Network Management Using Database Discovery Tools

Mario Gerla and Ying-Dar Lin
Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90024

Abstract

As the volume of network traffic increases due to the proliferation of distributed systems and the growth of real-time applications, a good understanding of traffic distribution and patterns becomes critical in network control and performance management. In this work, we upgrade the facilities of network management from traditional file systems to database and knowledge base systems and apply machine learning techniques to discover traffic patterns which are difficult to discern by human operators among a large volume of measurements. An experiment on interconnected LANs is conducted where some interesting patterns are found. The results show a strong traffic locality and some cyclic traffic patterns. The discovered rule base can describe the traffic distribution and patterns which need to be captured for any sophisticated performance management. The experiment has shown the high applicability of induction techniques to network management.

1 Introduction

Network management has become an important issue due to the rapid growth of networks in the business and research community and the increasing demand for fast, reliable networks to handle high traffic volume. With the introduction of real-time traffic including voice and video, the demand for managing this environment efficiently becomes more significant because real-time traffic requires massive bandwidth and fast response time. Flow control and congestion control problems will become critical since real-time traffic is not tolerant of delay incurred by traditional control mechanisms. In particular, the system needs to predict traffic demands and preallocate resources accordingly.

Network management has been classfied into several categories: configuration management, performance management, fault management, etc. We will focus on dynamic configuration management which permits us to tune the network dynamically. In general, to manage a system, we need to continously monitor its performance and keep track of its status. Thus, a lot of efforts has been devoted in the TCP/IP community on the definition of MIB (Management Information Base) [1], which specifies the information to be gathered, and the definition of SNMP (Simple Network Management Protocol) [2], which permits us to access local and remote MIBs. System status and performance should be stored in a well-defined information base. Based on this information, some short-term (connection acceptance/rejection, congestion control, etc.) and long-term (topology reconfiguration, bandwidth reallocation, etc.) decisions can be effectively carried out. One major problem however is how to handle the large amount of traffic measurements collected in various layers and stations in the network. Traditional manual operation of network management by navigating through the system to diagnose a malfunction or examine system performance will become extremely difficult. Due to the above reasons, it is obvious that the tasks of problem diagnosis, decision making, and control actions need to be handled automatically.

Our proposed approach to solve the problem of traffic measurement interpretation is to upgrade the facilities of network management from traditional file systems to database and knowledge base systems and apply state of the art Artificial Intelligence techniques to network management. In principle, we will monitor the system at different layers and stations. Then, we will organize and store the information in the distributed database. An induction tool will be applied to the database to discover traffic patterns and network malfunctions. The result is a set of rules stored in a knowledge base. With the knowledge base, the sys-

This research was supported in part by a MICRO grant from the State of California and Pacific Bell, in part by a MICRO grant from the State of California and Intelligence Ware, Inc., and in part by a grant from Mitsubishi Electric.

0742-1303/91 $1.00 © 1991 IEEE

tem may be able to diagnose problems, predict traffic, make decision, and trigger control actions by forward inference. The ultimate goal of this work is to make the system self-adjustable.

The stored database will be examined by a machine learning tool called IXL (Induction on eXtremely Large database) [3]. IXL is a software tool developed by IntelligenceWare Inc. and made available to us under a joint research project. It combines machine learning and statistics to distill knowledge from large databases. Basically, it constructs topological neighborhoods for database records and then performs generalizations on these neighborhoods to discover rules which show the correlations between attributes in a relation/view. [4] Discovered rules which represent traffic patterns, network malfunction, system status will be stored in a knowledge base. A traffic controller will then use deduction on those rules to diagnose, predict, and control the network. Thus, the network management system will integrate three subsystems, namely, network monitor, induction tool, and traffic controller. Figure 1 illustrates how induction and deduction techniques can be incorporated into network management.

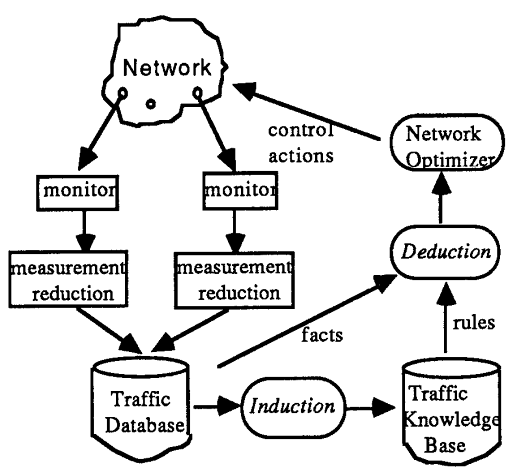

Figure 1. Induction/Deduction for Network Performance Management

Section 2 describes the design and implementation aspects of an IXL experiment based on interconnected LANs. In section 3, the result and analysis of the experiment are presented. Section 4 outlines some possible directions for improving the experiment and points to further research.

2 An Experiment on Interconnected LANs

The goals of the experiment are to understand the traffic distribution in the environment of interconnected LANs, to test the ability of IXL in discovering usual and unexpected traffic patterns, and to observe the stability of traffic patterns and explore its applicability in performance management.

The basic approach in this experiment is to monitor the system at the host and network levels. For each fixed period, we summarize the statistics and insert them into a database. After the whole experiment is completed, we apply IXL to the database to generate a set of rules. These rules will reflect the traffic patterns, and more specifically will give us a cause/effect knowledge about such patterns.

The database discovery technique can be applied to a variety of different traffic and network environments. The most obvious situation is that of a real network on which real traffic measurements are collected. In some cases, however, it may be of interest to inject artificial traffic in the network, to simulate one or more applications and to evaluate the traffic patterns resulting by the interaction of such applications. In other cases, the experiment may even be carried out on a computer simulated network environment, with the purpose of studying the effect of events which are difficult to control in a real network environment (e.g. link/node failures, packet loss, overloads, dynamic network reconfiguration, etc).

In this paper, we describe an experiment on a real, interconnected LAN environment with real traffic. The schema of a set of relations was defined to organize and store the management information. A program was written to process collected measurements and perform data analysis.

2.1 Environment

The experiment is based on the interconnected LAN environment at UCLA, Computer Science Department. There are eight Ethernets and one Appletalk interconnected by routers and more than 300 hosts (including mini computers, multi-user or single-user workstations, etc.), many terminal servers, file servers, news servers, and printers (see Fig. 2). Most hosts run under UNIX. The transport layer protocol is TCP suite. Different networks are interconnected via IP routers. Because of the structure of the TCP/IP address [5], we can tell which LAN a station belongs to by examining its address. This will help in analyzing the traffic flows between LANs. We monitored the traffic on the backbone Ethernet (i.e. 131.179.128 on Fig. 2) which is connected to the off-department network and to Los Nettos. The monitoring program runs on a SUN-4/280 minicomputer which is attached to the backbone Ethernet.

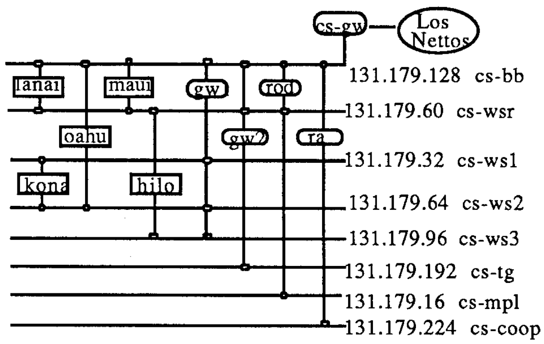

Fig. 2 Interconnected LANs in Computer Science Dept.

The transparent NFS (Network File System) is supported in a way that users can access their own file systems on any host without specifying where they are. This feature accounts for a significant portion of the traffic because of the large amount of file transfers between users' original hosts (or file servers) and current sites. E-mail delivery, remote procedure calls, news reading, file printing, tape backup, human-initiated terminal emulation sessions, and human-initiated file transfers also account for the accumulated traffic amount, in addition to the machine-initiated file transfers mentioned above. It is observed that NFS and window protocol (e.g. X window, SunView) traffic dominates traffic generated by the other protocols like "rlogin", "telnet", and "ftp". With the increasing number of diskless workstations and window users, the profile will become clearer.

2.2 Software Design

To monitor an Ethernet, we use the UNIX network maintainence tool "etherfind". Etherfind detects all the packets transmitted on the Ethernet. It dumps the IP headers and puts a timestamp on them. In the IP header, the following fields are particularly interesting to this experiment: source address, destination address, number of bytes, protocol type, and fragmentation flag. We wrote a program to handle the headers dumped by "etherfind" and couple them together by a pipeline.

Several buffer arrays are used to monitor the current active communication entities. When the "etherfind" handler receives a packet header, it checks the buffer arrays to see if the entity exists. If a match is found, the corresponding entry is updated. Otherwise, a new entry is created. This buffer is swept periodically, for each time slot T, and each entry is either promoted to file entry or flushed. In order to reduce the storage requirements while capturing the most significant traffic components, we promote only those entries which percentagewise contribute most to the traffic in that time slot. The entry with largest contribution will be promoted first and then the second one, etc. When the promoted entries capture P% of total traffic, the promotion process stops and the buffers are flushed. A new time slot then begins. This promotion process for data reduction is shown in figure 3. A typical experiment lasts one to several days. In our experiment, the capture ratio was set to P = 80.

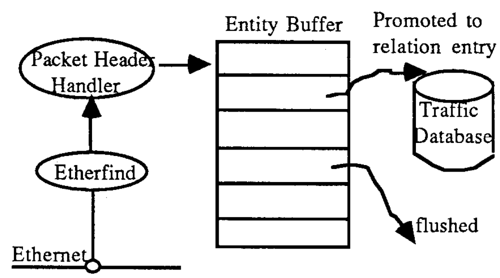

Figure 3. Promotion Process for Data Reduction

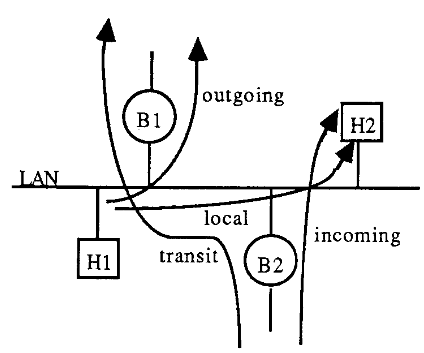

Figure 4. LAN's Internal and External Traffic

The structure of the buffer space is identical to the schema for storing promoted entries in a relational database. Four tables are defined for this experiment. "Summary" just summarizes the total traffic and connections. Traffic is also classified into local (within the local Ethernet), incoming (coming from remote LANs), outgoing (going to remote LANs), and transit (both source and destination are not on this LAN). (see Fig. 4) "Connections" keeps track of the current active communicating node pairs. The traffic amount and type for each pair are recorded. "BLANs" is similar to "Connections" except it is between source LAN and destination LAN, instead of between nodes. "Sources" traces the source nodes which contribute to the traffic on the monitored Ethernet. Here are the definitions of these tables: (note: the fields with underline are keys.)

SUMMARY :

<u>Slot</u> : start time of this time slot
Bytes : # of Kbytes successfully transmitted
TCP : % of tcp traffic transmitted
UDP : % of udp traffic transmitted
Connections : number of node connections
Promoted : %connections being promoted
Captured : %traffic contributed by promoted connections
LocalTraffic : %traffic with source and dest on this LAN
IncomingTraffic : %traffic with only dest on this LAN
OutgoingTraffic : %traffic with only source on this LAN
TransitTraffic : %traffic with source and dest not on this LAN

SOURCES :

<u>Slot</u> : start time of this time slot
<u>Source</u> : source statión address
Bytes : #Kbytes transmitted from this station
Percentage : %traffic from this station
Nodetype : local or remote node

CONNECTIONS :

<u>Slot</u> : start time of this time slot
<u>Source</u> : source station address
<u>Dest</u> : destination station address
Bytes : #Kbytes transmitted between this pair
Percentage : %traffic between this pair
Type : (local, incoming, outgoing, transit)

BLANS :

<u>Slot</u> : start time of this time slot
<u>SourceLAN</u> : source LAN address
<u>DestLAN</u> : destination LAN address
Bytes : #Kbytes transmitted between this LAN pair
Percentage : %traffic between this LAN pair
Type : (local, incoming, outgoing, transit)

The traffic measurements are transfered from SUN-4/280 to PC DOS disks via IBM RT after the monitoring process is completed. IXL then runs on those relational tables in an IBM PC/AT. Each IXL run takes from several minutes to several hours, depending on the size of relational tables and various discovery parameters set in IXL. Also, the number of generated rules depends heavily on the settings of discovery parameters. By properly setting these parameters, we can direct IXL to find the traffic distribution and patterns we need. In this experiment, we monitored for 5 days. The sizes of generated tables are from 300 to 5000 tuples. The running times of IXL on these tables are between 10 minutes to 5 hours. The numbers of discovered rules are between 10 to 100. Typically, several rounds of experiments are required in order to adjust table size, IXL running time, and focus of discovery.

IXL also supports the definition of "concepts", which are virtual fields derived from other existing fields. These "concepts" can reduce the running time of IXL and help focusing the discovery process. In our experiment, we define the concept "Traffic" to classify the levels of traffic volume. For example:

Traffic = very high if Bytes $\geq$ 10 Mbytes;
Traffic = high if 10 Mbytes > Bytes $\geq$ 5 Mbytes;
Traffic = medium if 5 Mbytes > Bytes $\geq$ 1 Mbytes;
Traffic = low if 1 Mbytes > Bytes;

IXL has a set of parameters which tailor its performance to the user's need. [3] Major discovery parameters include the following: maximum number of clauses in rules (an upper limit for the length of a rule), minimum number of records (a lower limit for the number of records involved in forming a rule), minimum confidence in rules (a lower limit for the confidence in a rule), maximum margin of error (an upper limit for the error invloved in estimating the confidence in a rule), minimum percentage of database (a lower limit for the fraction of the database invloved in forming a rule), minimum significance (a measurement of the quality of a range in terms of how the distribution of values in that range varies from the rest of the database where 0 means that almost all ranges are considered and 100 means that only highly significant ranges are considered), minimum generality (a upper boundary for the range sizes determined by IXL), maximum generality (a lower boundary for the range sizes), generality increments (an indicator of the number of ranges between the maximum and minimum generality parameters where 0 means only two ranges, maximum and minimum generality, are considered and 100 means up to 20 ranges are considered), and interest level (user's interest in the effect that a field has on the goal).

2.3 Architecture of the Traffic Pattern Observer

Figure 5 is the overview of the Traffic Pattern Observer which is composed of several tools integrated by user interface. The major components of the system are Monitors, IXL, and a set of utilities. Monitors will activate a set of tools to monitor traffic and, at the same time, a set of handlers to handle the traffic data generated by those tools. The utilities will provide an interface to the database query language and also contain tools for defining and setting up database. The Database Interface is the one providing transparency of the DBMS (Data Base Management System) used so that we can switch to another DBMS without affecting other components of the system.

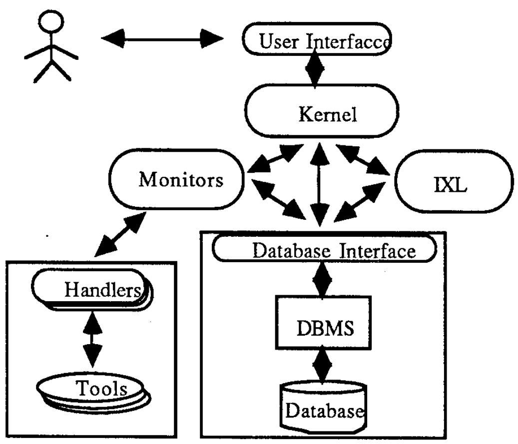

Figure 5. System Architecture for the Traffic Pattern Observer

Before the experiment can be conducted, the database schema must be defined in the DBMS. For each tool, there should be a base table in the database and a handler associated with it so that all of the tool handlers can work concurrently. Also, the structures of the tool handlers depend on the schema specified in the DBMS.

At the conceptual level of the database, the base table schema is fixed; however, the user may have the freedom to supply the view definition which depends on the expected knowledge to be discovered. If the user supplies his/her own view definition, there may be some meaningless results generated by IXL if the view definition has some defects like join of two base tables with no common column. To guarantee a reasonable result, the system will provide a set of view definition for users to choose from.

During the monitoring process, there are a set of processes working concurrently. Some are listening to the Ethernet and pumping the information they have captured, some are processing the pumped data and maintaining the data structures to keep track of the summarized statistics, some are busy with the database interface to insert records into the base tables. The data structures maintained in the handlers are inserted into base tables and purged every period of time. Although there are a lot of process working on this job, which can be considered as a considerable overhead to the system, only processes which fetch the host information by some remote execution mechanism will transmit packets on the Ethernets. The influence on the results concerning network traffic can be small. However, the performance of the local host may be affected. If the experiement program is run on a dedicated workstation, there will be no influence to other hosts. Furthermore, the communication overhead can be minimized if we run IXL at the same site where the database is located.

It is expected that we may have new tools for monitoring some other activities. If a new tool needs to be included, the system maintainer needs to do the following:

1. Supply a handler associated with the new tool and insert a new entry in the tool table of Monitors subsystem which may invoke the new tool during monitoring process.

2. Insert new base table definitions into the original schema and those new base tables will contain the information available via the tool.

3. Create new view definition associated with the new base tables and those new view definitions will be new alternatives for users to choose from before IXL is invoked.

3 Experimental Results

Table 1 contains sample data for the defined relations. The experiment includes two sample runs on the tables "summary" (288 tuples) and "BLANs" (2151 tuples) where numbers of rules found are 43 and 53, respectively. The IXL running time is 13 minutes for "summary" and 1 hour 50 minutes for "BLANs". In these sample runs, we focus on the discovery of relationship between traffic volume and other fields. Thus, we make the defined concept "Traffic" as our goal attribute in the rules to be discovered.

SUMMARY:

time	#pkts	#bytes	tcp	udp	#con	P	C
2155	157287	23890293	57	41	376	5	81
2255	180038	25853626	60	38	361	6	80
2355	203861	35543446	68	30	384	4	81
55	529828	278920170	97	2	357	0	89
155	371640	160224249	96	3	394	0	85
255	468054	297290623	73	26	330	0	89
355	92946	15864225	48	50	309	5	81
455	125636	22003480	55	43	327	4	81
555	73120	11954986	60	38	313	4	82
655	99099	18044675	61	37	330	4	80
755	159400	29554676	54	45	365	5	80

SUMMARY:(cont.)

loc	in	out	tran
32	33	31	2
39	28	26	4
21	26	43	6
2	4	92	0
3	5	89	0
0	51	23	24
10	38	46	3
22	33	38	3
13	36	44	4
13	37	44	4
24	46	25	3

CONNECTIONS:

time	src	dest	#pkts	#bytes	%	type
1655	outside	128.12	15568	2218746	5	incom
1655	128.34	128.11	8673	2033508	5	local
1655	128.34	128.61	11476	1800922	4	local
1655	96.44	outside	3947	1552406	3	trans
1655	128.34	128.12	4221	1517826	3	local
1655	96.11	128.12	4161	1478578	3	incom
1655	outside	128.34	9141	1302577	3	incom
1755	128.34	outside	19860	5786346	16	outgo
1755	128.34	128.61	14007	2196248	6	local
1755	outside	128.12	15475	2078924	5	incom
1755	outside	128.34	14100	1571707	4	incom

BLANs:

time	src	dest	#pkts	#bytes	%	type
1655	outside	128	35561	5229127	13	incom
1655	128	outside	28584	4600075	11	outgo
1655	128	64	14641	3578396	9	outgo
1655	96	128	12462	2626949	6	incom
1655	128	192	12971	1998201	5	outgo
1655	96	outside	3977	1558137	3	trans
1655	64	128	12227	1310143	3	incom
1755	128	128	75853	8996668	24	local
1755	128	outside	31711	6952494	19	outgo
1755	outside	128	38694	5135791	14	incom
1755	64	128	14465	2501906	6	incom

SOURCES:

time	src	#pkts	#bytes	%	type
1655	128.11	41513	5065412	12	loc
1655	128.13	31232	4548570	11	loc
1655	128.61	32790	2121223	5	loc
1655	96.11	5735	2016056	5	rem
1655	96.44	4261	1574839	4	rem
1755	128.34	42752	11623013	32	loc
1755	outside	45160	5615596	15	rem
1755	128.13	29109	4390597	12	loc
1755	128.11	34159	4146715	11	loc
1755	128.61	35554	2281736	6	loc
1755	64.12	8371	2105048	5	rem

Table 1 Sample relation tables

Of those discovered rules, some are particularly interesting to us:

CF=85
"traffic" = "very high"
IF
"0:55" ≤ "timeslot" ≤ "1:35"
AND
"91%" ≤ "outgoing" ≤ "94%" ;

CF=95
"traffic" = "high"
IF
"12:28" ≤ "timeslot" ≤ "13:53"
AND
"sourceLAN" = "131.179.64"
AND
"destLAN" = "131.179.192" ;

CF (confidence ratio) in the rule means the percentage of records satisfying the goal among the records satisfying the conditions of the rule. The first rule is discovered for "summary" where "very high" means volume is larger than 10 Mbytes in a 5-minute slot. This rule indicates that from 0:55AM to 1:40AM, outgoing traffic accounts for around 90larger than 10 Mbytes/slot. Actually, this happens when the system is backing up its file system to tapes every morning around 1:00AM to 2:00AM. That most traffic is outgoing implies that the backup tape is not on the backbone Ethernet. Indeed, the backup machine is "131.179.32.11", a SUN-4/280 on another LAN. We believe a peer rule will be discovered if we run the same experiment also on the LAN where the backup machine resides, except that "outgoing" becomes "incoming". Since the traffic volume caused by tape backup varies each day, there is a high degree of fluctuation in the periods of tape backup as shown in figure 6. However, we can still find the correlation and the cycle.

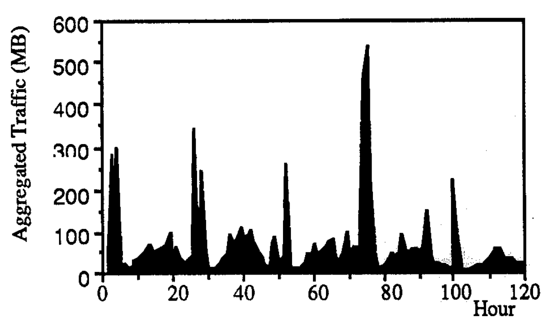

Figure 6. Traffic Cycle and Correlation

The second rule above is discovered for "BLANs" where "high" means volume is larger than 500 Kbytes/slot but smaller than 1 Mbytes/slot between source and destination LANs. This rule means that the traffic volume from LAN "131.179.64" to LAN "131.179.192" between 12:28PM and 13:58PM is between 500 Kbytes/slot and 1 Mbytes/slot. This type of rule can be very useful in understanding the traffic distribution with respect to topology and time. It captures the traffic distribution in a three-dimentional traffic matrix shown in figure 7.

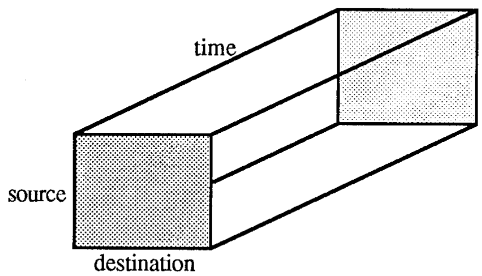

Figure 7. A Conceptual 3-D Traffic Matrix

The analysis of the rules discovered during the experiment leads to the following general observation:

Locality:

More than 80% of the traffic is contributed by less than 10% of communicating pairs, ie. traffic is not uniformally distributed. It is essential to capture this distribution in order to optimize the network configuration. A typical degree of locality is shown in table 2.

Correlation:

A temporal cycle exists in the traffic distribution. Being able to keep track of the distribution cycle will enable the dynamic configuration management which tunes the network dynamically.

Burstiness:

If we consider the burstiness in terms of different time scale, the inter-slot burstiness (long-term) is reflected by the cycle, while the intra-slot burstiness (short-term) can be approximated by a Batch Poisson or Markov-Modulated Poisson process. This is due to the fact that we summarize the measurements for each slot, thus some details within the slot are lost and can only be approximated by a stochastic process.

For the discovery process, tuning the learning parameters to fit the need of the application is not a trivial task. In order to have a reasonable set of discovered rules, IXL parameters must be carefully set. For example, too few rules will be generated if the minimum confidence is too high. The sizes of the ranges for "timeslot" in the rules will be too small if the generality increments are set to zero (default). One of the limitation of IXL is that it is not suitable to learn the correlation within the numerical values. It must rely on the proper classification of the numerical domain to reduce the number of unique values to be handled.

Since the discovered knowledge is expressed in production rules, the system administrator can easily understand the semantics of the traffic measurements. Meanwhile, as illustrated in figure 1, deduction engine can be applied to this knowledge base to diagnose and control the network.

C %	P %
10	0.9
20	1.3
30	1.9
40	2.7
50	3.6
60	4.7
70	6.8
80	9.7
90	16.5
95	28.4
98	41.2

Table 2 Degree of Traffic Locality

4 Conclusion and Future Work

Current applications of AI techniques to network management are mainly for fault management by expert systems where the knowledge is specified by the human experts, instead of being learnt from the historical data. [6][7] The experiment on the interconnected LANs of UCLA Computer Science Department has shown the high applicability of induction techniques to network performance management, especially for medium-term and long-term control schemes. An evaluation is made on the semantics of the rules generated. The discovered rule base can describe the traffic distribution and patterns which need to be captured for any sophisticated performance management. Our testbed has strong traffic locality where only a small subset of possible connections contribute significantly to overall traffic at any given time. We believe that the traffic locality in a large network with real-time applications is more stable and hence more predictable on a medium- to long- term basis.

We plan to run our experiment extensively to further justify our observations and explore other traffic patterns that can be captured from the database into the knowledge base within the interconnected LAN environment. The experiment will be augmented and refined to capture the exact information we need for our application. For example, we need to see the effect of adjusting the length of time slot and find its optimal value for a particular application. Each numeric attribute in the relations can be classified into several levels in order to cut down the running time for induction where too many distinct values will lengthen the discovery process.

Acknowledgements

The authors wish to acknowledge Kamran Parsaye and Rei-Chi Lee at Intelligence Ware, Inc. for the discussion on IXL. And thanks to Meng-Chang Chen at AT&T Bell Laboratory for providing perceptive comments on an early draft of this paper.

References

[1] McCloghrie, K., and M. Rose, *Management Information Base for Network Management of TCP/IP-based internets,* RFC 1066, August 1988.

[2] Case, J., M. Fedor, M. Schoffstall, and J. Davin, *The Simple Network Management Protocol,* RFC 1067, August 1988.

[3] IntelligenceWare, Inc., *IXL: The Machine Learning System User's Manual,* IntelligenceWare, Inc., 1988.

[4] Parsaye, K., et al., *Intelligent Databases: Object-Oriented, Deductive, Hypermedia Technologies,* p.404-415, John Wiley, New York 1989.

[5] Postel, J., Editor. *Internet Protocol DARPA Internet Program Protocol Specification,* Tech. Rept. RFC791, DARPA, 1981.

[6] Liebowitz, J., Editor. *Expert System Applications to Telecommunications,* John Wiley, New York, 1988.

[7] IEEE, *Special Issue on AI Applications to Telecommunications,* Journal on Selected Areas in Communications, June, 1988.