Users and Services in Intelligent Networks

Erol Gelenbe
Memb. Acad. Europ., FIEE FIEEE FACM
Dennis Gabor Chair
Intelligent Systems and Networks Group
Electrical & Electronic Engineering Dept.
Imperial College
London SW7 2BT
e.gelenbe@imperial.ac.uk

October 11, 2005

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) and price. Users can also monitor on-line the extent to which their requests are being satisfied. In turn the services will dynamically try to satisfy the user 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 locally distributed feedback control which proceeds at the same speed as the traffic flows and events being controlled. We review issues such as network "situational awareness", self-organisation, and structure, and relate these concepts to the ongoing research on autonomic communication systems. We relate the search for services in the network to the question of QoS and routing. We examine the need to dynamically protect the networked system from denial of service (DoS) attacks, and propose an approch to DoS defence which uses the detection of violations of QoS constraints and the automatic throttling or dropping of traffic to protect critacl nodes. We also discuss how this vision of an Intelligent Network can benefit from techniques that have been experimented in the Cognitive Packet Network (CPN) test-bed at Imperial College, thanks to "smart packets" and reinforcement learning, which offers routing that is dynamically modified using on on-line sensing and monitoring, based on users' QoS needs and overall network objectives.

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 improve and add value to human beings – or even beyond humans to other living beings.

In the field of telecommunications, 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 standards for entertainment by forcing a limited number of programs upon the public; it has often displaced reading, sophisticated cinema, theatrical and musical forms by the introduction of facile talk shows and soap operas. This is a great example of a tremendous success in technology which has not always been applied in the most broadly intelligent manner. The "one-to-very-many" broadcast nature of television does not give users, or communities of users, the possibility to significantly influence the system that they use. Other models of communications, such as the peer-to-peer concept which was born in the Internet, can offer a greater degree of user choice. Thus we suggest that through intelligent organisation of networks, which should include a just compensation for services and intellectual property ownership, one can achieve improved communications for the sake of an enhanced cultural and humanistic environment.

We envision Intelligent Networks (INs) 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 current 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. In fact 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 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 will offer the facilities for an intelligent and 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.

2 Towards Autonomic Communications

The growth in both communication needs and communication-intensive technologies and services through which we are living is a trend that is bound to continue and to intensify. In a not-so-distant future, everyday activities will be supported by a ubiquitous ITC environment or "networked service" that will cater dynamically to our needs in a situation-aware manner.

The Internet is becoming an immense organism of composite, highly distributed, pervasive, communication intensive services; in order to operate effectively, these services need to [27, 46] autonomously detect and organize the knowledge necessary to understand the physical, as well as human user based and social context. Services will need to autonomously adapt to the human, social and technical environment. They should improve our ability to interact with the world by providing us with any needed information about our surrounding physical environment. Imagine a traveller arriving at some location with given objectives (e.g. go to a meeting at hotel X, then lunch with A and restaurant Y with stop-over to do some work at coffee shop Z)) [23] and imagine how the relevant information can be collected by the communication intensive service via local networks, web sites, as well as the user's office system, possibly the ITC system related to the local or the organisation that is hosting the meeting and of person B, and then displaying all the situational information to the traveller on his laptop, mobile phone or PDA. Imagine also that this is done by using the best and cheapest network service, i.e. at the lowest cost and the best possible QoS [26, 38].

Such future services raise major challenges for today's networks, including protocol mismatches and differences that may exist between networks in terms of operational and behavioural semantics at the service level. The technical premises for these communication intensive services will require the integration into the networking world of technologies borrowed from other application domains and from research in ITC, including:

- *Situational awareness* which is a concept borrowed from defence applications. It represents the ability to understand, represent and evaluate the detailed context in which an operation will be conducted. In the military domain, situational awareness is designed to enhance and support the human in the loop, and is also designed in a manner which encompasses the role of human factors. In the present context the human factors aspects can also be important if we wish to provide a certain level of human control and decision making in the context of communications. Technology that can support situational awareness in communications includes sensors, location systems, user profiles, and tools for system and network monitoring [30], which can be utilised for the provisioning of adaptive services [49, 45].

- *Self-organization* will be required of such networks to be able to automatically exploit to their ability for situational awareness. However, this will need to go beyond the structural self-organisation that has already been addressed in communications in the context of Peer-to-Peer networks and computing [11], with techniques inspired from ant-based optimisation, reinforcement learning, and social systems [4, 5, 12, 21]. The systems we consider should create a bridge between robust but conceptually simple self-organising models and the far more complex semantics [18, 43] that context-aware systems will require.

- *Structure and Components* The traditional top-down hierarchical protocol stack that adopts a vertical hierarchy, going down layer by layer from the application to the physical connection at the bottom, that has dominated our thinking about networks for the last thirty years, must probably be abandoned in favour of a model composed of a collection of agents or "autonomic components". Such components could use common templates [27, 37] to provide both a conceptual and software framework for programmable network elements and their interactions. Although these ideas are still in their formative stage, they can benefit from research in other areas such as multi-agent systems [28] and programmable networks [7, 22].

3 A Simple Architecture for the Intelligent Network

A sketch of the IN architecture is shown in Figure 1. For reasons of backward compatibility, the IN will have to offer at least the appearance of a standard communication interface inspired by the Internet Protocol (IP). End users U (shown with small purple rectangles as U1, U2, etc.) will generally be mobile; they can be recognised via their ID and password. Users may have a "credit value" 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 Fugure 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 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:

- Most end users will in general be light-weight (a mobile phone, a PDA, or just a user ID and password).

- Other end users and services will be much more complex and may often be resident on one or more INRs. They may own one or more INRs for their needs.

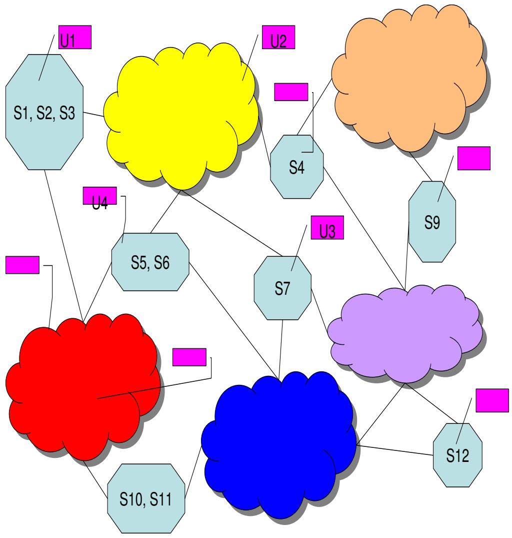

Figure 1: Architecture of the Intelligent Network

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.

4 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. If the user does not know how to find a service, in the worst case it can broadcast its request which will be relayed by INRs and directories. However som of the "smart probing" techniques described below can offer a more efficient solution.

4.1 Smart search and routing

Let us describe how search can be "smart", by extending some ideas from the Cognitive Packet Network (CPN) algorithm [5, 42]. In CPN, the purpose of the search is to find a network destination (rather than a service), and SPs and ACKs in CPN play a role that is identical the one we described earlier, except that we are looking for a path to some destination which optimises a QoS requirement. Thus we can imagine that the CPN algorithm which runs at the packet transport level and finds destination nodes, can be abstracted to a higher level where it searches for services.

In order to provide a practical grounding for the preceding discussion, we will discuss how the CPN protocol currently runs; more detail can be found in the collection of papers that are stored in the web site at $http : //san.ee.ic.ac.uk$. In CPN dumb packets (DPs) carry the payload traffic, while CPN routers are similar to INRs, and are interconnected either via portions of the Internet which plays the same role as the network clouds that we have described in Figure 1, or via point-to-point Ethernet or other (e.g. ATM) connections. SPs find routes and collect measurements, but do not carry payload.

DPs are source routed, using paths which best match the users' QoS requirements. On the other hand, SPs are routed using a Reinforcement Learning (RL) algorithm that uses the observed outcome of previous decisions 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 acknowledgement (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. The route brought back by an ACK is used as the 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 steps needed to establish a connection between some user U and a service S can then be listed as follows:

- 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 SP 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.

5 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.

In the aproach that we have suggested for finding services, the user U formulates some request (SX, QY, PZ) for a service SX at quality level QY and for the price PZ. Both the quality of service value and the price constitute “goals” in the sense that the term is used in the CPN algorithm [39]. They may be treated as separate goals to be minimised, and combined in some manner as outlined above, or combined into some single common metric.

For instance, if QY is some non-negative number such as “loss” or “delay”, we could combine the two considerations in a single metric such as $G = QY/PZ$ (quality for a given price), or as

$$G = PZ\ 1[QY < Qmax] + \frac{QY}{PZ} 1[QY \geq Qmax] \qquad (1)$$

where $1[x]$ is the function which takes the value one if the predicate $x = true$ and takes the value zero if $x = false$. Thus (1) means, for instance, that as long as the delay is less than some maximum acceptable value $Qmax$, we are happy to minimise the price; however if the delay is larger than this maximum value, we want simply to minimise the delay per price unit that we pay for the service.

5.1 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.
- 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

[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.

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.

6 Protecting the IN Against Denial of Service Attacks

Denial of Service (DoS) attacks are known to the network research community since the early 1980s. Although initially DoS attacks were the act of hackers who wanted to demonstrate their ability and power over computer systems, they have now become an important weapon in the hands of cyber-criminals and for cyber-warfare. DoS attacks have reportedly been used against business competitors, for extortion purposes, for political reasons, and even as a form of “legitimate” protest. It is this variety of targets and types of attack that dictate the need for flexible defence systems which can react according to both the attacker’s aims and the defender’s needs.

A DoS attack is very often distributed (DDoS): the attacker takes control of a large number of networked computers and orders them to send a large number of packets to a specific target node, server or web site. Typical targets of an attack may include servers or web sites which accomplish some function in the public interest, or the servers of e-commerce web-sites which can suffer significant financial loss. Other targets may be news web-sites, corporate networks, banks, etc. Often, the attacker needs to conceal its identity and the nodes that are used in the attack. It can do this does by introducing fake IP addresses into the packets (“IP spoofing”), providing a false identity for the nodes that generate the attack. Indeed, in a seminal paper on some of the weaknesses of the TCP/IP protocol [1] it is said that (quote)“The weakness in this scheme (the IP protocol) is that the source host itself fills in the IP source host id, and there is no provision to discover the true origin of the packet”.

As a result of an attack, the target’s links and other routers in its neighbourhood are overwhelmed, and a number of legitimate users’ traffic may be lost or backlogged. One of the first defensive measures proposed for DoS attacks is Ingress Filtering, which drops arriving packets if their IP addresses lie outside an acceptable range []. Another technique is IP traceback [8, 24, 33] uses probabilistic packet marking to allow the victim to identify the network path traversed by attack traffic without requiring interactive operational support from Internet Service Providers (ISPs). However false traceback messages can also be injected into the packet stream by attackers. IP traceback can also be used “post-mortem” [15] after an attack in order to understand how the attack has been conducted.

6.1 Protection and Defence Against DoS Attacks

Since attacks occur rapidly and unexpectedly, it is essential for the IN to incorporate an autonomic approach to DoS defence based on network self-observation and adaptive reaction. The effective protection of the IN from DoS attacks requires the combination of different elements:

- ***Detection*** of the existence of an attack. The detection can be either anomaly-based or signature-based, or a hybrid of those two. In anomaly-based detection, the system recognises a deviation from the standard behaviour of its clients, while in the latter it tries to identify the characteristics of known attack types.

- ***Classification*** of the incoming packets into valid (normal packets) and invalid (DoS packets). As in detection, one can choose between anomaly-based and signature-based classification techniques.

- ***Response*** The protection system should either drop the attacking packets or redirect them into a trap for further evaluation.

As a consequence:

- A node which is undergoing a DoS attack (the victim node) must have the ability to detect or to be informed about the attack, based either on a local or distributed detection scheme. All nodes upstream, from the victim up to the source(s) of the attack, should also be informed of the ongoing threat and they should be incorporated as much as possible into the defence scheme. The detection scheme is always imperfect, so that both false alarms and detection failures are possible. Imperfections are possible both with regard to the detection of the attack as a whole, and the identification of the packets that belong to this attack. Thus some attacking packets will be missed and some non-attacking packets may be incorrectly dropped.

- The victim node and the informed nodes must react by *dropping packets* which are thought to be part of the attack, or by diverting them to a sink node whose role is to absord such attack packets.

- The attack itself, if not properly deviated or eliminated, will produce buffer overflows and saturation of network resources such as CPU capacity, due to the inability of the nodes or routers to handle the resulting overwhelming traffic.

6.2 Using CPN for Defence against DoS Attacks

As we will briefly show in the sequel, a protocol such as CPN, because of its ability to react rapidly to changes in QoS, is particularly well adapted for DoS defence [50]. In CPN, each flow specifies its QoS requirements in the form of a QoS “goal”. SPs associated with each flow constantly explore the network, and obtain routing decisions from network routers based on observed relevant QoS information. SPs store the identities of the nodes they visit, and collect local measurements such as times and loss rates. At each CPN node, the SP uses a local reinforcement learning algorithm based on measurements collected by previous SPs and ACKs, to elicit a decision from the node as to the next hop to travel to. When a SP reaches the destination node of the flow, an ACK packet is generated and returned to the source according to the opposite (destination to source) path traversed by the SP from which all repeated nodes are removed. When an ACK reaches the source, the forward route is now stored and used by subsequent payload or dumb packets (DPs), which are source-routed to the destination, until a new ACK packet brings back a new route.

The CPN-based DDoS defence technique exploits the ability of CPN to trace traffic going both down- and up-stream thanks to SPs and ACK packets. When a node detects an attack, it uses the ACKs to ask all intermediate nodes upstream to drop the packets of the attack flow. Each node is allowed to select the maximum bandwidth that it will accept from any flow that terminates at the node, and the maximum bandwidth that it allocates to a flow that traverses the node. These parameters may vary dynamically as a result of other conditions, and they can also be selected based on the identity and the QoS needs of the flows. When a node receives a SP or DP from a flow that it has not previously encountered (e.g. with a new source-destination pair, or a new QoS class), it sends a Flow-ACK packet back to the source along the reverse path, and informs the source of its bandwidth allocation. The node monitors the flows that traverse it, and drops packets of any flow that exceeds the allocation; it may also inform upstream nodes that packets of this flow should be dropped. Other possible actions include diverting the flow into a "honeypot", or to a special network.

We illustrate the effect of this defence scheme with the following example. A fixed rate MPEG video stream is transferred from node 3 to node 30 in the CPN testbed shown in Figure 2 (top left). The video is initially uncorrupted by an attack, as shown in Figure 2 (top middle). Then, a DDoS attack is launched from nodes 1 and 2 to node 30. Without defence, the attack corrupts the video stream, making it unintelligible as shown in Figure 2 (top right). If node 30 does detect the attack because of the high traffic rate on incoming paths, it defends itself using the CPN trace-back mechanism, orders the packets to be dropped upstream, and drops the traffic coming into itself along those paths. The impact of the attack is reduced, which results in the clear video stream shown in Figure 2 (bottom right). The experiment shows CPN, or a similar adaotive network routing mechanism, coupled with the ability to detect an attack, can protect a sensitive real-time data stream.

7 Conclusions

We present an architecture for Intelligent Networks (INs) which offers a flexible and self-organising communication environment for users and services. The IN is composed of Intelligent Network Routers capable of supporting the user and service needs, and able to sense and adapt network paths and user to service connections dynamically as a function of network state and user and service quality of service needs. It uses smart packets for the search for services, as well as to offer QoS using on-line dynamic sensing and dynamically adaptive control. We suggest that these ideas can be supported by systems that incorporate some of the techniques offered by the CPN system. We also discuss the critical issue of denial of service attack and defence for such systems and show how QoS based adaptive techniques can help mitigate their effect. Other important issues, such as the energy efficient operation of wireless networks [41] have not been discussed in this paper.

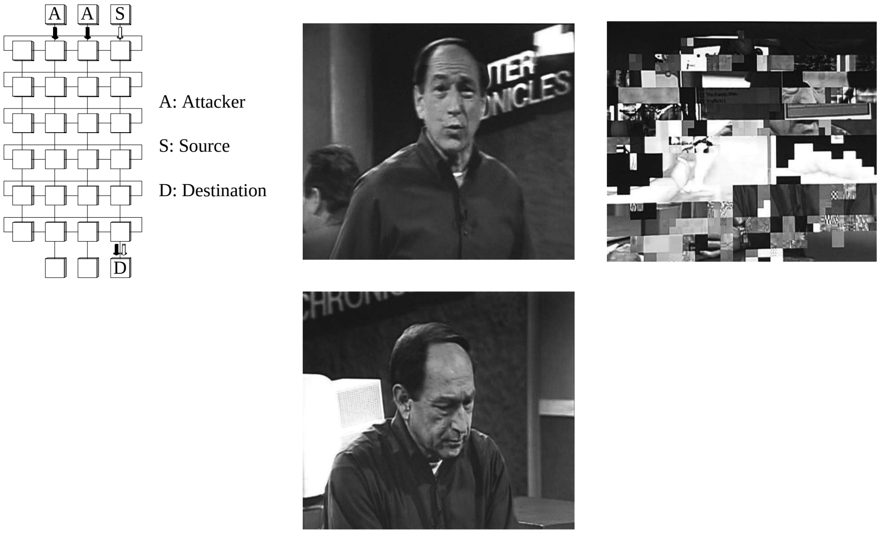

Figure 2: Experimental evaluation of our defence scheme. The top-left figure shows the CPN testbed that is used in the experiment. Top-middle picture shows a frame of video when there is no attack, while the top-right image is the sams video frame as seen during an attack with no defense. At the bottom we see a video frame during an attack, but with the defence mechanism.

References

[1] R.T. Morris. A Weakness in the 4.2BSD Unix TCP/IP Software. *Technical Report Computer Science #117*, AT&T Bell Labs, February 1985.

[2] P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing. *Tech. Rep. RFC 2267*, January 1998.

[3] D. Williams and G. Apostolopoulos. QoS Routing Mechanisms and OSPF Extensions. RFC 2676, Aug. 1999.

[4] E. Bonabeau, M. Dorigo, G. Theraulaz. Swarm Intelligence: From Natural to Artificial Systems. New York, NY, Oxford University Press, 1999.

[5] E. Gelenbe, Z. Xu, E. Şeref. Cognitive packet networks. *Proc. 11th IEEE Int. Conf. on Tools with Artificial Intelligence (TAI99)*, 47-54, Chicago, Ill., 1999.

[6] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical Network Support for IP Traceback. *Proc. ACM SIGCOMM*, 295-306, Stockholm, Sweden, August 2000.

[7] C. Tschudin, H. Lundgren, H. Gulbrandsen. Active Routing for Ad Hoc Networks. IEEE Communications Magazine, April 2000.

[8] D. Song and A. Perrig. Advanced and Authenticated Marking Schemes for IP Traceback. *Proc. Infocom 2001*, ISBN: 0-7803-7016-3, vol. 2, pp. 878-886, Anchorage, Alaska, USA, 22-26 April 2001.

[9] T. Berners-Lee, J. Hendler, O. Lassila. The Semantic Web. Scientific American, May 2001.

[10] G. P. Picco, A. L. Murphy, G. C. Roman. LIME: a Middleware for Logical and Physical Mobility. 22nd IEEE Intl. Conference Distributed Computing Systems, 2001.

[11] S. Ratsanamy,, P. Francis, M. Handley, R. Karp. A Scalable Content-Addressable Network. ACM SIGCOMM Conference, Aug. 2001.

[12] E. Gelenbe, E. Seref and Z. Xu. Simulation with learning agents. *Proceedings of the IEEE*, 89 (2), pp. 148-157, 2001.

[13] V. Paxson. An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks. *ACM Computer Communications Review 31(3)*, July 2001.

[14] E. Gelenbe, R. Lent and Z. Xu. Design and performance of cognitive packet networks. *Performance Evaluation*, 46, pp. 155-176, 2001.

[15] G. Rice and J. Davis. A Genealogical Approach to Analyzing Post-Mortem Denial of Service Attacks. *Secure and Dependable System Forensics Workshop*, University of Idaho, September 23-25, 2002.

[16] BBC News. Mafiaboy hacker jailed. (September 13, 2001), http://news.bbc.co.uk/1/hi/sci/tech/1541252.stm.

[17] 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.

[18] I. Horrocks, P. Patel-Schneider, F. van Harmelen. Reviewing the design of DAML+OIL: An ontology language for the semantic web" National Conference on Artificial Intelligence, Edmonton, Alberta, Canada, 2002.

[19] R. Mahajan, S. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker. Controlling High Bandwidth Aggregates in the Network. *ACM SIGCOMM Computer Communication Review*, ISSN: 0146-4833, Vol. 32, Issue 3, pp. 62–73, July 2002.

[20] 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, October 2002.

[21] R. Albert, A. Barabasi. Statistical Mechanics of Complex Networks", Rev. Mod. Phys. 74 (47), 2002.

[22] C. Borcea, et al.. Cooperative Computing for Distributed Embedded Systems", 22nd International Conference on Distributed Computing Systems, Vienna (A), IEEE CS Press, 227-238, 2002.

[23] D. Estrin, D. Culler, K. Pister, G. Sukjatme. Connecting the Physical World with Pervasive Networks", IEEE Pervasive Computing, 1 (1): 59-69, Jan. 2002.

[24] A. Snoeren, C. Partridge, L.A. Sanchez, C.E. Jones, F. Tchakountio, B. Schwartz, S. Kent, and W.T. Strayer. Single-Packet IP Traceback. *IEEE/ACM Transactions on Networking*, ISSN: 1063-6692, Vol. 10, no. 6, pp. 721-734, December 2002.

[25] W.G. Morein, A. Stavrou, D.L Cook, A.D. Keromytis, V. Mishra, and D. Rubenstein. Using Graphic Turing Tests to Counter Automated DDoS Attacks against Web Servers. *Proc. 10th ACM Int'l Conference on Computer and Communications Security (CCS '03)*, ISBN: 1-58113-738-9, pp. 8-19, Washington DC, USA, October 27-30, 2003.

[26] L. Capra, W. Emmerich, C. Mascolo. CARISMA: Context-Aware Reflective mIddleware System for Mobile Applications", IEEE Transactions of Software Engineering Journal (TSE) 29(10):929-945, 2003.

[27] J. Kephart, D. Chess. The Vision of Autonomic Computing", IEEE Computer, 36 (1), 2003.

[28] F. Zambonelli, N. Jennings, M. Wooldridge. Developing Multiagent Systems: the Gaia Methodology", ACM Transactions on Software Engineering and Methodology, 12 (3):317-370, 2003.

[29] M. Papazoglou, M. Aiello, M. Pistore, J. Yang. XSRL: A Request Language for Web Services (www.webservices.org)", 2003

[30] M. Philipose, K. Fishkin, M. Perkowitz, D. Patterson, D. Fox, H. Kautz, D. Hahnel. Inferring Activities from Interactions with Objects", IEEE Pervasive Computing, 3(4):50-57, 2004.

[31] S. Jing, H. Wang, and K. Shin. Hop-Count Filtering An Effective Defense Against Spoofed Traffic. *Proc. ACM Conference on Computer and Communications Security*, pp. 30-41, ISBN 1-58113-738-9, Washington DC, October 2003.

[32] G. Mori and J. Malik. Recognizing objects in adversarial clutter - Breaking a visual CAPTCHA. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2003 (CVPR '03)*, ISSN: 1063-6919, ISBN: 0-7695-1900-8, vol. 1, pp. 134-141, Madison, WI, USA, June 18-20, 2003.

[33] M. Sung and J. Xu. IP Traceback-Based Intelligent Packet Filtering: A Novel Technique for Defending against Internet DDoS Attacks. *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, pp. 861-872, September 2003.

[34] R. Thomas, B. Mark, T. Johnson, and J. Croall. NetBouncer: client-legitimacy-based high-performance DDoS filtering. *Proc. DARPA Information Survivability Conference and Exposition*, vol. 1, pp. 14-25, April 22-24, 2003.

[35] A. Hussain, J. Heidermann, and C. Papadopoulos. A Framework for Classifying Denial of Service Attacks. *Proc. ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication 2003*, ISBN: 1-58113-735-4, pp. 99-110, Karlsruhe, Germany, August 25-29, 2003.

[36] J. Mirkovic, P. Reiher, and M. Robinson. Forming Alliance for DDoS Defense. *Proc. 2003 workshop on New security paradigms*, 11-18, ISBN 1-58113-880-6, Ascona, Switzerland, August 2003.

[37] H. Liu, M. Parashar. Component-based Programming Model for Autonomic Applications. *Proc. First International Conference on Autonomic Computing*, New York, NY, USA, 2004.

[38] M. Mikic-Rakic, N. Medvidovic. Support for Disconnected Operation via Architectural Self-Reconfiguration. *Proc. First International Conference on Autonomic Computing*, (IEEE Computer Society), ISBN 0-7695-2114-2, New York, 2004.

[39] 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, 232-239, New York, 2004.

[40] E. Gelenbe, R. Lent, A. Nunez. Self-aware networks and QoS. *Proceedings of the IEEE*, 92 (9), pp. 1478-1489, 2004.

[41] E. Gelenbe and R. Lent. Adhoc power aware Cognitive Packet Networks. *Ad Hoc Networks Journal*, Vol. 2 (3), pp. 205–216, 2004 (ISN: 1570-8705).

[42] E. Gelenbe. Cognitive Packet Network. *U.S. Patent No. 6,804,201 B1*, Oct. 12, 2004.

[43] J. Frey, G. Hughes, H. Mills, M. Schraefel, G. Smith, D. De Roure. Less is More: Lightweight Ontologies and User Interfaces for Smart Labs. *UK e-Science All Hands Meeting*, Nottingham, 2004.

[44] D.K.Y. Yau, J.C.S Lui, F. Liang, and Y. Yam. Defending Against Distributed Denial-of-Service Attacks With Max-Min Fair Server-Centric Router Throttles. *IEEE/ACM Transactions on Networking*, 13 (1): 29-42, February 2005.

[45] L. Tummolini, C. Castelfranchi, A. Ricci, M. Viroli, A. Omicini. Exhibitionists and Voyeurs do it better: A Shared Environment Approach for Flexible Coordination with Tacit Messages. *Environments for MultiAgent Systems*. LNAI 3374, Springer-Verlag, January 2005.

[46] F. Zambonelli, M.P. Gleizes, M. Mamei, R. Tolksdorf. Spray Computers: Explorations in Self Organization. *Journal of Pervasive and Mobile Computing*, 1 (1), May 2005.

[47] S. Kandula, D. Katabi, M. Jacob, and A. Berger. Botz-4-Sale: Surviving Organized DDoS Attacks that Mimic Flash Crowds. *Proc. 2nd USENIX Symposium on Networked Systems Design and Implementation (NSDI '05)*, Boston, MA, USA, May 2-4, 2005.

[48] J. Mirkovic and P. Reiher. D-WARD: A Source-End Defense against Flooding Denial-of-Service Attacks. *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 3, pp. 216-232, July-September, 2005.

[49] P. Bouquet, L. Serafini, S. Zanobini. Peer-to-Peer Semantic Coordination. *Journal of Web Semantics*, 2 (1), 2005.

[50] E. Gelenbe, M. Gellman, and G. Loukas. An autonomic approach to denial of service defence. *Proc. of the IEEE Int. Symp. on a World of Wireless, Mobile and Multimedia Networks*, 537-541, June 2005.