User Cooperation and Search in Intelligent Networks*

Erol Gelenbe

Dennis Gabor Chair,
Department of Electrical and Electronic Engineering,
Imperial College London SW7 2BT
e.gelenbe@imperial.ac.uk

Abstract. We present a vision of an Intelligent Network in which users dynamically indicate their requests for services, and formulate needs in terms of Quality of Service (QoS), duration, and pricing. Users can also monitor on-line the extent to which their requests are being satisfied. In turn, services will dynamically try to satisfy the users as best as they can, and inform the user of the level at which the requests are being satisfied, and at what cost. The network will provide guidelines and constraints to users and services, to avoid that they impede each others' progress. This intelligent and sensible dialogue between users, services and the network can proceed constantly based on mutual observation, network and user self-observation, and on-line adaptive and distributed feedback control which proceeds at the same speed as changes in traffic flows and the events occurring in the network. We survey some of the technical problems that arise in such networks, illustrate the networked system we propose via an experimental test-bed based on the Cognitive Packet Network (CPN), and discuss the key issue of search for users and services.

Keywords: Network Intelligence, Autonomic Networks, Users and Services, User Goals and Quality of Service, Cognitive Packet Networks.

1 Introduction

Sheer *technological capabilities and intelligence*, on their own, are of limited value if they do not lead to enhanced and cost-effective capabilities that are of value to human – or even beyond humans – to living users. Fixed and then mobile telephony and the Internet have been enablers for major new developments that improve human existence. However advances in telecommunications have also had some undesirable and unexpected outcomes during the past century. A case in point is television broadcasting. It was initially thought that television broadcasting would become a wonderful medium for education. Unfortunately in many instances it has lowered public expectations with regard to the quality of entertainment by limiting the range of programs and content that

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-3-540-32993-0_29

* Research supported by UK EPSRC under Grant GR/S52360/01 and by the EU FP6 Marie Curie Programme under project MIRG-CT-2004-506602.

I. Stavrakakis and M. Smirnov (Eds.): WAC 2005, LNCS 3854, pp. 41–56, 2006.

are available. Interestingly enough, with few exceptions this effect has appeared both in socio-economic environments where television has been driven by purely commercial considerations, and in other environments where television broadasting has been driven by monopolistic considerations. This is a good example of a tremendous success in technology which has not been exploited in the most broadly intelligent manner. Since the massively "one-to-very-many" broadcast nature of television has not given communities of users the possibility to dispose of high quality content, one can hope that other models of communications, such as the peer-to-peer concept, can offer a greater degree of user choice and also offer some very high quality content for an enhanced cultural and humanistic environment. Thus we envision Intelligent Networks to which users can ubiquitously and harmoniously connect to offer or receive services. We imagine an unlimited peer-to-peer world in which services, including television broadcasts, voice or video telephony, messaging, libraries and documentation, live theater and entertainment, and services which are based on content, data and information, are available at an affordable cost. In these networks the technical principles that support both the "users" and the "services" will be very similar if they are framed within an autonomic self-managing and self-regulating system. This network will be accessible via open but secure interfaces that are compatible with a wide set of communication standards, including the IP protocol.

We imagine an Intelligent Network (IN) in which users and services play a symmetric role: users of some services can be services of other users, and services can be users of some other services. Users and services can express their requests dynamically to the network in terms of the services that they seek, together with Quality-of-Service (QoS) criteria that they need, their estimate of the quantity or duration of the requested service and the price that they are willing to pay. The users could also have the capability to monitor on-line to what extent their requests are being satisfied. In turn the services and the network would dynamically try to satisfy the user as best as they could, and inform the user of the level at which their requests are being satisfied, and at what cost. The network would also provide guidelines to users to avoid that the latter impede each others' progress. Similarly, network entities and services would also conduct a dialogue, so that they can collectively and autonomously provide a stable, evolving and cost effective network infrastructure. We will sometimes find it useful to distinguish between users and services, merely to indicate the relationship that exists between a specific user requesting a specific service. But we wish to stress that at a certain level of abstraction, these two entities are indeed equivalent. The IN should offer the facilities for a sensible dialogue between all users, including services, and it will adapt to users' needs based on mutual observation, network and user self-observation, and on-line distributed feedback control which acts in response to the events that are being controlled.

1.1 Research Issues

The vision we have described raises many interesting research questions, some of which are discussed below. An obvious research question relates to the network

architecture that can physically support the vision we have presented, in particular with respect to system software. We would expect that the IN would be some form of self-managing and programmable overlay network [14, 15] which is discussed in the next Section. The underlying hardware architecture would have to rely on network components, wired or wireless technologies and line speeds that are available at any given point in time, while routers should be programmable and they should not be limited to some pre-defined protocol such as IP. More research is also needed to better identify services and their characteristics, and the technologies which are necessary to support applications in a wide range of heterogeneous networks, with possible feeders coming in from the wide-spread networked sensors of the future. Research is also needed to design network modules whose role will be to recognise and match user needs to the networking context.

Although much excellent work has been done about QoS provisioning, QoS based routing mechanisms, and service differentiation [7, 8, 12], there is still much more that needs to be done in defining broad QoS metrics that are relevant to the end user, and seeing how these translate into mechanisms and policies that exploit the available variability including traffic engineering, and routing and searching methods [10, 11], so that both the users' needs and the networks' objectives can best be met. Understanding the interplay between cooperative or conflicting interests among different users and networks, including issues such as resource utilisation, provisioning, pricing and QoS [8, 9] has received considerable attention. A recent paper provides valuable insight and ideas on some of these issues [18]. Our impression is that a systematic approach to realistic modelling of the dynamic interplay between these different issues can still be of great value to a better understanding of network control. Furthermore we believe that the game theoretic ideas that have been developed should also lead to more experimental research, testing and evaluation in realistic environments, or in large-scale network test-beds.

Although there have been many studies that characterise network traffic, and considerable work has akready been done on various aspects of network observation such as network tomography, we suggest that further research is needed on approaches to network measurement whose primary objective is the real-time control of network performance, of traffic engineering, or of user QoS [17]. Resource provisioning in networks can of course be handled in an *a priori* manner. However when resources are tight, or when the networked environment is imperfectly known, or when the users are accessing very diverse services and resources, or when users, services and network resources are mobile, then the network state can only be observed by real-time measurements of the parameters that are of direct interest. Thus research which combines QoS considerations, with network control and measurement, appears to be of interest [16, 19].

Another important area of research is the design of algorithms that can help users or the network itself to discover and find "things" such as nodes, services, resources, etc. in very large, or even infinite, networked systems. Search problems have long been examined in artificial intelligence and robotics, as well as in the

context of combinatorial mathematics [6, 13]. It appears to us that this topic is of increasing importance, because users will operate increasingly in ad hoc networks, sensor networks, or more broadly in large and unknown networks where they will have to discover the best connections, services and modes of operation. For instance, a user may have to connect his/her terminal device automatically to some network in a city that he/she has never visited before and about which he/she only has very sketchy information.

1.2 An Architecture for the Intelligent Network

A sketch of the Intelligent Network (IN) is shown in Figure 1. The IN will based on a standard communication interface derived from the Internet Protocol (IP). Users U (shown with small purple rectangles as U1, U2, etc.) are generally mobile and can be recognised via their ID and password. Users have a credit with the network and with certain network services, as represented by a credit allocation or via a "pay as you go" scheme (e.g. with a credit card), or they can access certain free services or services that may be paid for by the service provider (e.g. advertisements). Users can have a user terminal which may be as simple as a Personal Digital Assistant or mobile phone, or as complex as intelligent network routers (INRs) shown as blue octagons in Figure 1. Users are connected to the IN via INRs or directly to a network cloud (shown as clouds of different colours). Services S (shown as S1, S2, ..) are very similar to users in that they have an ID

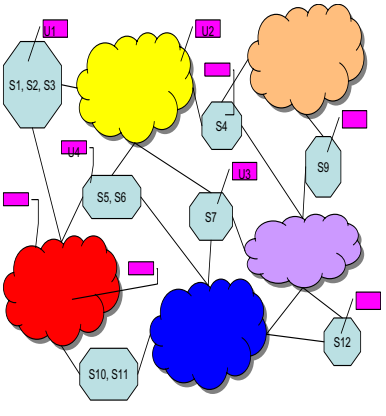

Fig. 1. Architecture of the Intelligent Network

and they may have a credit allocation; they can also receive credit when their services are used by users, just as users may be reimbursed by services or by other users. Services can also be mobile. However:

- Users will in general be light-weight (a mobile phone, a PDA, or just a user ID and password),
- While services will be much more complex and may often be resident on one or more INRs, or they may own one or more INRs for their needs.

When some other user or service asks something of a user, the chances are that there will be an automatic answer saying "sorry no; I am just a simple user". On the other hand, services will often be equipped with authentication schemes to recognise the party who is making a request, billing schemes that allow for payment to be collected, schemes allowing a service to be used simultaneously by many users, and so on, depending on the complexity of the service being considered.

INRs are machines or clusters which can be identified by the community of users and services. Network clouds on the other hand are collections of routers internally interconnected by wire or wireless and which are only identified as far as the users and services are concerned via the ports of INRs which are linked to a cloud; in other words, users and services do not actually know who and what is inside a network cloud. However INRs, and hence users and services, can observe the QoS related to traversing a network cloud; this may include billing of the transport service by the cloud. Also, clouds may refuse traffic, or control and shape the traffic that wishes to access them, depending on the clouds own perception of the traffic.

The IN architecture we have described can be viewed as an overlay network composed of INRs with advanced search, QoS (including pricing and billing), that links different communities of users and services. The networked environment of the future will include numerous INs, and there may be specific INs whose role is to find the best IN for a given user. Some of the se INs may be quite small (e.g. a network for a single extended family), while others would be very large (e.g. a network that provides sources of multimedia entertainment, or educational content). In the three following sub-sections we will discuss three important enabling capabilities of the system: finding services and users, routing through the network, and self-observation and network monitoring to obtain the best QoS and performance.

2 Finding Services and Users

We expect that the IN will have different free or paying directory services that will be used to locate users and services. When appropriate, these directories may provide a "street address and telephone number" for a service that is being sought out; however, since in many cases the services will have a major virtual component, they will especially provide a way to access them virtually, either via an IP address, or more probably via one or more INR addresses or one or more network paths.

The directory services will offer "how to get there" information similar to a street map service, providing a network path in terms of a series of INRs or of network clouds, from the point where the request is made, to the INR where the service can be found. Directory services may have a billing option which is activated by services to reward the directory for being up-to-date, or services or users can subscribe to them, or they may be paid for via advertisement information, and so on. These directories will be updated pro-actively by the services

or by the directories themselves, or on demand when the need occurs. Updates would also occur when INR or network cloud landmarks change. Directories can be "smart" in the sense that they offer information about faster or less congested paths to services that are requested, or paths to less expensive services, or paths that are better in some broader sense. An approach for achieving this based on the Cognitive Packet Network (CPN) [10, 20] protocol is described in Section 3.

As a way to understand how a connection can be established between some user U and a service S in the IN, let us go through some of the steps that may be involved, using smart packets (SPs), based on ideas from [17, 20] and some extensions of these ideas.

- U first searches for a directory; assuming he finds one, U formulates his request in the form of (SX, QY, PZ) meaning that he wants a service SX at QoS value QY for a price of PZ. The directory either is unable to answer the request, or it provides one or more paths $\pi(U, SX, QY, PZ)$ which best approximate this request for several possible locations of the service.
- Assuming that the directory does provide the information, U sends out (typically via the INR) a sequence of smart packets SPs which have the desired QoS information, with several following each of the possible designated paths. The first SP for each of the paths will follow it to destination, with the purpose of verifying that the information provided by the directory is correct. Subsequent SPs on each route will be used to search for paths: they will invoke an optimisation algorithm at all or some of the INRs they traverse so as to seek out the best path with respect to the user's QoS and pricing requirements.
- INRs collect measurements and store them in mail boxes (MB). These can concern both short term measurements which proceed at a fast pace comparable to the traffic rates, and long term historical data. INRs will measure packet loss rates on outgoing links and on complete paths, delays to various destinations, possibly security levels along paths (when security is part of a QoS requirement), available power levels at certain mobile nodes, etc.. This constant monitoring can be carried out using the SPs and other user related traffic, or using specific sensing packets generated by the INRs.
- The network monitoring function can also be structured as a special set of users and services whose role is to monitor the network and provide advice to the users and to the directories.
- Each SP also collects measurements from the INRs it visits which are relevant to its users QoS and cost needs, about the path from the INRs which it visits.
- When a SP reaches a service SX, an acknowledgement ACK packet is sent back along the reverse path back to U; the ACK carries the relevant QoS information, as well as path information which was measured by the vSP and by the ACK, back to the INRs and to the user U. The ACK may thus be carrying back a new path which was unknown to the directory.
- For a variety of reasons, both SPs and ACKs may get lost. SPs or ACKs which travel through the network over a number of hops (ERs or total number including routers within the clouds) exceeding a predetrmined fixed

number, will be destroyed by the routers to avoid congesting the IN with "lost" packets.
– Note that the SPs and ACKs may be emitted by the directory itself, rather than by U. This would be an additional service offered by certain directories. One could also imagine that both users and directories have this capability so as to verify that the request is being satisfied.

Some of these features are illustrated in the CPN (Cognitive Packet Network) system [17, 19, 20] test-bed that we have implemented at Imperial College.

2.1 Individual Versus Collective QoS Goals

The usual question that any normally constituted telecommunications engineer will ask with respect to the vision that we have sketched is what will happen when individual goals of users and services conflict with the collective goals of the system. We are allowing for users to set up the best paths they can find, from a selfish perspective, with services, and for services to actually do the same, in parallel with the behaviour of users. This has the potential for:

– Overloading the infrastructure, because services have an interest in maximising their positive response to user's needs, and they may even overdo it in terms of solliciting users; because of the possibility of billing, portions of the infrastructure itself may have an interest in getting overloaded.
– Creating traffic congestion and oscillations between hot spots, as users and services switch constantly to a seemingly better way to channel their traffic.
– Opening the door to malicious traffic whose sole purpose may be to deny service to legitimate users through the focused creation of overload in the services or the infrastructure (e.g. denial of service attacks).

The first of these points, which does not relate to malicious behaviour, can be handled through overall self regulation of the INRs, the users and services:

– When a new part of the infrastructure joins the IN, for instance a INR, it will be allocated an identity within the IN. We could have a virtual regulating agency (VRA) which sets up a dialogue with the INR to provide it with its identity, and which ascertains its type and nature from its technical characteristics. The VRA then enables the INRs operating systm with a set of parameters which in effect limit the number of resident processes and the amount of packet traffic that this particular INR can accept.
– Services and users which join the IN, also need to be identified by the VRA. Just as a shop rents a certain space in a building and on a particular street, the VRA can provide the service with a "footprint", depending on the rent it is willing to pay, and on the VRA's knowledge of currently available resources. This footprint can then determine the fraction and amount of processing power and bandwidth that it is allowed inside the IN and at any given INR.
– Note that the overall quality and seriousness of the VRA will make a particular IN more or less desirable to users and services.

The second point is related to dynamic behaviour. Each INR, in its role as a service support centre enabled by the VRA, will run the dynamic flow and workload control algorithms for each service and user that it hosts. However it will also run a monitoring algorithm which has IN-wide implications.

- For some user U assume that $RU(S)$ is the rank ordered set of best instantaneous choices for some decision (e.g. what is the best way to go to service S with minimum delay).
- At the same time, let $RN(U, S)$ be the rank ordered set of best instantaneous choices for the network (e.g. what is the best way to go to where service S is "sitting" so that overall traffic in the IN is balanced).
- The decision taken by the INR will be some weighted combination of these two rank orders. The weights can depend on the priority of the user, of the price it is willing to pay, and so on.
- Choices which are impossible or unacceptable to either of the two criteria (user or network) will simply be excluded. If there are no mutually possible choices, then the request will be rejected. When there are ties between choices, any one of the tied choices can be selected at random.

As an example, suppose that the ranking indicating the user's preference, in desecending order, among six possible choices is $\{1, 2, 3, 4, 5, 6\}$, while the network's preference ranking could be $\{5, 4, 2, 3, 1, 6\}$. If we use rank order as the decision criterion and weigh the INR and the user equally, then the decision will be to choose 2 whose total rank order is 5. If the network's role is viewed as being twice as important, we can divide the network's rank for some choice by 2 and add the resulting number to the rank that the user has assigned to that choice, which results in a tie between the three top choices $\{1, 2, 5\}$. If the network's role is three times more important, then we get a tie for the top choice between $\{1, 5\}$, and so on.

2.2 The Eternal Problem of Scalability

It is often said that the main impediment to the broad use of QoS mechanisms in the Internet is the issue of scalability. Indeed, if each Internet router were enabled to deal with the QoS needs of each connection, it would have to identify and track the packets of each individual connection that is transiting through it. The routing mechanism we propose for all requests through the IN is based on dynamic source routing[1]. In other words, the burden of determining the path to be used rests with the INR that hosts the service or user. In our proposed scheme, routers have two roles:

- The INR generates SPs for its own use that monitor the IN as a whole, and the user or service process resident at a INR generates the SPs and ACKs which are related to its connections to monitor their individual traffic.

[1] Note that MPLS is a form of distributed virtual source routing where label switching at each node maps virtual addresses into physical link addresses.

- As a result of the information that it receives from SPs and ACKs, of the information similarly received by users and services that are resident at the INR, and of the compromise between global (IN) and local (user and service) considerations, the INR generates source routes for its resident users and services.
- Each INR also provides QoS information to SPs and ACKs that are not locally generated but which are transiting through it, such as "what is the loss rate on this line", or "what time is it here now", or "what is the local level of security".

Thus we propose to avoid the scalability issue by making each INR responsible only for local users and services, much as a local telephone exchange handles its local users. Source routing removes the burden of routing decisions from all but the local INR, reducing overhead, and removing the need of "per flow" information handling except at INRs where the flows are resident. However, it comes at the price of being less rapidly responsive to changes that may occur in the network. This last point can be compensated by constant monitoring of the flow that is undertaken with the help of SPs and ACKs. Our scheme also requires that INRs be aware of the overall IN topology in terms of other INRs (but there is no need to know what is inside the "clouds"), although this can be mitigated if one accepts the possibility of staged source routing, i.e. with the source taking decisions up to a given intermediate INR, which then takes decisions as far as some other INR, and so on. Note that this scheme is more general than the one we will describe in the next section which consists of an experimental system that discovers destination nodes, and paths to destination nodes which optimise user specified QoS metrics.

3 Searching and Routing with CPN

Distance Vector and Link-State algorithms are the usual methods for finding the shortest paths to IP addresses in the Internet's Routing Information Protocol (RIP) [7], where a table in each router stores information for each destination in a sub-network with a preferred outgoing link from the node, and an estimate of the time and hop count to the destination. These metrics are updated at regular intervals, or when the network topology changes, via router update messages. This allows each router to update its database with the fastest route being communicated from neighboring routers. However, many factors including non-negligible delay, infrequent link state update due to overhead concerns, and the link state update policy can impact global network state information.

CPN [20] is a distributed algorithm that provides QoS driven routing based on searching for the best path to a given destination. CPN searches for the destination *and* searches for the best path leading to it. It uses *Smart or Cognitive Packets (SP)* that discover routes using a reinforcement learning (RL) algorithm based on a QoS "goal" such as packet delay, loss, hop count, jitter, etc.. The "goal" may be defined by the user, or by the network itself. SPs find routes and collect measurements, but do not carry payload. The RL algorithm uses the

observed outcome of a previous decision to "reward" or "punish" the mechanism that lead to the previous choice, so that its future decisions are more likely to meet the QoS goal. When a SP arrives to its destination, an ACK packet is generated; the ACK stores the "reverse route" and the measurement data collected by the SP. It will travel along the "reverse route" which is computed by taking the corresponding SP's route, examining it from right (destination) to left (source), and removing any sequences of nodes which begin and end in the same node. For instance, the path $< a, b, c, d, a, f, g, h, c, l, m >$ will result in the reverse route $< m, l, c, b, a >$. Note that the reverse route is not necessarily the shortest reverse path, nor the one resulting in the best QoS. Finally, *Dumb Packets (DP)* carry payload and use dynamic source routing. The route brought back by an ACK is used as a source route by subsequent DPs of the same QoS class having the same destination, until a newer AND/OR better route is brought back by another ACK. A *Mailbox (MB)* in each node is used to store QoS information. Each MB is organized as a Least-Recently-Used (LRU) stack, with entries listed by QoS class and destination, which are updated when an ACK is received. The QoS information is then used to calculate the reward in the SP routing algorithm. We use recurrent random neural networks(RNN) [5] with reinforcement learning (RNNRL) in order to implement the SP routing algorithm. Each output link of a node is represented by a neuron in the RNN. The arrival of *Smart Packets(SPs)* triggers the execution of RNN and the output link corresponding to the most excited neuron is chosen as the routing decision. The weights of the RNN are updated so that decisions are reinforced or weakened depending on how they have been observed to contribute to the success of the QoS goal. The RNN is an analytically tractable spiked random neural network model whose mathematical structure is akin to that of queuing networks. It has "product form" just like many useful queuing network models, although it is based on non-linear mathematics.

The experimental results concerning search in CPN that we presently from a recent paper [21] use the test-bed consisting of 17 nodes shown in Figure 2. Each pair of INRs is connected by point-to-point 10Mbps Ethernet links. All tests were performed using a flow of UDP packets entering the network at constant bit rate (CBR) with $1024B$ packets. Each measurement point is based on $10,000$ packets that were sent from the source to the destination, and we inserted random background traffic into each link in the network with the possibility of varying its rate. The CPN routing algorithm is used throughout the experiments using three

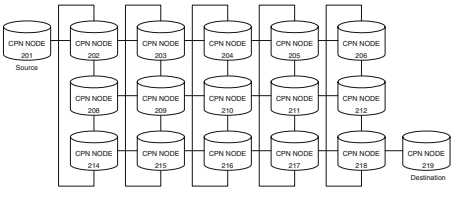

Fig. 2. The test-bed topology used in the experiments

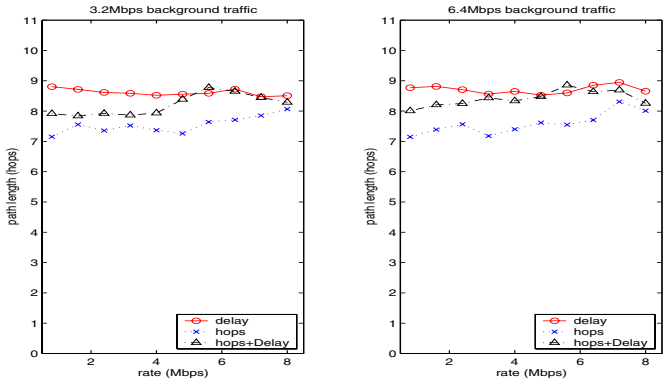

Fig. 3. Path length comparison

different QoS goals: (a) delay [*Algorithm-D*], (b) hop count [*Algorithm-H*] and (c) the combination of hop count and forward delay [*Algorithm-HD*]. Measurements concern average hop count, the forward delay and packet loss rate under different background traffic conditions. From Figure 2, we see that the shortest path length from the source node (#201) to the destination node (#219) is 7, and there are only five distinct shortest paths. For example, one of them is route $\langle 201 \rightarrow 202 \rightarrow 214 \rightarrow 215 \rightarrow 216 \rightarrow 217 \rightarrow 218 \rightarrow 219 \rangle$. Figure 3 reports the average number of hops traversed from source to destination when different algorithms are used. When hop count is used as the QoS goal, we see that the average number of hops under different background traffic conditions is close to the minimum of 7. It is interesting to observe that whenn the forward delay is used as the QoS goal, the average number of hops actually used is no longer the minimum number. We see that the average path length is close to 7 when the connection's traffic rate is low or medium, and close to 9 hen it is high. With respect to path length the experiments confirm our expectations: *Algorithm-H* is the best, and *Algorithm-HD* is better than *Algorithm-D*. These results show that one need not use fixed non-adaptive algorithms for routing; it is possible to find shortest paths adaptively without fixed prior knowledge. The comparison of an adaptive algorithm using the number of hops, or the path delay, or a mixture of the two as the way to select paths, also provides some interesting insight. Note however that in our case we are not always using the *same* shortest path, since the adaptive routing algorithm will be able to vary the paths it is using even when it is instructed to use find the shortest path. As a result, even the adaptive shortest path algorithm should be able to improve observed QoS over a fixed shortest path, since it will distribute traffic over a larger number of paths.

4 Evaluating the Search Time

As mentioned earlier, searching for objects, geographic locations, data, and so on, is becoming of fundamental importance in all areas of networking. The sheer number of different services, nodes and networks, and the mobility and variability

of all of these entities, will make it impractical to find them via some fixed addressing or routing scheme. However, other approximate characteristics such as their location or movement patterns can help us find the objects we seek. If one deals with routing in a packet network, one can augment an internet address with some information about its physical location. In a wireless ad hoc network and in a wireless sensor network, the physical location of nodes is an important element of information when one tries to convey data or packets to or from a geographic area.

For instance, if you consider two physical locations $o = (0, 0)$ and $d = (N, M)$ in the integer valued (x, y) plane, in an unconstrained routing scheme (i.e. one that allows the packet to go to any neighbouring point which it can reach) the packet progresses from point o with the aim of reaching d, and at some intermediate time it reaches a location (x, y). Its next step will be to move to one of its reachable and available neighbours, and it will prefer a neighbour which is in the direction of the destination d. An exact probabilistic or combinatorial analysis of the time it would take the packet to reach d from o appears difficult. Thus in this section we develop a model that represents the search process so as to estimate the time it would take to find a destination, or more generally an object, in a search space. The model will be based on a continuous space and time diffusion process.

For the migration process of a packet from o to d that begins at $t = 0$, what matters is the distance of the packet to its destination at some time $t > 0$. This distance will be represented by the real valued stochastic process $Y = \{Y(t) : \ t \geq 0\}$, where $Y(0) = D$, $D = ||d - o||$. Quantities of interest include:

- $T_1 = inf\{t : \ Y(t) = 0\}$, the time it takes the packet to reach its destination,
- $\Pi(\delta) = P[Y(t) > \delta, 0 \leq t < T_1]$ for $\delta > D$, the probability that the packet has gotten too far away from its destination, and
- the probability $\pi(\epsilon) = P[Y(t) < \epsilon), 0 \leq t < T_1]$ that the packet is within an $\epsilon - neighbourhood$ of its destination.

To simplify the analysis, we replace the transient process Y by an ergodic process Z which will allow us to compute all these quantities of interest.

Consider the process $Z = \{Z(t) : \ t \geq 0\}$ which is identical in sample path to Y until time T_1. After T_1 the process Z will reside at the point $z = 0$ for a random time H_1, after which Z jumps to point D and then stochastically repeats its previous behaviour. Let H_i, $i = 1, 2, \ ..$, be independent and identically distributed positive random variables, and let $T_{i+1} = inf\{t : \ T_{i+1} > T_i, \ Z(T_{i+1}) = 0\}$ for $i = 1, 2, \ ...$ Then Z has the renewal property:

$$P[Z(t) > z] = P[Z(t + T_i + H_i) > z] \tag{1}$$

for any $t \geq 0$, $z \geq 0$, and the instants $\{T_i + H_i\}$ are renewal instants of the process for $i \geq 1$. The random behaviour of the search packet can be represented by a Brownian motion in one dimension, where the distance of the search packet to its destination is the dimension being considered[1, 2]. However, in order to take into account the holding time at the boundary $z = 0$ described above, the process Z

will be represented as a Brownian motion [4] which is modified to have holding times at the boundary $z = 0$ and jumps to interior points as suggested in [3]. Thus in addition to the usual diffusion equation, the process we consider will have a discrete (i.e. not continuous) component as described below. The diffusion process representing the distance of the search packet, or of the searcher, from its destination at time t is defined as follows:

- We assume that on the average each move of the packet gives preference to a direction which reduces the distance to the destination. This is a reasonable assumption because, if the medium is isotropic, it is natural for the packet to select a direction of motion which brings it closer to its destination whenever it can. However, either for lack of knowledge or because other options are impossible, a move may sometimes increase its distance to the destination. Thus the drift parameter b of the diffusion is negative. In the sequel we will also consider the case where $b = 0$, which corresponds to "ignorance on the average".
- Its second moment parameter is some finite quantity $c \geq 0$. Note that $c = 0$ represents the case where the time duration of each step from neighbour to neighbour is constant, while a large value of c would imply a more erratic search process.
- Suppose that at some time $t = T_i$ the destination is reached, i.e. $Z(T_i) = 0$; then after a random time H_i we assume that the search process starts again, so that at $t = T_i + H_i$ the process Z jumps back to the starting point of the search and $Z(T_i + H_i^+) = D$, and the search process is re-initialised. This process repeats itself indefinitely.
- Without loss of generality with respect to the computation of $E[T]$, we will assume that $P[H_i > v] = e^{-v}$, for $v \geq 0$, so that $E[H_i] = 1$.

Result 1. Let $E[T] = E[T_i]$ for any i, and let

$$P = \lim_{t \to \infty} P[Z(t) = 0]. \tag{2}$$

Then:

$$E[T] = \frac{1}{P} - 1. \tag{3}$$

Sketch of Proof. This follows from the fact that the process $\{X(t),\ t \geq 0\}$ defined by $X(t) = 1[Z(t)]$ is a two state (0 and 1) semi-Markov process, where $1[y]$ is the characteristic function $1[y] = 1$ if $y > 0$, and $1[y] = 0$ otherwise.

We will skip the details of the representaton of the process Z which is based on the equations that the probability density function $f_{z,t}dz = P[z < Z(t) \leq z+dz]$, $z > 0$, $t \geq 0$ must satisfy. We will just summarise the main analytical results that we have obtained:

Result 2. The average search time is given by the expression:

$$E[T] = \frac{D}{-b} \tag{4}$$

This result assures us about the feasibility of a search: it tells us that as long as $b < 0$ then the search time will be finite on the average. Although it has a very simple and intuitive form (note that we have assumed that b is negative), it does tell us that the average time it will take for the search to be successful is simply the distance D to the destination, divided by the average distance traversed per unit time. Thus a detailed knowledge of the way the movement occurs towards the destination (e.g. the second moment of the distance traversed per unit time) is not needed. However, we will see that if packets are subject to a time-out so that they self-destroy if they have been travelling for too long a time, then average time to reach the destination will also depend on the second moment of the diffusion process.

In many cases, some mechanism will be incorporated into a search packet so that it is destroyed if it has meandered for too long a time, or too far away, and has not found its destination. We incorporate this property in the diffusion model, so that at any distance z from the origin, $r(z)dt$, with $r(z) \geq 0$, is the probability that the search packet is destroyed in the interval $[t, t + dt[$ when it is at distance z from its destination. We can now use a similar artifact as previously to compute $E[L]$ which is the new value of the average time it takes the packet to find its destination. The artifact now is:

- As before, after the search packet reaches its destination, wait for an exponentially distributed random time of average value one and then generate a new search packet so as to re-start the search process, and
- Generate a new search packet immediately after it is destroyed by the time-out.

Result 3. Assuming an exponentially distributed time-out of average value λ^{-1}, and $b < 0$, the average time for a search packet to reach its destination is given by

$$E[L] = \frac{1}{P} - 1 \tag{5}$$
$$= \frac{-2D}{b - \sqrt{b^2 + 2\lambda c}}.$$

Notice from (6) that, contrary to (3), the average time that a packet reaches destination now depends on the variance parameter c of the diffusion process. Furthermore, when $\lambda = 0$, i.e. when the time-out is in effect removed, we revert as expected to *Result 2* given in (3). Notice also that if $\lambda > 0$ and $b \to 0$, then:

$$E[L] = D\sqrt{\frac{2}{\lambda c}}, \tag{6}$$

which says that even though each step of the search does not, on the average, get the search packet closer to the destination, the fact that we use the time-out mechanism does allow us to get to the destination in a time which is finite on the average due to the repeated usage of the time-out. Furthermore the expression in (6) can also be used as an approximation when $2\lambda c >> |b|$ and $b \leq 0$.

5 Conclusions

We present an architecture for autonomic networks which offers a universal peer-to-peer communication environment for users and services, composed of Intelligent Network Routers capable of supporting the user and service needs. The network allows users to sense and adapt network paths, and identify user to service connections, dynamically as a function of network state and of user and service quality of service needs. The architecture uses smart packets for the search for services, and for on-line dynamic sensing and control. These ideas are extrapolated from an experimental test-bed for QoS driven network routing, the CPN system, which is based on similar concepts with completely decentralised control. We then study the search process itself in order to estimate the time it would take to find another user or a service in the network which was initially at distance D from the object conduction the search. We assume that the search is conducted with a search packet which moves through the network. We model the search via the distance to the destination, some time t after the search begins. Closed form analytical results are derived for the average search time as a function of the initial distance from the point from which the search is being initiated, to the point where the object being looked for is to be found. We consider the case where time-outs are used to destroy packets that have been in the network for too long without reaching their destination, and are then replaced with fresh packets, as well as the case where time-outs are not used.

References

1. A. Einstein. Investigations on the Theory of Brownian Motion. Dutton, New York, 1926, reprinted by Dover, New York, 1956.
2. W. Feller. An Introduction to Probability Theory and its Applications, Vols. I and II. Wiley, New York, 1966.
3. E. Gelenbe. On approximate computer system models. *Journal ACM*, 22 (2), pp. 261-269, April 1975.
4. J. Medhi. Stochastic Models in Queueing Theory. pp. 373 ff. Academic Press, New York, 1991.
5. E. Gelenbe. Learning in the recurrent random neural network. *Neural Computation*, 5 (1), pp. 154–164, 1993.
6. S. Alpern. The rendezvous search problem. *SIAM J. Control & Optimization* 33, 673-683, 1995.
7. D. Williams and G. Apostolopoulos. QoS Routing Mechanisms and OSPF Extensions. RFC 2676, Aug. 1999.
8. H. Yaiche, R.R. Mazumdar and C. Rosenberg. A game theoretic framework for bandwidth allocation and pricing in broadband networks. *IEEE/ACM Transactions on Networks* 8 (5), pp. 667-678, 2000.
9. N. Semret, R. R.-F. Liao, A. T. Campbell and A. A. Lazar Pricing, provisioning and peering: dynamic markets for differentiated Internet services and implications for network interconnections. *IEEE J. Sel. Areas Comms.* 18 (12), pp. 2499-2513, 2000.
10. E. Gelenbe, R. Lent and Z. Xu. Measurement and performance of a cognitive packet network. *Computer Networks*, 37, pp. 691-791, 2001.

11. E. Gelenbe, R. Lent, and Z. Xu. Cognitive Packet Networks: QoS and performance. *Proc. IEEE MASCOTS Conference*, ISBN 0-7695-0728-X, pp. 3-12, Fort Worth, TX, Oct. 2002.
12. N. Christin and J. Liebherr. A QoS architecture for quantitative service differentiation. *IEEE Comms. Mag.* 46 (6), pp. 38–45, 2003.
13. S. Alpern and V. Baston. A common notion of clockwise helps in planar rendezvous. Rendezvous on a Planar Lattice. *CDAM Research Report Series 2004-7*, Centre for Discrete & Applicable Mathematics, London School of Economics, 2004.
14. Autonomic Computing Initiative, http://www.ibm.com/autonomic/
15. A. Galis, S. Denazis, C. Brou and C. Klein (eds). Programmable networks for IP service deployment. Artech House Books, ISBN 1-58053-745-6, 2004.
16. A. Asgari, R. Egan, P. Trimintzios and G. Pavlou. Scalable monitoring support for resource management and service assurance. *IEEE Network* 18 (6), pp. 6–18, 2004.
17. E. Gelenbe, M. Gellman, R. Lent, P. Liu, Pu Su. Autonomous smart routing for network QoS. *Proc. First International Conference on Autonomic Computing*, (IEEE Computer Society), ISBN 0-7695-2114-2, pp. 232-239, May 17-18, 2004, New York.
18. S.K. Das, H. Lin and M. Chaterjee. An econometric model for resource management in competitive wireless data networks. *IEEE Network* 18 (6), pp. 20–26, 2004.
19. E. Gelenbe, R. Lent, A. Nunez. Self-aware networks and QoS. *Proceedings of the IEEE*, 92 (9), pp. 1478-1489, 2004.
20. E. Gelenbe. Cognitive Packet Network. *U.S. Patent No. 6,804,201 B1*, Oct. 12, 2004.
21. E. Gelenbe and P. Liu. Qos and routing in the cognitive packet network. To appear in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Jun 2005.