamental importance.

What makes the contextual viewpoint different is that it encourages the explicit articulation of information. In the layered world one might deploy RSVP in order to accomplish smooth video display; in the contextual world one might state that (for example) an application is displaying video and requires certain transport characteristics in order to achieve a viable user experience. These are the facts that inform the decision to use RSVP: however, in certain circumstances an alternative decision might equally satisfy the requirements with less overhead, or might be able to meet other requirements that have been implicitly discarded. This looser specification is then an opportunity for adaptation.

While simple data and isochronous media represent two ends of a spectrum of service requirements, there is an important middle ground in which data is transferred using standard protocols and networks but is handled subtly differently depending on its content. Content-based routing (for example [8, 9]) allows routing algorithms to take account of higher-level issues concerning the content and use of the data being routed. A context-based approach can generalise these solutions so that (for example) routing priorities take account of content, user task, information relevance and any other available factors.

If we consider a network as a producer of context, we may integrate its auditing and management functions directly into the overall context model. This is an immediate consequence of the tendency to articulate all available information in a consistent format, and allows management functions to make use of both low- and high-level information on a network's performance. This increases the amount of information available for decision-making, which hopefully leads to improved accuracy.

4.2 Application concerns

Applications collect together sets of requirements, some of which will impact the network. While traditional packaged applications may not be the best approach to pervasive computing[10], user-visible functionality can still be used to *prima facie* inform the adaptation of the network, and *vice versa*.

A good example of this co-dependence is where an application wants to adapt its presentation according to the bandwidth available. In a layered system this is often problematic, as the application does not have access to the lower layers in which bandwidth concerns are handled. Although some protocols expose the necessary information, many do not. A context system might have the transport sub-system write its current and expected bandwidth availability into the model, where changes can trigger any application expressing an interest in these facts. Conversely a transport system might read applications' expected future transfer requirements from the model and use them to pre-allocate resources.

4.3 Stakeholder concerns

A major issue in any pervasive system is user acceptance. Any system that is responsive to user context must obviously model that context, and there are few guarantees that the model will be used only for purposes users will accept. The price of adaptation seems to be increased surveillance and more sophisticated models of behaviour.

In many cases it is clear that users will forgo a degree of privacy in exchange for monetary benefit or increased ease of use. It is however equally clear that this trade-off is both difficult to make and problematic to enforce.

There is no obvious answer to these concerns at present. One may observe, however, that systems that articulate changes in parameters that control decisions may also audit how those decisions are made. Extending this paradigm to broader systems may allow a degree of traceability in decision-making that can be used to detect *post facto* some violations of acceptable use.

4.4 Some questions and directions

Automated configuration based on requirements sounds very attractive. It does however require a very subtle reasoning process in order to decide on a particular choice of (for example) network protocol from a given constellation of facts, and one might question whether automated reasoning will be adequate to the task – especially given the performance requirements of many network systems. There are also questions about the extensibility of such solutions, given that rule-based systems are often fragile with respect to changes in their rules.

A related question concerns the extensibility of context models. While one may conjecture about networks that use semantic information from applications, this requires as a basis that the network component understands the information model being used by the applications. This seems to imply a considerable degree

of coupling between components, which is problematic for both engineering and commercial reasons.

A more semantic issue is the stability of systems reacting to rich models. One can easily conceive of a system that is finely balanced between two possible choices and oscillates between them, incurring costs at each oscillation. This is especially hard to find when the changes are effected by different components reacting in conflicting ways to separate parts of the model.

We highlighted the ability to integrate low- and high-level information into management tools. The benefit of this approach is that is simplifies the expression of end-to-end properties, in that there is a "trace" from high-level requirements to management decisions. Creating this pathway implies that we can provide suitable decision procedures and handle conflicts between requirements and tasks.

The use of pervasive computing as a surveillance tool is a widespread concern, and one which has perhaps not received sufficient attention from the technical communities so far. It goes significantly beyond the normal issues of cryptography, into the realms of traffic analysis used by the military to gain information from who is talking to whom (regardless of whether the actual communications can be read). Users (and corporations) will not use systems that may leak information to marketeers (or competitors, or governments). It is an open question where autonomic communications stands in the space of adaptivity *versus* intrusiveness.

5 Conclusion

We have investigated how the traditional notion of layering in a network may not be optimal for accessing the richer spaces of information that may be available to next-generation networks, and especially in pervasive computing systems. We have argued for a flatter, more extensible context model to allow sub-systems to access information from any appropriate semantic level, and have explored some of the issues that such a model raises.

Our intention in this paper has purely been to explore the attractions and feasibility of applying a more holistic and contextual approach to information to autonomic communications and self-adaptive networks. Our conclusions would be (firstly) that such an approach does have significant attractions, and (secondly) that there are significant challenges in terms of user control, information leakage and the extensibility of decision procedures. Nevertheless we conclude overall that a semantically well-founded approach to autonomic communications based on techniques from contextual modelling is worthy of further attention in the future.

The challenges that we have identified relate primarily to issues of knowledge representation and automated reasoning – what might broadly be called artificial intelligence. Not all the capabilities we have identified exist currently in a form suitable for use in the way we have suggested. However, by developing and applying these techniques to autonomic communication we hope to improve

their capabilities for adaptation and management while retaining a degree of confidence in the correctness and compositionality of these capabilities. Both are vital if communications are to become truly autonomic.

Acknowledgements

The work in this paper expands on discussions that took place at a European Union workshop on Autonomic Communications held in Brussels in March 2004, for which the author acted as one of the rapporteurs. Many of the participants raised points that contributed greatly to the ideas presented here, although any misunderstanding of these points is of course due to the author alone.

References

1. Open systems interconnection – basic reference model: the basic model. Technical Report ISO/IEC 7498-1:1994, ISO, 1994.
2. Janet Abate. *Inventing the Internet*. MIT Press, 1999.
3. R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – version 1 functional specification. RFC 2205, 1997.
4. Markus Debusmann and Kurt Geihs. Towards dependable web service. In *Proceedings of the 10th IEEE/IFIP International Pacific Rim Symposium on Dependable Computing*, pages 5–14. IEEE Press, March 2004.
5. Simon Dobson. Putting meaning into the network: some semantic issues for the design of autonomic communications systems. In Mikhail Smirnov, editor, *Proceedings of the 1st IFIP Workshop on Autonomic Communications*, LNCS. Springer Verlag. To appear.
6. Simon Dobson. Applications considered harmful for ambient systems. In *Proceedings of the International Symposium on Information and Communications Technologies*, pages 171–176. ACM Press, 2003.
7. Ora Lassila and Ralph Swick. Resource Description Framework model and syntax specification. Technical report, World Wide Web Consortium, 1999.
8. Jerome Saltzer, David Reed, and David Clark. End-to-end arguments in system design. *ACM Transactions on Computer Systems*, (4):288–288, 1984.
9. Kevin Savetz, Neil Randall, and Yves Lepage. MBONE: Multicasting tomorrow's internet, 1995.
10. Mark Weiser. The computer for the 21st century. *Scientific American*, September 1991.
11. Chu-Sing Yang and Mon-Yen Luo. Efficient support for content-based routing in web server clusters. In *2nd USENIX Symposium on Internet Technologies & Systems*, 1999.