Traffic Management
A Multi-Agent System Approach

Paulo Monteiro, Luís Correia[1]

Alcatel Telecom – SRD Cascais
Portugal
E-mails: Paulo.Monteiro@srd.alcatel.pt, lc@di.fct.unl.pt

Abstract. This paper proposes an approach to the management of traffic in telecommunication networks. It was motivated by the growing need for intelligent network management in current telecommunication networks such as PSTN, ISDN or transmission networks. The goal is to constantly act upon the network in order to control the traffic flow, trying to maximize the number of calls that reach their destination or simply to reduce the congestion level of the network. In this approach, the traffic control task is based on a Multi-Agent System with learning capability. This control system depends on other peripheral systems, which supply processed traffic information (mainly performance alarms), interact with the user and forward traffic controls to the corresponding Network Elements. Each agent is dedicated to the control of a single Network Element. The whole structure of the Multi-Agent System will control the communication network. In order to provide adaptation and evolution of the control strategies each agent has on-line learning capability with an algorithm combining concepts from Cobweb and Classit.

1 Introduction

The subject of this article is closely related to one of the services of the Telecommunication Management Networks (TMN) known as Traffic Management. The TMN management services are specified in a series of recommendations produced by an international organisation called International Telecommunications Union (ITU) (previously CCITT). The main goal of ITU is the standardisation of a set of services in the area of telecommunication networks (TNs).

The Traffic Management (TM) service is devoted to the traffic management of circuit switched networks such as Public Switched Telephone Network (PSTN), Integrated Services Digital Network (ISDN) and transmission networks such as Synchronous Digital Hierarchy (SDH). The main purpose of this service is to maximise the number of successful calls across the network (NW) (and/or to maximise revenues, e.g. increase the number of expensive international/national calls sacrificing local calls). This goal is achieved through the maximisation of the usage of all-available equipment and facilities under any condition.

The TM service is also seen as a NW performance monitor, allowing whenever necessary, the possibility to take appropriate actions in order to control the traffic flow, in such a way that the overall NW performance is enhanced [E411-IT92]. In a first attempt and to assure an easier service specification, the TM service is oriented towards the Network Element (NE), here assumed to be the digital exchange.

The TMN Operations System (OS) collects traffic information from the NEs, analyses it and sends them commands to change their operation or to change the NW configuration. The NEs may report the traffic information on a periodic basis or triggered by some threshold value. The TMN OS may change those threshold values and/or the periodicity of the traffic information reports. The data delivered by the NEs may be processed within the TMN, via Mediation Devices or by the OS. The OS must collect all the relevant TM indicators in order to get an overall NW status information.

---

[1] CENTRIA - Departamento de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Portugal.

Figure 1 shows the main flow of data and control between the monitored NEs and the corresponding OS. The unit we are focusing in this work is the rectangular box labelled "Intelligent Traffic Control".

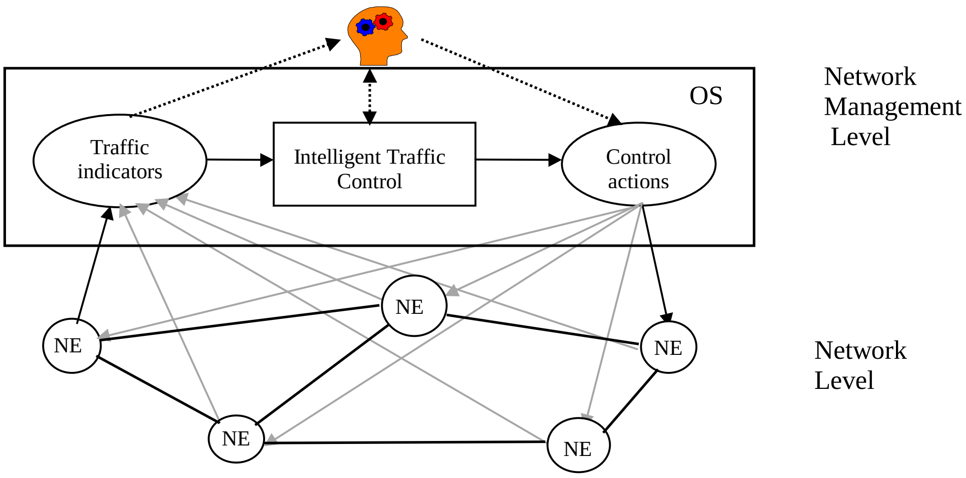

Figure 1 Basic information/control cycle

The OS may influence the NW by issuing traffic volume controls and traffic routing controls. Note, however, that the TM is done at NW management level and not at NW level. This is an important issue, since the NW may be heterogeneous, i.e. with NEs from different vendors. There are other traffic control techniques, which try to deal with some of the problems described in this paper, but which are solved at NW level. For instance some NEs support automatic congestion controls, dynamic call routing [OB96] or allow different adaptive routing algorithms [CD97]. The dynamic TM described in this paper is specially useful in adverse traffic conditions, e.g. mass call situations caused by natural disasters, public events, etc., since these kind of problems need a quick response time from the NW management operators. This system may be proactive or reactive in the TM, however the main focus will be put in the reactive TM.

2 Control strategy using Multi-Agents

The main motivation for using a Multi-Agent System (MAS) is the great complexity of the resources being managed, i.e. the NW, its distributed nature and the amount of information to be processed and correlated. The control task aims to achieve a balance between two contradictory goals: the maximisation of the overall NW performance, e.g. increase the number of successfully handled calls and the minimisation of the congestion level which reduces the performance degradation, e.g. reduce the traffic volume.

In NWs with a large number of NEs, a solution based on a single global agent becomes too complex to be handled. Since the NW traffic may only be controlled at NW nodes, it seems natural to provide each node with a Control Agent (CA). Furthermore, this association enables the CA to adapt itself to the specificity of the managed NE, for instance the NE type (transit, local, international, etc.) or the NE vendor (which may impact on the NW management controls available). Another advantage of this architecture is the use of dedicated Data Communication Network (DCN) links (see section 5) which protects the MAS from congestion problems in the NW and/or NE to NE link problems[2]. This contrasts with architectures where agents are located in the managed resources and are subject to their availability.

The MAS approach has also the following general advantages:

- It is easier to design and implement a control agent which is concerned only with a single NE and its neighbourhood;
- The distributed control architecture is more robust, in the sense that a failure in one control agent does not compromise seriously the overall NW control;
- The NW topology known by each Control Agent is reduced (synchronised with the central database in the host system).

The main disadvantage of the MAS approach to the traffic control problem is that the overall behaviour of the system is difficult to understand due to the multiple interactions between agents. On the other hand, each control agent does not have the full picture of the state of the NW and the control knowledge is distributed and replicated among the several CAs.

Each CA must take decisions that affect its own NE and in many cases also affect the surrounding NW (sub-network view). Quite often the control agent needs to exchange status information with the neighbourhood. Each control agent takes this status information into account on its control decision process.

The main function of a CA is to answer in a suitable manner to each traffic event occurring in the NW. Traffic event means, for instance, the crossing of a threshold value for some traffic indicator. The CA response to a traffic event is either to deliver a control command to the corresponding NE, to advise the human operator about an appropriate control action, or to signal other CAs about some problem which needs to be handled elsewhere.

It is important to note that there is not an explicit overall goal task to be fulfilled by the MAS. The global control task is an implicit task, which is accomplished by the individual actions of each CA on the corresponding NEs. In other MAS several co-ordination techniques are used, for instance the Partial Goal Planning (PGP) [DE88]. In this MAS the co-ordination is achieved through the use of messages, which basically carry state information about each of the corresponding NEs (see section 6).

The solution of a MAS based on reactive agents provides a clear and simple mapping of the agents to the NEs. The problem of managing traffic in TNs is well suited to be handled with approaches needing only local knowledge [CD97], which will be propagated through agent communication. However, unlike other approaches, the agents in our approach are static (or sedentary).

3 Control agent architecture

The control agent model corresponds to the reactive agent model [RN95, DY94]. The basic idea of a reactive system is to avoid task planning, using the current situation as a clue to decide the next action to take [RK91]. The control agent has the following information structure:

- Set of basic rules to guide the internal behaviour of the agent,
- Set of domain rules describing the agent control task (dynamic set),
- Partial domain model (sub-network model and control hierarchy model)
- State information

---

[2] Only when the DCN is dedicated to the management tasks, i.e. it does not use the public ISDN.

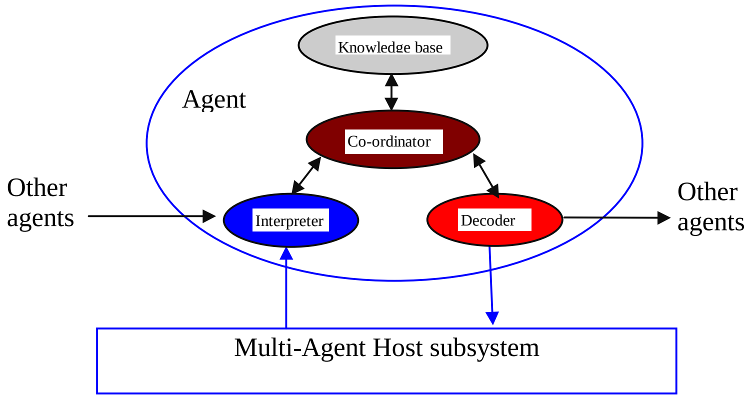

Figure 2 Model of the agent

The agent has four main components: the Interpreter, the Knowledge base, the Decoder and the Co-ordinator. The Interpreter component receives input from the environment that must be analysed and combined with the agent state information. The Interpreter builds the scenario that is then used to trigger the agent's reaction against the environment. It contains the agent's partial NW model, i.e. its corresponding NE and related NW (monitored, i.e. supervised) entities. This model is updated whenever the agent detects any inconsistency or through a notification from the host system. The Knowledge base component uses the information given by the Interpreter to decide which action to take next. This decision is taken based on the set of domain rules contained in the Knowledge base. The Decoder component is responsible for the execution of the actions towards the environment. It contains the control hierarchy model. Finally the Co-ordinator component co-ordinates the agent internal tasks and behaviour. This component also keeps track of the surrounding agents in order to exchange NW state information with them.

4 Learning behaviour

The CA supports two learning paradigms at the same time: the supervised learning and the unsupervised learning. The supervised learning is used at the NW management level through interaction with the system operators or by previous agent training. To classify NW traffic events, such as performance alarms, CA uses the unsupervised learning. For this purpose an algorithm for incremental concept formation is used: COBWEB3/C, developed based on the Classit algorithm as described in [CJ90] and also based on the COBWEB/3 lisp implementation as described in [TM91].

Each event is decomposed into a set of attribute/value pairs and it is classified into an existing concept or serves as basis for a new concept, according to an evaluation function (similar to Fisher's evaluation function).

A concept attribute may be nominal or continuous, i.e. it may represent an enumerated characteristic like an alarm severity or a real value like traffic intensity. A concept is represented directly in terms of its nominal and continuous attributes. An attribute is represented by a normal probability distribution if it is a continuous attribute or a probability value for each of its values if it is a nominal attribute. Probability values are updated every time the algorithm processes a new instance. COBWEB3/C organises concepts in a hierarchy in the same manner as the COBWEB algorithm [CJ90]. The most generic concepts represent many event instances and are located near the top of the tree, whereas the more specific concepts are located further down in the hierarchy. However, unlike the later, our algorithm may halt at some intermediate level if the new instance is similar enough to the generic concept. For that purpose a *cut-off* parameter is used in the evaluation function (as defined in the Classit algorithm).

In COBWEB3/C, classes are created based on similarities and differences of the attribute values that compose instances. The heuristic function that decides in which class a given instance belongs to tries to reach the following two goals:

- to maximise the similarities between the instances that compose a concept and
- to maximise the difference between concepts.

The COBWEB3/C evaluation function tries to weight these two goals using the intra-class attribute probabilities (IAP) and the overall attribute probabilities (OAP) [CJ90]:

- IAP = $\sum_{i}^{I_1} 1/\left(\sigma_{ik}^2 \sqrt{\pi}\right) + \sum_{i}^{I_2} \sum_{j}^{J} p\left(A_i = V_{ij} \mid C_k\right)^2$,
- OAP= $\sum_{i}^{I_1} 1/\left(\sigma_{i}^2 \sqrt{\pi}\right) + \sum_{i}^{I_2} \sum_{j}^{J} p\left(A_i = V_{ij}\right)^2$,

where C_k is the class *k*, Ai is the attribute *i*, V_{ij} is the value *j* of attribute *i*, I_1 is the number of continuous value attributes, I_2 is the number of discrete attributes, J is the number of discrete values within an attribute, K the number of classes considered at the insertion level, σ_{ik} is the standard deviation of continuous attribute *i* in class *k* and σ_i is the standard deviation of attribute *i*.

To complete the evaluation function the IAP value is summed across all *K* classes and compared to the OAP value. Then the overall result is normalised by the number of attributes and number of classes:

$$\left(\sum_{k}^{K} p(C_k) \left[IAP_k / (I_1 + I_2) \right] - OAP / (I_1 + I_2) \right) / K$$

The use of the incremental concept formation in this problem offers some advantages over other traditional learning techniques such as the neural networks [HS94] or even hierarchical algorithms like ID3 [CH94] or C4.5 [QJ93, ET94]:

- incremental learning - the CA is capable of continuous adaptation to new event types since the concept hierarchy is modified by each new instance observed;
- generalisation - due to its concept hierarchy where general concepts are higher in the hierarchy, some events may fall in general cases;
- specialisation - for the same reason, it is always possible to expand the concept hierarchy to accommodate different instances;
- handle incomplete instances, i.e. instances with unknown attribute values.

The most important requirement is to equip the agent with the capability to continuously adapt itself to new problem classes. This means that new situations will be correctly identified as such and the CA may alert the user to provide the necessary input. This is another important aspect of the agent’s behaviour (dialogue with the user), allowing the agent to learn the high-level control rules.

The supervised learning part of the agent deals with control actions or other actions such as agent-to-agent messages. These actions are stored in the agent’s knowledge base as a set of inference rules whose pre-conditions are the concepts created by the unsupervised learning algorithm (COBWEB3/C). Left for further development, is the unsupervised learning derived from the feedback of the agent’s actions received from the NW. The following questions must be addressed in order to solve this problem:

- How does an agent perceive its own actions in the NW? How does one agent really know that its actions (alone) rectified the problem?

• If a problem is not rectified, how does an agent react to this and how does it update its knowledge base? For instance a reinforcement mechanism could be applied to reward successful actions or to punish unsuccessful actions.

The overall picture of the agent's inference structure is the shown in Figure 3, similar to the *Heuristic classification* model described in [TH93].

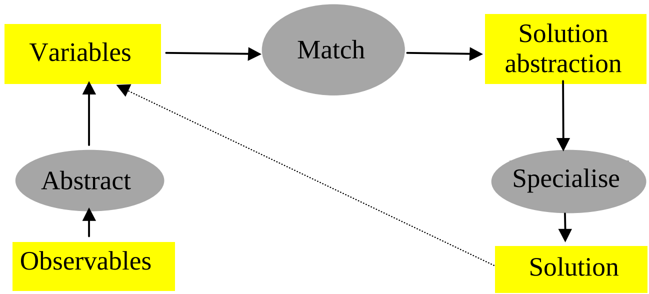

Figure 3 Agent's inference structure

The *Observables* are observable phenomenon, e.g. traffic performance alarm. The *Variables* are the value placed on the observed data, from the system's perspective. The *Solution abstraction* is the abstract classification of a problem (or concept). The *Solution* is the specific identified solution, e.g. a traffic control. The last step is to update the system's internal view of the modifications executed in NW by the CA: configuration or traffic control actions executed by the control agent. This information is merged with the abstracted data received from the NW to produce the *Variables* set.

The *Abstract* is the process of abstracting observed data into variables. This task is done by the Interpreter component (see Figure 2). The *Match* is the process of matching heuristically the variables with a concept in a classification hierarchy as described above (COBWEB3/C). This task is done by the Knowledge base component. The S*pecialise* is the process of redefining solution abstractions into more specific solutions. The concept found using the *Match* step is used to trigger a specific rule. Currently rules of the type C => A_1,..,A_n are supported, where C is a concept and A_i is a specific action, e.g. traffic control. Every action in a rule is applied sequentially when the rule is triggered. This task is done by the Knowledge base and Decoder components.

In COBWEB3/C the most sensible parameter is the *cut-off* parameter. The algorithm uses this parameter to decide when an existing node is "good enough" to classify the new instance. When the parameter is too low then the concept hierarchy tends to grow with an excessive specialisation. On the other hand if the parameter is too high then the algorithm tends to build a shorter concept hierarchy with an excessive generalisation. In order to facilitate the task of parameter tuning, the CA is constantly evaluating the results of its own suggestions to the user and adjusts the parameter accordingly. The fluctuation of the *cut-off* parameter will diminish has the CA learns the correct rules and the concept hierarchy stabilises.

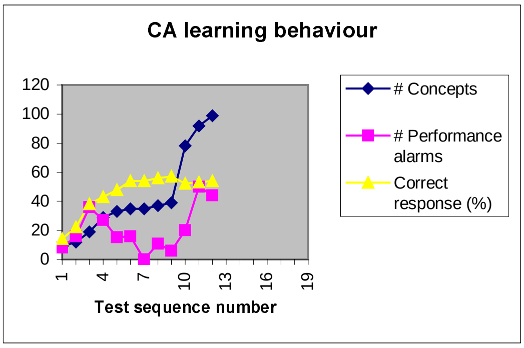

Figure 4 Learning behaviour

In Figure 4 one can see the evolution of the CA response to incoming performance alarms. It is clear that the percentage of correct responses does not increase when new concepts are being created. In this example the CA is still learning since the number of new concepts is still increasing.

5 Multi-Agent system

The Multi-Agent System (MAS) is composed of the Multi-Agent Host System (MAHS) and by the set of Control Agents (CAs). In the subsequent text, the term MAHS will be used to identify the part of the OS that is exclusively dedicated to the CA management.

Figure 5 illustrates the interaction between the CAs and the MAHS and between the MHAS and the peripheral subsystems:

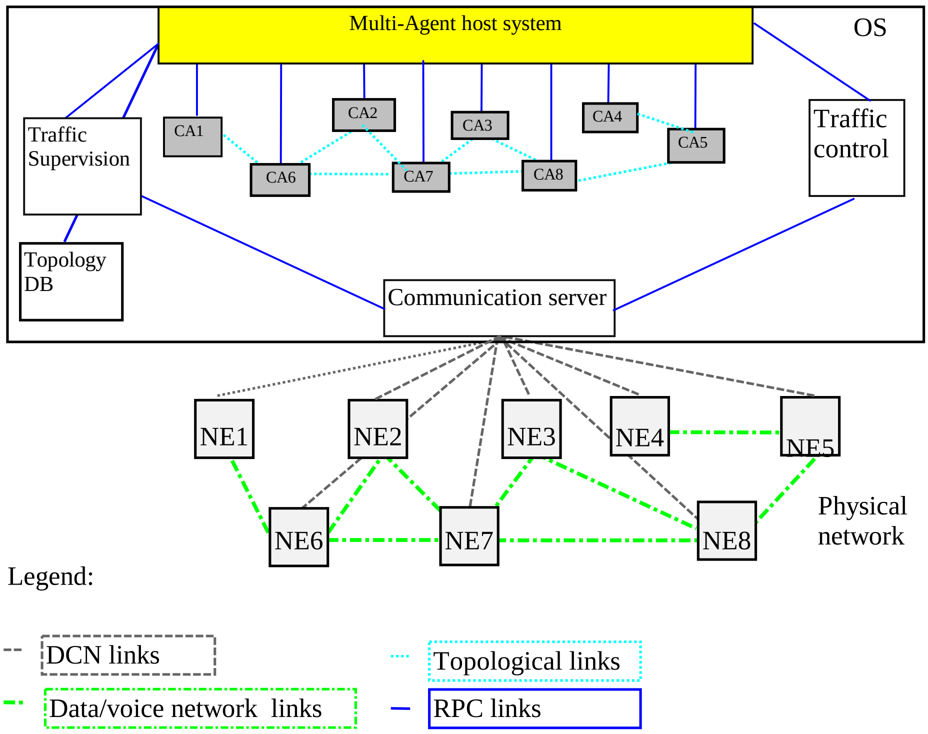

Figure 5 Multi-Agent environment

The main purpose of the MAHS is to provide an environment in which the control agents may run and communicate between each other and with other peripheral systems. This means that the MAHS is the main interface for control agents, with both the user and the other systems. For instance, the user can launch a new control agent for a new NE or kill an existing agent for a NE that does not exist any more. The MAHS is also responsible for notifying the control agents if any change in the NW status or topology occurs.

The first external interface of the MAHS is the graphical user interface. This interface is used to exchange data between the user and the MAHS and between each control agent and the user. The user is able to visualise the whole set of control agents and display detailed information about its current activities. The agent may ask or suggest any response action to the operator for any traffic problem, thus learning from the operator expertise.

The second external interface of the MAHS is the interface to a topology database, which is used to keep the control agents up-to date about the NW topology (e.g. NE relations and contained NW objects).

The third external interface of the MAHS is the interface to the Traffic Supervision system. Through this interface the host system receives the NW status data, basically performance alarms and/or traffic indicators, and forwards it to the control agent responsible for the corresponding NE.

The fourth external interface of the MAHS is the interface to the Traffic Control system. This interface enables the control agents to send their NW management controls to the corresponding NEs.

The MAHS defines the communication paths between CAs according to the topological links between the NEs. Each CA is allowed to communicate with all of its connecting neighbours. The reason for this kind of connectivity between CAs is to enable the communication between agents, which may work together on a global control strategy.

The MAS is made up of a set of Unix processes that may run in a distributed environment. The communication between the CAs is based on SNAP© RPCs through the MAHS process.

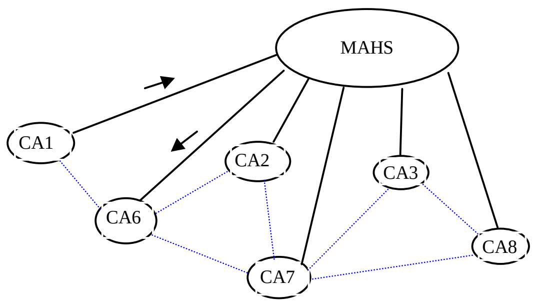

Figure 6 Inter-agent communication model

The MAHS contains an internal model of the MAS and their topological connections. For instance, in Figure 6 the CA1 may send messages to agent CA6 but is forbidden by the MAHS to communicate with the other CAs. This model is simpler from the implementation point of view and allows the above described agent connectivity (indicated by the dashed lines).

In the OS described in Figure 5, the NW topology data, the Traffic Supervision subsystem and the Traffic Control subsystem are centralised in the system. On the other hand, the MAS is fully distributed. This could lead to a performance bottleneck in the interfaces Traffic Supervision/MAHS and MAHS/Traffic Control subsystems. However, this is acceptable in this system for the following reasons:

- Most of the performance alarms are generated in the Traffic Supervision system, thus part of the traffic indicators which do not give any relevant information are not injected in the MAHS;

• Even if the MAS processes many alarms simultaneously, part of them will produce no traffic control actions towards the Traffic Control subsystem. The traffic control at this level is not intended to be an intensive task like the dynamic routing management, even thought some traffic controls affect traffic routing.

• The user needs to interact with the MAS, thus communication CA/user is mediated by the MAHS.

In the future, the MAS could interact with several MAHSs, which in turn could have interfaces to multiple Traffic Supervision subsystems and Traffic Control subsystems. This would provide further distribution and processing power in the OS to manage any number of NEs without changing the overall MAS architecture, and each CA would remain unchanged.

6 Case study

To better understand how the MAS achieves the NW traffic control task, let's take a look at two short scenarios. Assume the MAS and the NW topology as depicted in Figure 5. The examples shown in this section are based in a prototype system named CODITRA (see also MP98).

Usually the Traffic Supervision system is monitoring the whole NW using the traffic indicators received periodically from the NW. In this system it is possible to define threshold values to certain combination of traffic indicator values, in order to produce traffic performance alarms. These performance alarms are delivered to MAHS, which forwards them to the corresponding CA. For instance, alarms derived from traffic indicators originated in NE1 will be delivered to CA1, in NE2 to CA2, and so on. As described in section 4, the received performance alarms and associated set of parameters constitute the CA *Observables.* These are then analysed by the CA and mapped into internal variables (set of attribute/value pairs). The combination of the incoming (abstracted) data with the CA internal view of the NW state produces what has been described as the *Variables.*

The *Variables* set are matched by the COBWEB/3C into a concept that may or may not already exist. If the CA has been already trained then it is likely that the concept already exists. Therefore an inference rule is triggered by that concept. The rule determines what are the next activities of the CA, including the possibility to apply or remove NW management controls, propagate the alarm, deliver a message to the user, etc. If the concept is new then this means that the situation is new and that the agent requires attention from the user. The user has then the opportunity to create a new rule by selecting a set of actions to produce a new inference rule.

6.1 Traffic Congestion at destination (Mass calling situation)

In this scenario the Traffic Supervision system is monitoring a set of traffic destinations which may be the target of many calls simultaneously, for instance the number of a TV show. MAHS receives the traffic performance alarm for NE8 and destination 015473594 produced by the Traffic supervision system. The alarm has the following contents: {Severity = critical, Origin = Supervision, Parameters = (ASR=5%, OCC=80)}. MAHS forwards the alarm to CA8, which classifies the problem into class C_0 (Normal NE congestion state, normal sub-network congestion state and critical congestion on destination).

The knowledge base of CA8 has previously learned from the operator that this kind of situations may be handled (in that particular NE version) with a traffic volume control applied to the destination, e.g. a Call Gapping control. However, given this kind of problem CA8 has also learnt that alarm propagation to the neighbour CAs may also help to reduce the traffic near the source. So the agent CA8 will respond to the problem with two actions: a traffic volume control to the NE8 (delivered through the Traffic control subsystem) and a new internal alarm delivered to the CAs corresponding to the NEs at the traffic source side: CA7, CA3 and CA5. Then, each CA that receives the new alarm needs to decide if it should apply the same strategy or not. If the NE is a transit NE then maybe the CA decision is to forward the alarm further back to the traffic source without applying any traffic control. The OS operator tailors this behaviour for each CA during the initial training. The traffic volume controls applied in this scenario will be removed as soon as the situation is back to normal; i.e. the alarm is cleared.

6.2 Traffic congestion at NE links

In this scenario the Traffic Supervision system is monitoring the traffic in the links (Circuit subgroups) between the NEs. MAHS receives a traffic performance alarm for NE7 and circuit subgroup O78 (outgoing link from NE7 to NE8). The alarm has the following contents: {Severity = major, Origin = Supervision, Parameters = (OCC=86%)}. MAHS forwards the alarm to CA7 that classifies the problem into class C_1 (normal NE congestion state, normal sub-network congestion state, major congestion on outgoing communication link).

The knowledge base of CA7 has previously learned from the operator that this kind of problem may be solved by making available new circuit sub-groups which are normally idle or by making both-way circuit subgroups one-way circuit subgroups (if the traffic in opposite direction is below average). Therefore, the CA7 reaction to the problem is to select an alternative circuit sub-group (assuming that one exists, e.g. circuit subgroup from NE7 to NE3) and then apply the traffic route control TAR (Temporary Alternative Routing) in the NE7 (delivered trough the Traffic Control subsystem). The traffic route control applied in this scenario will be removed as soon as the traffic volume goes back to its normal value; i.e. the alarm is cleared.

7 Discussion

It was found that in some particular cases (complex situations) the inference step *Specialize* described in section 4 could be improved to allow a higher focus on the right traffic control to be applied. This is an important issue since the right response may depend not only on the current traffic alarm but also on specific information extracted from other traffic alarms, topological information or even external events. For instance, in the scenario 6.2, the CA7 must decide the right traffic control to apply, but the right control depends on a variety of constraints: are there any alternative routes available, is the traffic in the opposite direction low, are all the current alternate routes already congested? All these questions could be transformed in pre-conditions of the CA rules for that same situation, making the control much more focused. These restrictions could be handled by evolving from a simple reactive agent to an intentional agent [OJ96] where the agent could learn to revise its control strategy according to the above conditions.

In general the CA behaves well when classifying alarms which are similar enough (same object type, same severity, similar parameters). This allows the adequate use of the high-level control rules defined by the operator. The COBWEB/3C shows also some sensibility to the *cut-off* parameter, where an excessive value for this parameter leads to a smaller tree, but with highly inaccurate rules being fired. On the contrary, setting the *cut-off* parameter to a very low value will lead to a more accurate classification process but with a poor supervised learning rate.

8 Conclusions and Future work

This paper proposes an alternate/complementary approach for the management of traffic in TNs. Due to the TN complexity, it is necessary to provide the NW operators with systems (OSs) which provide full NW management at its different functional areas: fault, configuration, performance, security, billing, etc. The TM is related with the Service and NW management layers applied to the Configuration and Performance areas.

Despite the use of several routing algorithms and congestion control mechanisms at NW level, there is the need to provide the operator with a centralised TM system, which enable the monitoring/control of the NW at a single point. Having solved many of the problems related to the traffic data collection, analysis and storage, the issue now is how this information may be used in an effective way to maximise the NW resource usage. For instance, to find spots in the NW where resources are used over the limit (congestion) and in other parts resources are being wasted. This service is specially focused to handle abnormal NW conditions derived from a variety of reasons: public events, natural disasters or other events that affect the normal traffic distribution (see also section 1).

This Multi-Agent system works embedded in a TM application. As explained in section 2, the motivation behind this work is to develop a system that is capable of dealing with a heterogeneous NW, providing an automatic or semi-automatic traffic monitoring and control. Since the traffic control is targeted at each NE, the MAS contains a set of CAs with communication paths that follow the NW topology. This CA disposition was chosen to reduce the communication overhead between CAs that are not connected by a common traffic path.

The main feature of the CA is the *Heuristic classification* of traffic performance alarms as described in section 4. The supervised learning is achieved through the CA interaction with the human operator (indirectly through the MAHS, see section 5) or by off-line training. The learning capabilities of the MAS provide a very flexible control system, since each CA can adapt itself to the specificity of its own NE or to the traffic characteristics in the NE.

Another important feature of the CA is its ability to improve its knowledge by continuously adapting its classification hierarchy to new incoming data and by interaction with the operator. At the beginning the operator has to provide most of the high level control strategy, but after some time the CA starts to become independent, requiring little attention from the operator.

As discussed in section 4, the MAS could benefit from the NW feedback to validate its own control actions. This is still a hard problem since the NW traffic control is a stochastic distributed multi-objective problem [CD97].

It was found that, despite the control flexibility provided by the MAS, in some specific problems the CA might have responses that are not focused enough. This is the case in some traffic routing controls where some extra inference steps could provide a better solution (see section 7). This issue will be the subject of future improvements of the CA inference structure.

Another issue to be further explored is the CA co-operation in the Traffic Control tasks. In what measure would a stronger co-operative model (using techniques such as the PGP [DE88]) improve a global control strategy? In this system, the CAs already exchange state and alarm messages between them. Through this mechanism one agent induces other agents to react on a problem that is affecting its NE, but in a more effective way. For instance, the congestion on a destination may be handled further back near the source of the traffic, making the control much more efficient.

Finally the MAS could benefit from the mobile agent approach in order to reduce the communication overhead with the support peripheral systems. For instance it may use mobile agents that are sent to several sites in search of external event information that may be used in correlation with the current performance alarms. An agent may find out that at the same time there was an equipment fault in some NE or even an unexpected disaster in the NE neighbourhood. This is especially useful for the diagnosis of a problem, but might as well make a difference in the control strategy used by an agent. For the application of some controls quite a lot of topological information is needed, e.g. the Temporary Alternate Route, therefore specialised mobile agents could have tasks like “find the best alternate route for this route”, which would take into account alternate route availability, price, congestion state, etc.

9 Bibliography

CD97 Caro, Gianni Di, Dorigo, Marco (1997). AntNet: A mobile Agents Approach to Adaptive Routing. IRIDIA, Université Libre de Brussels, Belgium, Technical Report IRIDIA 97-12.

CH94 Cohen, William W., Hirsh, Haym (1994). Machine Learning, Proceedings of the Eleventh International Conference, July 10th-13th, Rutgers University, New Brunswick, New Jersey.

CJ90 Carbonell, Jaime (1990). Machine Learning, Paradigms and Methods, edited by Jaime Carbonell, The MIT Press, Massachusetts, USA.

DE88 Durfee, Edmund H. (1988). Coordination of Distributed Problem Solvers, University of Massachusetts at Amherst, Kluwer Academic Publishers, Boston/Dordrecht/London.

DY94 Demazeau, Yves (1994). "I - Cognitive Agent Models. II - The Reactive Agent Paradigm" *in Artificial Intelligence for multi-agent systems: methodologies and applications, Advanced school on artificial intelligence´94, Vol II, Troia-Setubal, Portugal.*

E411-IT92 ITU (1992). E.411 - Telephone Network and ISDN Quality of Service, Network Management and Traffic Engineering - International Network Management - Operational Guidance, 10/92.

ET94 Elomaa, Tapio (1994). *"*In defense of C4.5: notes on learning one-level decision trees*" in* Machine learning, Proceedings of the eleventh international conference, July 10th-13th, Rutgers University, New Brunswick, New Jersey, pp. 62-69.

HS94 Haykin, Simon (1994). Neural Networks, A comprehensive foundation, Simon Haykin, Macmillan College Publishing Company, Inc, USA.

MP98 Monteiro, Paulo (1998). Controlo dinâmico de Tráfego em Redes de Telecomunicações, Dissertação de Tese de Mestrado, Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia, Lisboa.

OB96 Offers, Bert (1996). "A new role for automated traffic management" *in* Telephony, November 18, 1996:52-60.

OJ96 O´Hare, G.M.P. and Jennings N.R. (1996). Foundations of Distributed Artificial Intelligence, A Wiley-Interscience Publication, John Wiley & Sons, Inc., New York.

QJ93 Quinlan, J. R. (1993). C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, Inc., San Mateo, CA.

RK91 Rich, Elaine; Knight, Kevin (1991). Artificial Intelligence, Second Edition, MacGraw-Hill, Inc.

RN95 Russel, Stuart, Norvig, Peter (1995). Artificial Intelligence - A modern approach, Prentice Hall, Upper Saddle River, New Jersey 07458.

TH93 Tansley, D.S.W., Hayball, C.C. (1993). Knowledge-based systems analysis and design - A KADS Developer´s Handbook, Prentice Hall, UK.

TM91 Thompson, Kevin; McKusick, Kathleen (1991). Cobweb/3: A Portable Implementation, Kevin Thompson, Kathleen McKusick, Sterling Software/AI Research Branch, NASA Ames Research Center, Mail Stop 269-2, Moffet Field, CA 94035 USA, November 3, 1991.