Learning-Based Call Admission Control Framework for QoS Management in Heterogeneous Networks

Abul Bashar[1], Gerard Parr[1], Sally McClean[1], Bryan Scotney[1], and Detlef Nauck[2]

[1] School of Computing and Engineering, University of Ulster, Coleraine BT52 1SA, UK
{bashar-a,gp.parr,si.mcclean,bw.scotney}@ulster.ac.uk
[2] Research and Technology, British Telecom, Adastral Park, Ipswich IP5 3RE, UK
detlef.nauck@bt.com

Abstract. This paper presents a novel framework for Quality of Service (QoS) management based on the supervised learning approach, Bayesian Belief Networks (BBNs). Apart from proposing the conceptual framework, it provides solution to the problem of Call Admission Control (CAC) in the converged IP-based Next Generation Network (NGN). A detailed description of the modelling procedure and the mathematical underpinning is presented to demonstrate the applicability of our approach. Finally, the theoretical claims have been substantiated through simulations and comparative results are provided as a proof of concept.

Keywords: Quality of Service (QoS), Call Admission Control (CAC), Bayesian Belief Networks (BBNs), Next Generation Network (NGN).

1 Introduction

In a telecommunication network, the Network Management System (NMS) plays a key role in maintaining the network infrastructure, assuring smooth running of services, controlling the operational costs of the network and providing increased revenue to the service provider. As networks evolve in terms of their architecture, variety of services and application demands, the functions of NMS are constantly under pressure to perform efficiently. In recent times, the ability of providing quadruple services (voice, video, data and mobility) by bringing together the fixed and mobile network infrastructure and creation of novel applications has led to the emergence of the Next Generation Network (NGN). In essence, NGN is an IP-based packet switching dedicated network with special emphasis on guaranteed QoS support to multimedia services. The International Telecommunication Union (ITU-T) has defined the NGN to be "a packet-based network able to provide telecommunication services and able to make use of multiple broadband, QoS-enabled transport technologies and in which service related functions are independent from underlying transport-related technologies"[1]. The key challenges of managing such a network have been aptly summarized in [2] and they form the basis of our motivation to propose a novel solution.

Call Admission Control (CAC) is a most effective technique to provide the required QoS by limiting the traffic into the network [3]. However, it has to deal with

F. Zavoral et al. (Eds.): NDT 2010, Part II, CCIS 88, pp. 99–111, 2010.

two conflicting objectives of maximizing resource utilisation and minimising traffic congestion. Decision-making is a key function of any CAC technique, where it has to decide whether a new call be admitted into the network or blocked from entry, keeping in view the required QoS criteria. It is to be emphasised here that the QoS of both the ongoing calls and the new (or requested) call should be taken into account, while making the decision. We believe that apart from the existing CAC schemes, there is a need for another class of scheme, which we term as learning-based CAC. The major driving forces for this proposal are – the utilisation of network measurement data and real-time decision making. The first objective is met by harnessing the huge volumes of network management data collected by network management protocols (e.g. SNMP). The second objective is met by using machine learning techniques which learn from the available data and also provide intelligent decisions which improve with time and experience. As such we propose the CAC framework based on the Bayesian learning technique for QoS optimization in heterogeneous networks.

The remainder of this paper is structured as follows. In Section 2 we provide a background and related work in this research domain. Section 3 presents the conceptual architecture developed for the study. Section 4 emphasizes the capabilities of Bayesian Belief Networks (BBNs) in terms of managing dynamic systems. Section 5 presents a case study and comparative results using the BBN to model a typical CAC scenario in a NGN. Section 6 concludes the paper by suggesting possible future work.

2 Background and Related Work

Network management systems collect the status of the network elements which are present in the network. Two major protocols which support this task are - SNMP for TCP/IP networks [4] and CMIP for OSI-based networks [5]. SNMP has been widely used for internet data collection. It uses the polling procedure to access the Management Information Base (MIB) objects present on each of the managed elements. To draw meaningful knowledge from this collected data, researchers have applied machine learning methods, which employ offline and online procedures to capture trends and identify anomalous conditions. We examine the literature related to this area in the following paragraph.

Decision trees are one such approach, which has been used to achieve proactive network management by processing the data obtained from the MIB objects of an SNMP managed network [6]. Fuzzy logic has been studied to assess its suitability for network management tasks like bandwidth brokering [7]. Bayesian Belief Networks (BBNs) have been used for providing proactive network fault detection in a telecommunication network [8]. Dynamic Bayesian Networks (DBNs) have been used to model the communication network, considering the temporal dimension which manifests itself in the case of highly dynamic networks [9][10]. Bayesian network modelling has been applied to the SNMP MIB variables to achieve network fault management, employing intelligent software agents [11]. Combined use of Bayesian Belief Networks and Genetic Algorithms for fault management in telecommunication systems suggests better performance [12].

The above literature suggests that machine learning techniques can assist in carrying out network management tasks and provide improved performance. But majority of the research concentrates on *fault* management. Therefore we now choose

Bayesian Belief networks (BBNs) to help in *performance* management by implementing the learning-based CAC.

3 Conceptual Architecture

The domain under consideration for our work is shown in Fig. 1. A typical router in the core of the IP network is used to extract the Network Management data (e.g., SNMP MIB variables).

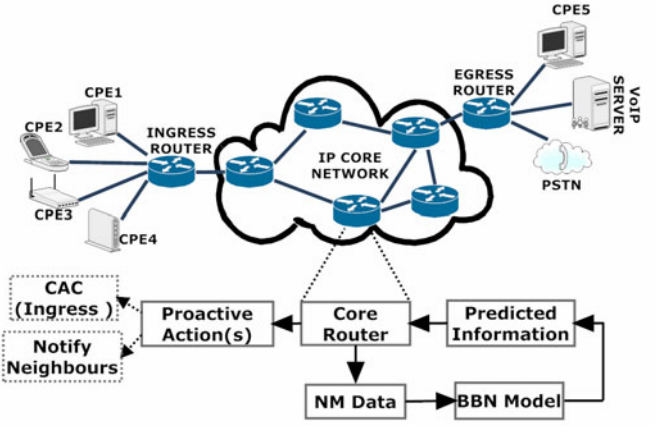

Fig. 1. Conceptual architecture

This monitored data provides status regarding the routing protocols, flow control, link utilization, QoS level, etc. In fact the volume of this collected data is huge and *mining* it is key to extracting *intelligent* information for aiding proactive management. Hence, the NM data is fed into the BBN model to extract intelligence. The procedure of building the BBN model and generating predictive information is described in Section 5. A feedback (e.g., prediction of congested link) from the BBN to the router helps to initiate proactive actions. This could mean signalling the ingress router to adjust the Call Admission Control algorithm or informing the neighbouring core routers of impending congestion. Proper execution and timing of these actions would then be instrumental in guaranteeing the QoS levels as prescribed in the SLAs, thus resulting in efficient utilization of the network resources, customer satisfaction and improved revenues. We concentrate on the CAC proactive action and describe it in detail in Section 5. The following section lays the foundation for the theory of BBN and its features which can be applied to our problem.

4 BBNs and Influence Diagrams

A BBN is a graphical structure that allows us to represent and reason about an uncertain domain [13]. It consists of nodes (representing domain random variables)

connected by directed arcs (representing causal dependencies). The nodes and arcs form a Directed Acyclic Graph (DAG). An example BBN, representing the effects (*Delay* and *Packet_Loss*) of *Buffer_Status* at a particular router in a network, is shown in Fig. 2. To quantify the strength of relationships among the random variables, conditional probability tables (CPTs) are associated with each node. For a typical node A, with parents $B_1, B_2, ...B_n$, there is attached a CPT as given by $P(A \mid B_1, B_2, ...B_n)$. For root nodes, the CPT reduces to prior probabilities.

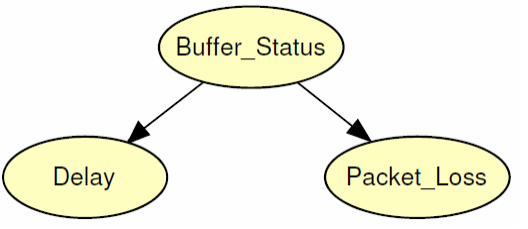

Fig. 2. An example of BBN

The main principle on which BBNs work, is Bayes' rule,

$$P(H \mid e) = \frac{P(e \mid H)P(H)}{P(e)} \tag{1}$$

where $P(H)$ is the prior belief about a hypothesis, $P(e \mid H)$ is the likelihood that evidence e results given H, and $P(H \mid e)$ is the posterior belief in the light of evidence e. This implies that belief concerning a given hypothesis is updated on observing some evidence. In response to evidence, the beliefs of all the nodes in the BBN are updated except the observed node(s). This is called belief propagation, which can be achieved by applying efficient algorithms [14]. This belief updating provides the latest status about the domain under study.

To incorporate decision making capabilities, the BBN is converted to an influence diagram (ID), by adding decision nodes and utility nodes. A decision node represents the decision being made at a particular point in time. The values taken by the decision nodes are the actions which must be chosen by the decision maker. A utility node quantifies the *usefulness* of the outcomes resulting from the actions of decisions. To achieve this, the utility node assumes a utility table with one entry for each possible instantiation of its parents. In our example, we have added a decision node (*Signal_Alarm*) which decides either to indicate *Safe* or *Danger* level of packet drops by the router. Also, a utility node (*Utility*) is added, which will have a utility table with 4 entries (indicating *usefulness* of a particular decision) representing the combinations of *Signal_Alarm* and *Packet_Loss* states. The resulting ID is shown in Fig. 3. The ID is a useful tool in quantifying our decisions based on some observations. This capability will be used for making *intelligent* decisions for CAC.

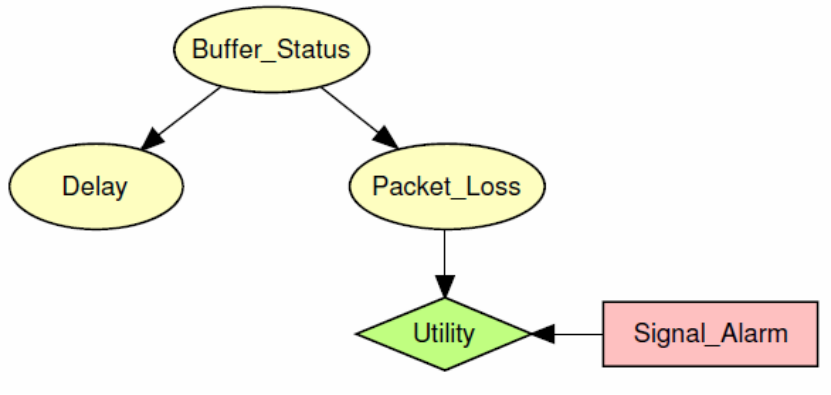

Fig. 3. Influence diagram for the BBN of Fig. 2

5 CAC Case Study

Let us assume that we are interested in managing the performance of a dynamic network (e.g. NGN), which primarily supports converged multimedia services. From the service provider's point of view, one would like to maximize the number of incoming calls into the network (which means a proportional increase in revenue). But from the management point of view, this could result in traffic congestion at some point in the network (e.g., at a router). To strike a balance between the two conflicting processes, we can make use of the probabilistic inference feature of the BBN.

5.1 Call Admission Control and Congestion Avoidance

Congestion is the major cause of network performance degradation, which negatively affects the Quality of Service (QoS) of the applications running over it. Congestion in any part of the network (typically, a router) may be caused by bursty traffic patterns, multiple sources and single destination, insufficient memory, slow processor, low bandwidth of the link or unbalanced upgrades of lines/equipment [15].Various control schemes have been developed to deal with congestion [16]. A preventative approach to deal with congestion is termed Call Admission Control (CAC) [3]. The CAC scheme deals with the question of whether to accept a new call, based on whether this new call can be supported with the desired QoS.

5.2 Simulation Setup

For the purpose of simulation and data collection, we have used OPNET Modeler [17]. A typical core router (as conceptualized in Fig. 1) was modelled by extending the generic M/M/1 FIFO queue to support multiple sources (n=7), as shown in Fig. 4. The traffic patterns of the sources were modelled as Poisson processes, with varying

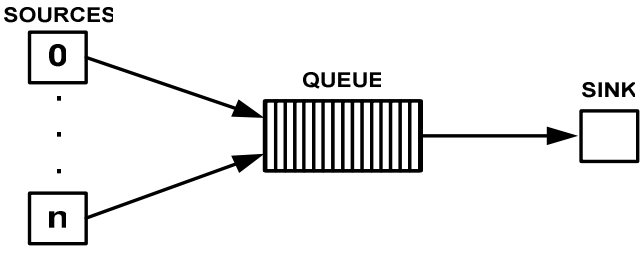

Fig. 4. A typical model of a core router

Inter Arrival Times (IAT). The queue size was fixed at 100 packets and the service rate to 2,500 bits per second. Bursty traffic (which is a feature of NGN) was simulated by choosing sources as shown in Table 1. The simulation was carried out for a network simulation time of 6 hours.

Table 1. Parameters used for a sample simulation

Source	Packet IAT	Start Time (s)	End Time (s)
0	exp(1)	10	∞
1	exp (0.1)	170	182
2	exp (0.1)	7310	7323
3	exp (0.1)	9099	9107
4	exp (0.1)	12700	12711
5	exp (0.05)	14450	14457
6	exp (0.1)	16250	16258
7	exp (0.077)	18017	18024

5.3 Building the BBN

The first step is to identify the variables of interest in the domain which is being modelled. This means choosing network statistics (nodes of BBN) which are indicative of causes and effects of congestion in a router. In our case, we derived the nodes (from OPNET data) as shown in Table 2. It must be noted that the number of values (or levels) which a node can take is a trade off between accuracy level desired by the domain modeller and computational complexity in updating the posterior probabilities in CPTs. This study assumes the nodes to be discrete having two or three values, as described in Table 2.

Table 2. BBN nodes and their values

Node Name	Type	Values
Traffic_Volume	Ordered	{*low, med, high*}
Buffer_Status	Ordered	{*low, med, high*}
Delay	Ordered	{*low, med, high*}
Packet_Loss	Ordered	{*low, med, high*}
Congestion	Boolean	{*true, false*}

One of the challenges faced was the discretisation of continuous variables which were measured from the OPNET simulations. It should be emphasized that Bayesian Networks do face some constraints when dealing with continuous variables [18]. Hence, the sophistication in discretisation has an impact on the BBN structure and the CPTs.

The next step is building the network structure using the nodes and directed arcs. The direction of the arc is from the causal node to the effect node. The dependencies among the nodes can either be known (e.g., from the domain expert) or can be learnt from the data (e.g., historic data of the domain). In this study we have learnt the structure from the historic data collected from the OPNET simulations. Fig. 5 shows the BBN structure obtained by using the structure learning feature of Hugin Lite 6.9 which uses the Necessary Path Condition (NPC) algorithm [19]. It should be mentioned here that the NPC algorithm provides an opportunity to the system modeller to incorporate domain knowledge about the variable dependencies, in the case of uncertain relationship information due to limited data.

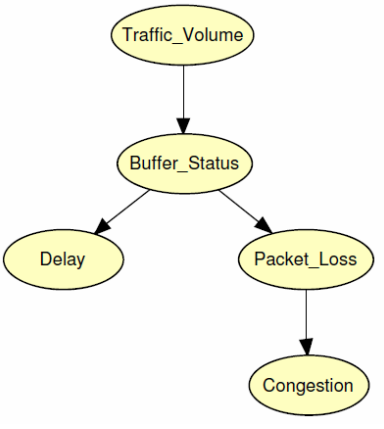

Fig. 5. BBN for congestion control

Further a quantitative measure is specified for each node by defining conditional probability tables (CPTs) (when dealing with discrete domains). The size of the CPT for a particular node is determined by the number of parents it has. The probabilities can be based on the observed historical data about the events in the past or can be provided by the Subject Matter Expert (SME). In this study the Expectation-Maximization (EM) algorithm of Hugin Lite 6.9 has been used to derive the CPT values from the historic data obtained through OPNET simulations. Table 3 shows the CPTs for each node of the BBN shown in Fig. 5. As an example, the probability of *Buffer_Status* being in low state, given that the *Traffic_Volume* is in high state, denoted by P(*Buffer_Status=Low*| *Traffic_Volume=High*)=0.769

Table 3. CPTs for the nodes of the BBN

Traffic_Volume			
Low	0.398		
Med	0.586		
High	0.016		
Buffer_Status			
Traffic_Volume	*Low*	*Med*	*High*
Low	0.001	0.007	0.769
Med	0.101	0.489	0.127
High	0.898	0.504	0.104
Delay			
Buffer_Status	*Low*	*Med*	*High*
Low	0.036	0.006	0.497
Med	0.112	0.972	0.501
High	0.852	0.022	0.002
Packet_Loss			
Buffer_Status	*Low*	*Med*	*High*
Low	0.273	0.995	0.998
Med	0.596	0.002	0.001
High	0.131	0.003	0.001
Congestion			
Packet_Loss	*Low*	*Med*	*High*
True	0.002	0.801	0.717
False	0.998	0.199	0.283

5.4 Evidence and Belief Updating

The process of observing/monitoring a node gives new information about that particular node. This evidence is used to update the belief (posterior probabilities) and propagate this new belief from the evidence nodes to the query nodes (nodes which do not have any evidence). BBNs support two types of reasoning: predictive and diagnostic reasoning. In predictive reasoning the causal nodes act as evidence nodes and the beliefs of the effect nodes are updated, whereas in diagnostic reasoning the effect nodes become the evidence nodes and the beliefs of causal nodes are updated. There are various algorithms (e.g. Kim and Pearl's Message Passing Algorithm [14], Lauritzen Speigelhalter Algorithm [20]) to carry out inference in Bayesian Networks. A combined use of both types of reasoning is very helpful in managing highly dynamic systems.

Let us look at the steps involved in belief updating for singly connected networks, such as Fig. 5. For a detailed analysis of the algorithm, the reader is referred to [14]. Consider a typical node X having m children, $Y_1,....,Y_m$, and n parents, $U_1,....,U_n$, as shown in Fig. 6. Three groups of parameters are required for distributing the belief of variable X.

1) Current causal support π from incoming links $U_i \rightarrow X$:

$$\pi_X(u_i) = P(u_i \mid e^+_{U_i X}) \tag{2}$$

where the '+' sign indicates evidence(s) observed in the ancestor nodes of X.

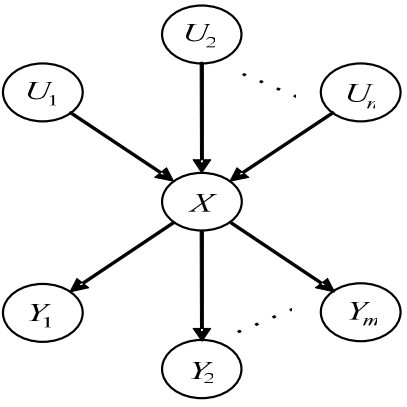

Fig. 6. A typical node X with parents and children

2) Current diagnostic support λ from outgoing links $X \to Y_j$:

$$\lambda_{Y_j}(x) = P(e^{-}_{XY_j} \mid x) \tag{3}$$

where the '-' sign indicates evidence(s) observed in the descendant nodes of X.
3) The fixed conditional probability matrix $P(x \mid u_1,....,u_n)$ relates the variable X to its parents. Using these parameters the local belief updating is done using the following three steps in any order.

Step 1 – Belief updating: The node X simultaneously inspects the π messages from its parents and λ messages from its children and updates its belief measure to

$$BEL(x) = \alpha\lambda(x)\pi(x), \tag{4}$$

where

$$\lambda(x) = \prod_j \lambda_{Y_j}(x), \tag{5}$$

$$\pi(x) = \sum_{u_1,....,u_n} P(x \mid u_1,....,u_n) \prod_i \pi_X(u_i), \tag{6}$$

and α is a normalising constant to make $\sum_x BEL(x) = 1$.

Step 2 – Bottom-up propagation: The node X computes new λ messages to send them to its parents:

$$\lambda_X(u_i) = \beta \sum_x \lambda(x) \sum_{u_k : k \neq i} P(x \mid u_1,....,u_n) \prod_{k \neq i} \pi_X(u_k) \tag{7}$$

Step 3 – Top-down propagation: The node X computes new π messages to send them to its children:

$$\pi_{Y_j}(x) = \alpha\left[\prod_{k \neq j} \lambda_{Y_k}(x)\right] \sum_{u_1,\ldots,u_n} P(x \mid u_1,\ldots,u_n) \prod_i \pi_X(u_i) \tag{8}$$

$$= \alpha \frac{BEL(x)}{\lambda_{Y_j}(x)} \tag{9}$$

5.5 Influence Diagram

Now we convert our BBN (Fig. 5) to an Influence Diagram (Fig. 7). We add a decision node (*Admission_Control*) and a utility node (*Utility*) to make a decision (whether to admit a call into the network or not) on the status of the *Congestion* node.

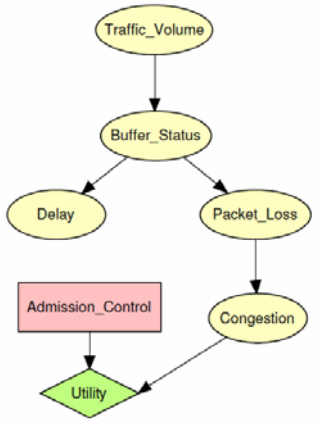

Fig. 7. Influence diagram for CAC

Table 4 shows the utility table (associated with the *Utility* node) which will be used in making the decision. The utility values (chosen on an intuitive basis in our case) indicate the decision maker's preferences with respect to a particular decision being taken in a given situation. When evidence is observed at any node, the belief updating process is performed (which uses both predictive and diagnostic supports, as described in section 5.4). Hence, the decisions made by the ID can be passed to the performance management functions to initiate proactive actions to deal with the congestion situation. This probabilistic support achieves proactive performance management by facilitating the network management functions, as shown below.

Table 4. Utility table for the ID of Fig. 7

Utility				
Admission Control	*Admit*		*Block*	
Congestion	True	False	True	False
Utility	-100	50	100	-50

Now, let us see a sample decision (based on predictive reasoning) made based on the evidence of *Traffic_Volume* node having a *high* value. This means that the router is under an influence of bursty input traffic. Now, the ID decides to not admit any further calls into the network (with a utility factor of +16.2 in deciding to *Block* a call), as depicted in Fig. 8. Note that the monitor windows beside the nodes display the updated probabilities for each state. Based on this result we conclude that BBN-based CAC can provide support to network manager to choose appropriate actions in case of congestion and implement a better CAC system.

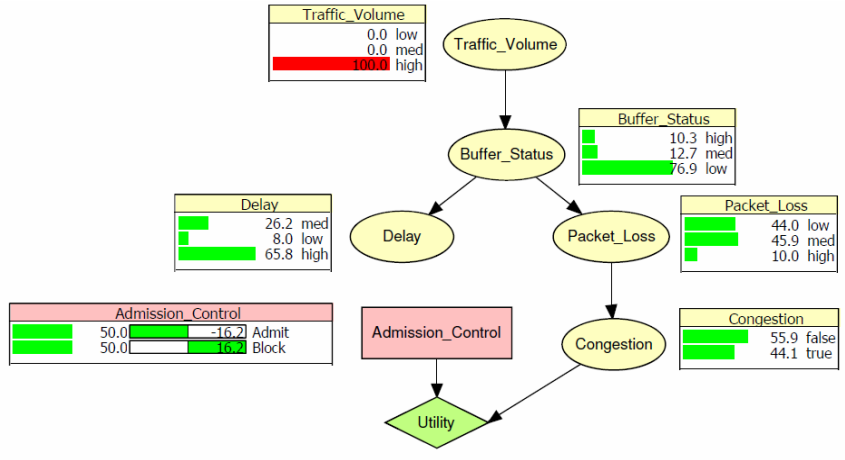

Fig. 8. A sample decision to block a call when *Traffic_Volume* is *high*

5.6 Congestion Prediction Using BBN

One crucial point which can be inferred from the above detailed procedure is that, it is necessary to accurately estimate the state of *Congestion* node and then use it to make the decision to either block a call or admit it. Such a situation can occur where it can be difficult (or costly) to measure the status of *Congestion* node on a frequent basis. So we evaluated the accuracy of this prediction by validating our BBN model on various test data (TD) by changing parameters of the network elements like queue size, service rate, number of sources, etc. These results are presented in Table 5. We compared the predictions of our approach NPC-EM (described in section 5.3) with an existing approach called Tree Augmented Naïve Bayes (TAN) training method. The accuracy of the prediction of the *Congestion* node is represented as percentage of correct estimates (based on the observation of other BBN nodes) and it was found that our approach performed consistently better than the TAN method and provided prediction accuracy of more than 95%. This means that in the absence of direct *Congestion* information, it is still possible to make correct admission decisions based on the learned BBN model. Thus our simulation results have shown that our proposed framework can model the router behaviour and provide intelligent admission decisions for improved QoS management in situations where there is lack of some observations or unavailability of measurement data.

Table 5. Prediction accuracy comparisons for different test data for *Congestion* node

Test data (TD) used	NPC-EM	TAN
TD1 (same as training data)	99.8 %	100 %
TD2 (changed simulation seed)	96.2 %	94.6 %
TD3 (service rate = 1200 bps)	95.8 %	93.7 %
TD4 (queue size = 80 packets)	97.5 %	94.2 %
TD5 (no. of sources = 4)	98.8 %	94.1 %
TD6 (sources shifted by 100s)	97.1 %	95.0 %

6 Conclusions and Future Work

This paper demonstrated that BBNs are capable of modelling the behaviour of a router and implementing a learning-based CAC scheme for improved QoS management. As far as we know, there does not exist a similar solution with which we could compare our approach. However, we successfully showed that BBNs can predict congestion (with more than 95% accuracy) and help the CAC scheme to make accurate decisions in the absence of sufficient data. The case study presented in this paper provides a foundation for performing exciting future research work. As a possible extension to this work, a detailed performance evaluation of BBNs is planned with regard to processing time, CPU utilisation and memory requirements. It would be interesting to compare the BBN based CAC in a centralized and distributed environments and also to see the performance of BBNs when the network scales up to include multiple routers. Also we plan to transform our current setup into an online learning system, which can then make real-time decisions.

Acknowledgments

The authors would like to acknowledge the support of the University of Ulster for funding this research work through a Vice Chancellors Research Studentship.

References

1. ITU-T Y.2001: General overview of NGN. ITU-T Recommendation (2004)
2. Pras, A., et al.: Key research challenges in network management. IEEE Communications Magazine, 104–110 (2007)
3. Wright, S.: Admission control in multi-service IP networks. IEEE Communications Surveys Tutorials, 72–86 (2007)
4. Harrington, D., Presuhn, R., Wijnen, B.: An architecture for describing SNMP management frameworks. RFC 3411, IETF (2002)
5. ITU-T X.711: Open systems interconnection (OSI) common management information protocol: specification. ITU-T Recommendation (1997)
6. Kulkarni, P.G., McClean, S.I., Parr, G.P., Black, M.M.: Deploying MIB data mining for proactive network management. In: Proc. 3rd Intl. IEEE Conference on Intelligent Systems, pp. 506–511 (2006)

7. Sohail, S., Khanum, A.: Simplifying network management with fuzzy logic. In: IEEE Itnl. Conf. on Communications, pp. 195–201 (2008)
8. Hood, C.S., Ji, C.: Proactive network fault detection. IEEE Transactions on Reliability, 333–341 (1997)
9. Ding, J., Kramer, B., et al.: Predictive fault management in the dynamic environment of IP network. In: Proc. IEEE International Workshop on IP Operations and Management, pp. 233–239 (2004)
10. Sterritt, R., Marshall, A.H., Shapcott, C.M., McClean, S.I.: Exploring dynamic Bayesian belief networks for intelligent fault management systems. In: Proc. IEEE International Conference on Systems, Man and Cybernetics, pp. 3646–3652 (2000)
11. Ekaette, E.U., Far, B.H.: A framework for distributed fault management using intelligent software agents. In: Proc. IEEE Canadian Conference on Electrical and Computer Engineering 2003, vol. 2, pp. 797–800 (2003)
12. Sterritt, R., Bustard, D.W.: Fusing hard and soft computing for fault management in telecommunications systems. IEEE Transactions on Systems, Man and Cybernetics 32(2), 92–98 (2002)
13. Korb, K.B., Nicholson, A.E.: Bayesian Artificial Intelligence. Chapman & Hall /CRC Press (2003)
14. Pearl, J.: Probabilistic reasoning in intelligent systems. Morgan Kaufmann, San Mateo (1988)
15. Tanenbaum, A.S.: Computer Networks. Prentice Hall, India (2003)
16. Yang, C., Reddy, A.V.S.: A taxonomy for congestion control algorithms in packet switching networks. IEEE Network, 34–45 (1995)
17. Opnet Modeler, http://www.opnet.com
18. Jensen, F.V.: Bayesian networks and decision graphs, p. 69. Springer, New York (2001)
19. Hugin Lite 6.9, http://www.hugin.com
20. Lauritzen, S.L., Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society, Series B (Methodological) 50(2), 157–224 (1988)