A Neural Network Based Predictive Mechanism for Available Bandwidth

Alaknantha Eswaradass, Xian-He Sun, Ming Wu
Department of Computer Science
Illinois Institute of Technology
Chicago, Illinois 60616, USA
{eswaala, sun, wuming}@iit.edu

Abstract

Most recent developments of computer sciences, such as web services, Grid, peer-to-peer, and mobile computing, are network-based computing. Their applicability depends on the availability of the underlying network bandwidth. However, network resources are shared and the available network bandwidth varies with time. There is no satisfactory solution available for network performance predictions. This lack of prediction limits the applicability of network-based computing, especially for Grid computing where concurrent remote processing is essential. In this study, we propose an Artificial Neural Network (ANN) based approach for network performance prediction. The ANN mechanism has been tested on classical trace files and compared with the well-known system NWS (Network Weather Service) for performance. Experimental results show the ANN approach always provides an improved prediction over that of NWS. ANN has a real potential in network computing.

Keywords: Performance prediction, Network bandwidth, Artificial neural network, Distributed computing

1. Introduction

The concept of available bandwidth has been the center of attraction throughout the history of packet networks. For many data media applications, such as file transfers or multimedia streaming, and network technologies like content distribution networks, end-to-end admission control and video/audio streaming, the bandwidth available to the application directly impacts application performance. Peer-to-peer applications form their dynamic user level networks based on available bandwidth between peers. Overlay networks can configure their routing tables based on the bandwidth of overlay links. Though in Grid environments, computing resource can be reserved through the 'Service-Level Agreement', network bandwidth is not a subject of reserve at this time. The availability of network bandwidth is an important factor in choosing web service [1].

In addition, accurate and timely bandwidth measurement and prediction are useful for mobile computing. The mobile computers frequently have more than one network interface with varying bandwidths. The mobile hosts will have the flexibility to choose the highest bandwidth interface if the future available bandwidth is known [2]. Finally, at the network level, we could also use bandwidth information to build multicast routing trees more efficiently and dynamically. Ideally, multicast routing trees would be built so that packets travel along a tree that minimizes duplicate packets and latency while maximizing bandwidth [2].

Two throughput metrics that are commonly associated with a network path are end-to-end capacity C and available bandwidth A. In a shared environment, the available bandwidth varies with time. We need to measure and predict the available bandwidth in a timely fashion.

2. Background and Related Work

Performance monitoring and forecasting is an active area of research. We can classify the bandwidth measurement and prediction methodologies in several ways like active or passive, end-to-end or hop-by-hop, Simple Network Management Protocol (SNMP) gathered versus actively probed [4], network intrusive versus network friendly [5] and so on. There are Variable Packet Size (VPS) probing methodology that measures the hop-by-hop metrics, and Packet Train Dispersion (PTD) probing methodology that measures the end-to-end metrics. The two famous tools in this category are

Pathrate [6] and Pathload [7] used for estimating capacity and available bandwidth respectively [8].

All these methodologies provide only the measurement part of the available bandwidth, and they are not concerned with the prediction of the available bandwidth, which serves as useful information in routing decisions and provides guideline in task scheduling. Unfortunately, due to the heterogeneity and the constantly varying nature of the network traffic, there are only a few works available to provide prediction of network performance in terms of available bandwidth and latency in a heterogeneous environment, such as Grid computing. The Network Weather Service (NWS) [9, 10] is a well-used network performance measurement and prediction system in Grid computing. It periodically uses active network probes to collect the end-to-end network performance data and dynamically forecasts the performance of the various networks. However, its simple prediction methods (mean-based, median-based, autoregressive) cannot satisfactorily capture the complicated short and long-range temporal dependence characteristic of heterogeneous network traffic. While NWS has made its contributions, new prediction methods for network traffic need to be explored for better solutions.

In this work, we investigate the Artificial Neural Network (ANN) prediction approach, via machine learning algorithms that learns the model of the system from the system itself. The ANN approach can develop a model specific to each network system and provides a good approximation of the underlying real system.

Several works of using neural network for online-prediction can be found in the literature, including financial forecasting in the stock market, electric load forecasting in power networks, fault prediction in process control, call admission control, link capacity allocation in ATM networks and prediction of network congestion [11-14]. Some efforts have also been made in prediction of a specific application's network traffic with the neural network. A multiresolution learning neural network has been constructed to predict VBR video traffic for dynamic bandwidth control using real-world VBR video traffic traces [15]. Yousefi [16] used neural network to model bursty teletraffic pattern in terms of packet number. His work was tested on artificially generated traffic by chaotic maps. Different from other's work, we use the neural network to predict the available bandwidth of any application by analyzing the network traffic, instead of specific applications. Our work has been tested on long-term real-world network traffic traces.

There has been some other work in using the linear system representation structures like ARMA, ARX and ARMAX [17] for bandwidth predictions. However, they lack the potential of accurately describing the behavior of extremely complex systems. Neural network scheme is a non-linear representation of the system alleviating the problems of the linear models.

The rest of the paper is organized as follows. Section 3 presents our proposed predictive mechanism for predicting the available bandwidth using ANN; Section 4 presents experimental results and analyzes them; Section 5 discusses the run time prediction; and finally Section 6 gives a summary and discusses future research.

3. Neural Network Based Bandwidth Prediction

Today, there are various forecasting models prevalent, employed for various types of applications, such as experience-based deterministic models, statistical models, probabilistic and stochastic models, AI and machine learning models, and genetic algorithm based approaches. Each model has its own pros and cons, and today researchers are striving hard to use these models individually or in combination to achieve accurate forecast results. NWS is based on statistical modeling. Our previous work is based on probability and stochastic modeling [18, 19]. In this study we investigate the solution using artificial neural networks.

We describe the network traffic prediction problem as follows: Given the observed traffic data at time unit i, $T(i)$, where i=1…t, the prediction is to generate an estimate of $T(i+s)$, where s is the prediction horizon or the future time unit. The network traffic data is highly non-linear and varies with time. It changes abruptly when entering or leaving a congestion hour. Therefore, to predict the dynamic nature of the traffic data we need to devise an accurate prediction model. We use neural networks, with their remarkable ability to learn from examples and derive meaning from complicated or imprecise data, to extract patterns and detect trends of available bandwidth.

An ANN is a non-linear classifier of machine-learning algorithms. We use the Weka machine learning software [20] in Java in our study. Weka is a collection of machine learning algorithms for solving real-world data mining problems. The strengths of ANN are its outstanding learning abilities, robustness to noise, and little a priori knowledge needed. An ANN builds itself through the process of learning from "experience". This process is called ANN training. By online learning, ANN model can take into account the changes in the environmental conditions and adapt itself to the changes. These characteristics of the ANN have made it a potential solution for the prediction of network traffic, which

presents complicated short and long-range temporal dependence.

An ANN is constructed with many computing cells, called perceptrons, as shown in Figure 1. Each perceptron unit calculates the output by applying a non-linear continuous function over the sum of the product of its input and the corresponding weights [21].

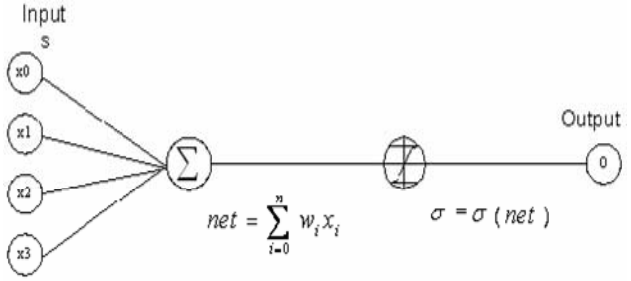


Figure 1. Perceptron of neural network

In our study, we use the squashing function as the non-linear function. The squashing function shown below has the derivative that can be represented in terms of its output, thus simplifying the computation.

$$\sigma(y) = \frac{1}{1+e^{-y}}$$

$$\frac{d(\sigma(y))}{dy} = \sigma(y) \times (1 - \sigma(y))$$

A simple perceptron is one in which the weight changes by an amount proportional to the difference between the desired and the actual output, and the network adapts accordingly. At some point, when the difference between the actual and the desired output is almost 0 for all patterns, the weight ceases to adapt indicating that the network has finished learning.

A multi-layer perceptron (MLP), is one in which weights are changed by an amount proportional to the error at that unit times the output of the unit feeding into the weight. Learning in MLP networks is effectively learning the weights in the network and is performed by the back propagation algorithm. According to this algorithm, the error in the output is used to update the weights in the network. Depending on the learning rate, the effect of the error is propagated back throughout the network. This leaning process is repeated epoch-by-epoch until the difference between the neural network predicted output and the actual value reaches a predefined threshold. The principal advantages, such as simplicity, reasonable speed, and ability to acquire arbitrarily complex nonlinear mappings, support us to incorporate the back-propagation algorithm in the multi-layer perceptron network for our prediction model.

3.1 Network traffic data statistics

Collecting wide-area network traffic data is a challenging task due to dynamics of network traffic, difficulties to access production network, and sensitivities of data privacy. Thus, instead of collecting network traffic on our own, we use trace files, from WAND, ITA [22], and MOAT [23] to verify the effectiveness of the ANN mechanism. Trace files are the network traffic, bi-directional packet data, i.e. packet traces with accurate timestamps. These NSF-funded projects collect network traffic data from real production systems. These data are more representative for wide-area traffic than that of our local network environments.

3.2 Algorithm: Statistics of Network Traffic Data

The network traffic data provided by WAND, ITA and MOAT are collected using passive measurement methodologies that use the trace history of existing data transmission. This methodology is potentially very efficient and accurate. The trace files represent different types of network traffic, and hence, experimenting on them verifies the accuracy and correctness of the proposed prediction scheme. However, the raw trace file needs to be pre-processed before an application like neural network can consume it.

Each trace contains packet arrivals, with IP header and TCP header. Each individual packet contains some specific properties of the packet, such as time stamp, packet length, source and destination IP addresses, which could be obtained from the IP Header. Compared with the individual packet information, we believe that the number of packets in each second and the number of bits in each second are sufficient to produce estimates of the consumed bandwidth over time. Thus to collect the information about number of packets in each second and the number of bits in each second from the raw trace file, we present the following algorithm.

1) Parse the entire trace file.
2) Parse the timestamp and right shift by 32 bits to get only the data in seconds discarding the nanoseconds.
3) For each unique timestamp, the corresponding entries for the number of packets and the number of bits are added.
4) Finally a table with the following columns is obtained. Time, number of packets in one second, number of bits in one second.
5) From this table the average number of bits per second gives the link utilization.

The available bandwidth of link i can be calculated as follows: $A_i = C_i(1 - u_i)$, where u_i is the link utilization and C_i is the link capacity.

3.3 Bandwidth prediction using neural network

The resultant network traffic data obtained from the raw traces must be assigned to specific bins in order to serve as input to the predictive system. The network traffic packets are binned into non-overlapping bins of a specific size called the bin size. According to user's prediction requirement, different bin sizes are chosen in generating training data for neural network. For example, when the prediction of available bandwidth in the next 5-minute is required, the bin size is set as 300 seconds. We often call this a one-step prediction. For each bin size, using the measured network traffic data up to the i_{th} bin, the bandwidth of $(i+1)_{th}$, $(i+2)_{th}$, …, $(i+10)_{th}$ bins are predicted respectively. Prediction of the $(n+i)_{th}$ bin using the measured bandwidth data up to the i_{th} bin is referred to as the *n*-step prediction.

One problem in the construction of neural network model is the tradeoff between prediction accuracy and cost. The cost includes training cost and prediction cost, which are related to the number of input parameters. In general, with more input parameters, the prediction accuracy is better. However, the increase of input parameters will lead to higher cost, which is a set back for real-time prediction. Instead of using all possible input parameters, we need to identify a small set of necessary data. In our experiments, we initially generate and use as the network traffic parameters for each bin size: timestamp, minimum packet rate, maximum packet rate, average packet rate, minimum bit rate, maximum bit rate, average bit rate, average bit rate in the past *n* minutes, average bit rate in the past $n+m$ minutes, average bit rate in the past $n+2m$ minutes and so on till average bit rate in the past $n+i*m$ minutes. Here, packet rate denotes the number of packets in one second, bit rate denotes the number of bits in one second, *n* is the initial value, *m* is the successive difference between each past interval, and *i* denote the number of past data required for training the neural network.

After exhaustively examining all combinations of the above defined input parameters based on trace files from WAND, ITA, and MOAT, we have chosen the following parameters as they are useful information for bandwidth prediction: timestamp, average packet rate, average bit rate, and their past information. The maximum and the minimum packets/bits rates in that bin size are irrelevant attributes. The relevant attributes will form the training input to the neural network. We also observed if the traffic exhibits a high variability in the number of bits transferred, the values of *m*, and *n* should be set small and the value of *i* should be set large, as more training would be required to make accurate predictions.

Selecting an appropriate training size for network bandwidth prediction is another problem in ANN construction. In our experiment, we tested short-term and long-term trace files. We found that, the performance of the ANN prediction is not satisfactory for short-term trace file containing traffic data for a couple of hours or less than 1 day. This is caused by inadequate training data. For long-term trace files containing traffic data more than 3 week or a month, we observed network traffic data in 7-10 days is enough for neural network training. A longer training phase doesn't improve the prediction accuracy.

We present the prediction mechanism below. Different bin sizes we choose for our experimentation are 10, 60, 100, 300, 600, 900 and 1800 seconds. We choose the prediction steps from one-step to five-step. The training input data for the neural network for each bin size contains all the relevant attributes mentioned above.

1) Choosing appropriate parameters in the neural network model building

a. Learning rate: The learning rate controls the level of change applied to the weights after each training period. With a high learning rate the neural network will react fast to abrupt changes, which is not favorable. Therefore, we set the learning rate to 0.01 so that the overall bandwidth pattern will be learned.
b. Maximum number of epochs: The number of times a data set is trained on is the number of epochs of training. Training termination is determined either by setting a maximum number of epochs or if the neural network's output error becomes less than some preset threshold. For the best prediction results, we set the epoch number to 700.
c. Number of layers and the number of perceptron units used in each layer:
    i. Input Layer: The input layer adds one perceptron unit for each of the defined training input parameter: timestamp, average packet rate, average bit rate, and its past information.
    ii. Hidden Layer: The neural network contains more than one hidden layer in order to improve the accuracy of the prediction. The input of each neuron of the next layer is connected with the outputs of all neurons of the previous layer. Analyzed data are treated as neuron excitation parameters and are fed to inputs of the first layer. These excitations of lower layer neurons are propagated to the next layer neurons, being amplified or weakened according to weights

ascribed. The number of such hidden layers and the number of nodes to be used in each layer are important parameters. We set the number of hidden layers as 3 and the number of perceptron units in each hidden layer as 3 for accurate prediction results. Our experiments indicate a neural network model with more than 3 hidden layers does not improve the prediction accuracy.

iii. Output Layer: This layer contains the perceptron unit for the final output attribute of the neural network. In our case, the output is the available bandwidth.

2) Using constructed neural network to predict network bandwidth

The output of the neural network is the bandwidth data predicted at the next *n*, $n+m$, $n+2m$ and so on up to $n+i*m$ minutes. The values of *n*, *m* and *i* depend again on whether the traffic exhibits a high variability or not. If the variability is high the values of *n*, *m* and *i* are small; otherwise they are large.

A neural network as described above is constructed and experiments are carried on using the different types of network trace files. The experiments and the results are described in the next section.

4. Experimental results

The AUCKLAND trace files [24] have been chosen for the entire analysis. AUCKLAND IV is a continuous 6 1/2 week trace between February and April 2001 at the University of Auckland uplink, and AUCKLAND II is a collection of 1-day trace between December 1999 and June 2000 at the University of Auckland uplink. Neural networks need training data. We choose AUCKLAND traces since they are long-term traces. Auckland's traffic reaches the Internet via a single ATM link; all packets (in both directions) can be seen. The connection has a packet peak rate of 2 MBits/sec set in each direction (4MBits/sec when the two trace files are aggregated).

Delay measurements in general require accurate high-resolution timestamps. NTP is not good enough for this purpose, since it only provides millisecond accuracy. All interfaces involved in measuring delay must be synchronized. In order to satisfy this requirement, the WAND group has developed the DAG cards: Ethernet (10/100 Mbps) and ATM (OC3, OC12, OC48), AAL5 and PoS. Packet traces at Auckland are collected using two DAG cards in a Pentium based PC. The DAG cards connect to the Auckland ATM link via optical splitters [25, 26].

To verify the efficiency of our Neural Network based prediction approach, comparison is made with that of Network Weather Services (NWS), a widely used model for prediction. Experiments were conducted to test the performance of the predictive model presented here that uses ANN.

The primary metric we have used for evaluation is the relative prediction error,

$$err = \frac{\Pr edictedValue - ActualValue}{ActualValue}$$

where *err* is the relative prediction error, *PredictedValue* is the bandwidth predicted for the next *n* seconds and *ActualValue* is the bandwidth measured for the next *n* seconds where *n* is chosen by the user. Mean error which we use for our computations is calculated by averaging all of the relative errors.

Figure 2-5 show the performance of bandwidth prediction using the ANN mechanism and NWS on various bin sizes and various time periods of the AUCKLAND II and IV traces. One step, 2 step, …, 5 step predictions are performed for each bin size. The charts show that the predictability is good for the one-step and the two-step predictions. With subsequent step predictions, the prediction error gradually increases. This is mainly because, as the step number increases, the prediction moves from short term to long term and the accuracy slope down. Thus the ANN mechanism and NWS are both more accurate for short-term predictions. For the AUCKLAND IV traces, it could be observed from the graphs that the best prediction is obtained at the bin size of 60 seconds for one-step prediction in both the ANN mechanism and NWS. For the AUCKLAND II traces, we can infer from the graph that the best prediction is got at a bin size of 300 for one-step prediction. The variation is very low in the prediction errors between the chosen bin sizes for each step prediction with the ANN mechanism. Thus, bin size is not a determining factor of the success of the ANN based prediction approach.

4.1 Performance comparisons of the ANN mechanism and NWS

The performance comparisons of the ANN approach and NWS are made for varying bin sizes for one step and two-step prediction of AUCKLAND II and IV traces.

Figure 6-7 shows the performance comparisons of the ANN and NWS for varying bin sizes for one and two-step prediction of AUCKLAND IV and II traces. Fig (a) and Fig (b) show the one-step and the two-step predictions, respectively. From the graph we can see that the prediction results of ANN supercede those that of NWS for each bin size, illustrating that the performance of the ANN mechanism is noticeably better than that of NWS. The best results at the bin size of 60 for the AUCKLAND IV trace file have been found to be 10.54% mean error using the ANN and 12.74% mean error with NWS with

the one-step prediction. In the case of AUCKLAND II traces, the results are best at a bin size of 60 seconds with 13.72% mean error using the ANN approach and at a bin size of 300 seconds with 15.80% error using NWS with the one-step prediction.

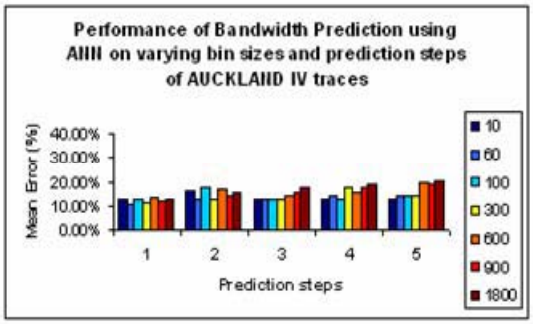


Figure 2. AUCKLAND IV: ANN

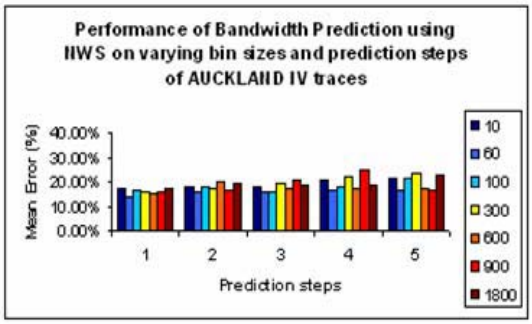


Figure 3. AUCKLAND IV: NWS

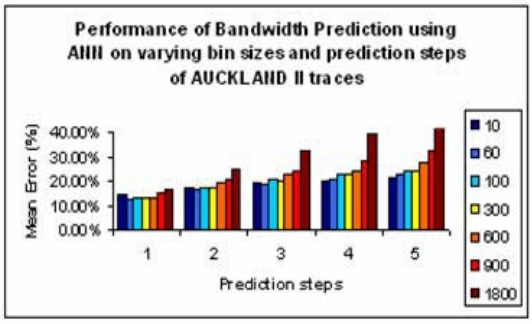


Figure 4. AUCKLAND II: ANN

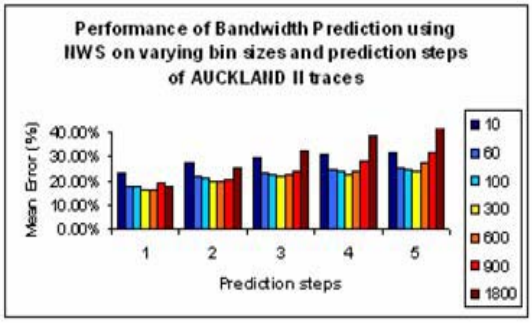
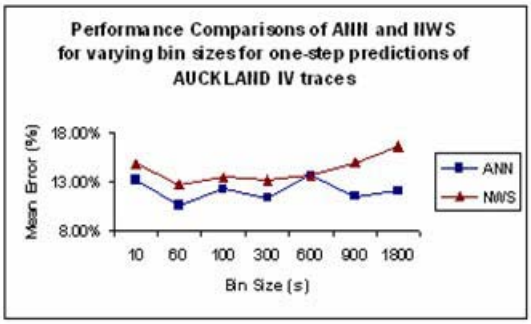
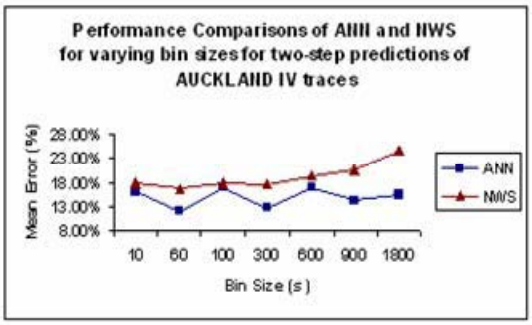


Figure 5. AUCKLAND II: NWS

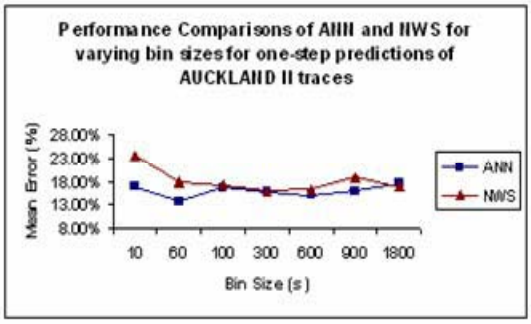


(a) One-step Prediction

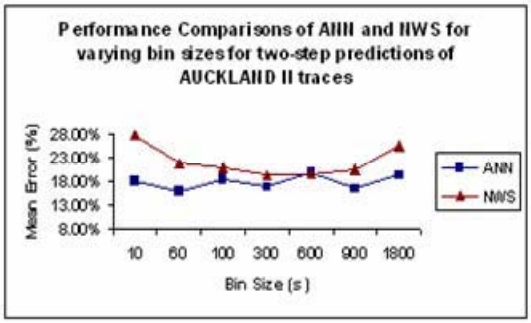


(b) Two-step Prediction

Figure 6. AUCKLAND IV



(a) One-step Prediction



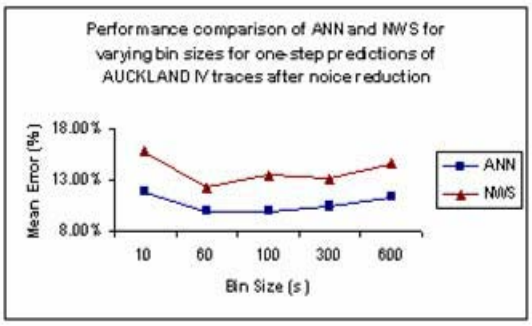
(b) Two-step Prediction

Figure 7. AUCKLAND II

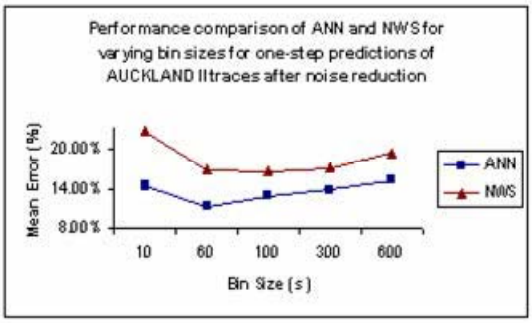
Network traffic is usually formed by different applications. Each application may have its own network communication pattern. The strength of ANN is its outstanding learning abilities to capture the existing pattern from training data, if a pattern exists. If we build a neural network for each different application individually, the prediction should be more accurate for each application, and then, with an appropriate integration, the combined prediction should result in a better overall traffic prediction as well. Instigated by this thought, we examined the network traffic composition in the trace files. We found the network traffic in AUCKLAND trace file is composed of different types of application traffic like TCP, UDP, ICMP and others. The application traffics statistics indicate that each type of traffic data presents a different traffic pattern. Using one neural network to characterize the overall traffic pattern may hide the heterogeneity of the network traffic. Instead of training one neural network to learn different patterns, we constructed an individual neural network model for each of them. For each bin size, we isolated TCP, UDP, ICMP and other traffic data and used them to build neural networks separately. Then we combined the individual prediction results into the overall traffic prediction. Experiment results show the bandwidth prediction accuracy is improved in this way. For example, the ANN prediction error is reduced from 13.60% to 12.31% at bin size of 600 for AUCKLAND IV trace file. In the experiment, we observed that the prediction accuracy of UDP and ICMP traffic is significantly lower than that of TCP. This is because around 95% of network traffic in AUCKLAND trace files is TCP data. The limited UDP and ICMP traffic data prevents ANN from learning their traffic patterns sufficiently. As a result, they are considered as noise and eliminated. We then use the ANN model constructed from TCP traffic to predict the overall network bandwidth. After noise reduction, we obtained the predictions at different bin sizes for one-step with AUCKLAND IV and II traces using the ANN and compared with NWS.

Figure 8 (a) and (b) illustrate that the prediction results of the ANN mechanism are further improved compared with results in Figure 6 (a) and Figure 7 (a). The experimental results indicate that the ANN prediction approach has a potential to present a better performance than NWS in any situation.

Table 1 summarizes the experimental results of the original traffic prediction, the prediction after analyzing the network traffic composition (before noise reduction) and the prediction after removing the negligible constituents of the network traffic (after noise reduction) for one-step. From the table we see that, constructing individual neural networks for each type of traffic and integrating to obtain the overall prediction reduce the error percentage. By noise reduction, the error percentage is further reduced. Compared with the prediction error of NWS (before noise reduction), the performance gain is 26.1% for AUCKLAND IV and 34.4% for AUCKLAND II. Table 1 also shows that noise reduction does not benefit NWS. This is probably due to the internal limitation of NWS in learning complex application patterns.



(a) AUCKLAND IV



(b) AUCKLAND II

Figure 8. Performance comparison of ANN and NWS after noise reduction

Table 1. Performance comparison after and before noise reduction

Prediction Model	AUCKLAND	Original Prediction	Prediction analyzing the traffic composition	
			Before noise reduction	After noise reduction
ANN	IV	12.16%	11.33%	10.74 %
	II	15.63%	14.65%	13.48%
NWS	IV	13.55%	13.66%	13.88%
	II	18.12%	18.33%	18.48%

In the above experiments, we consider UDP and ICMP data as noise because the TCP is the dominant constituent of the network traffic, around 95% in AUCKLAND trace files. When UDP and ICMP

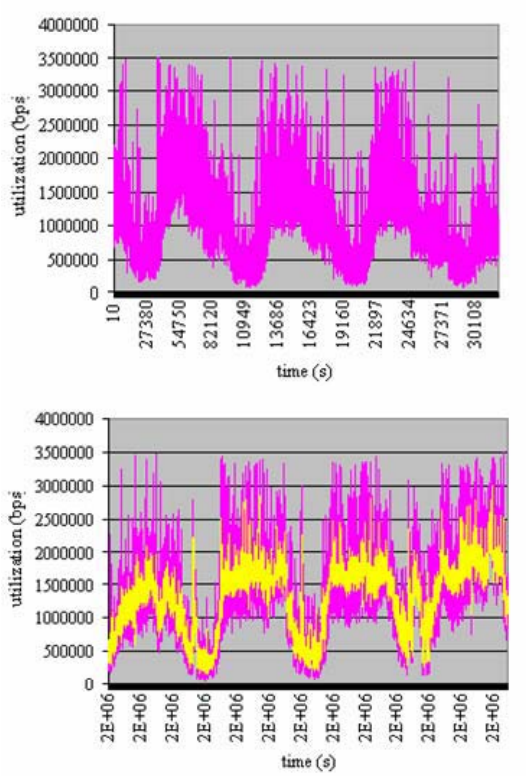
contribute nontrivial constituents of the overall traffic, individual neural networks need to be built and combined into the overall traffic prediction. Please notice that we use the prediction of TCP to predict AUCKLAND. If the trace file is based on one application, the prediction can be even more accurate. This demonstrates the learning ability of the ANN.

4.2 Traffic prediction example

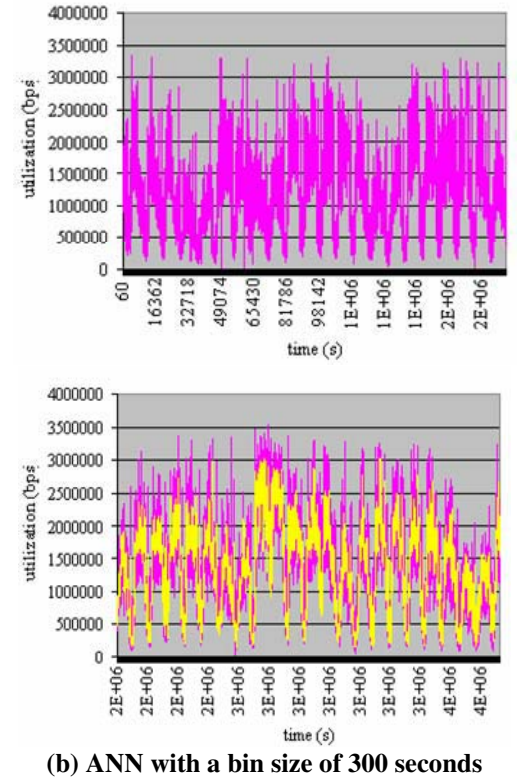
As described earlier, predictors are trained using the first half of the trace data, while predictions are made on the second half. Figure 9 shows the traffic prediction on AUCKLAND IV trace. The curve shown in pink is the training data and the one in yellow is the ANN prediction. We can see that the ANN prediction does capture the communication pattern well.

4.3 The cost of the ANN approach

The cost of a predictive system is very important for its delivery in practice. Because the ANN approach builds a non-linear model for network traffic prediction, its cost in general is greater than those prediction systems that use linear prediction models. Is the cost of the ANN approach reasonable? To answer this question, we tested the ANN cost in both training phase and prediction phase. In general, the neural network requires a large training input data for better performance. We have examined the performance of neural networks for the prediction of available bandwidth with respect to the time complexity on the training process. The experiments were conducted using the different training input size to measure the ANN prediction and training cost on a Pentium IV, 2.6GHz, 512 MB RAM on Windows XP. In general, the size of traffic data binned at lower bin size is larger. Table 2 presents the time taken for training the neural network for various bin sized network traffic data files for the AUCKLAND IV traces. We can see that the training time is linear with respect to the training sample size. When the sample size is 32921, the training time is 3 minutes. The cost is reasonable because the training frequency is far lower than prediction frequency. In our experiment, the training only happened once. At run time the training process of a larger sample space could be done in parallel to network performance prediction so that a longer training process would not affect the timeliness of performance prediction. The neural network prediction is much faster. It takes *1.5* milliseconds to make a prediction.



(a) ANN with a bin size of 60 seconds



(b) ANN with a bin size of 300 seconds

Figure 9. Comparison with trace data

Table 2. ANN Cost

Bin Size (s)	60	100	300	600	900	1800
Training Sample Size	32921	19752	6582	3290	2192	1095
Training time (s)	185	110	37	20	12	8

5. Run-time Prediction

Using pre-measured trace files, we have demonstrated the potential of the ANN approach for network bandwidth prediction. The next question is how to extend the ANN approach so that it can provide run-time performance prediction. Trace information can be collected at run-time. For instance, at each router in the end-to-end path, the DAG card can be fitted to measure the network traffic and the bandwidth data file can be created using the statistics of network traffic data algorithm described in Section 3. Then, the ANN prediction approach can be applied. This trace-based passive measurement approach uses actual observed traffic data without perturbing the network, but requires the access to the routers. The ANN based prediction is not limited to passive measurements. It can use active probe to collect its training data. We have used the input data of NWS to compare the performance of the ANN mechanism and NWS. We consider the traffic data measured between different source and destination pairs at the University of California at Santa Barbara for our experimentation [27]. We perform one-step predictions with the bin size of 60 seconds. Table 3 provides the comparison results. It is evident that the ANN approach provides better accuracy. The ANN mechanism can be used with the NWS monitor subsystem or be incorporated into the NWS system as alternative prediction functionality to provide improved performance prediction at run-time.

Table 3. Run time Comparison of the ANN with NWS

	blind.cs.ucsb.edu to cartman.cs.ucsb.edu	blondie.cs.ucsb.edu to joplin.cs.ucsb.edu
ANN	14.48%	10.15%
NWS	20.08%	15.45%

6. Conclusion and Future Work

In this paper, we have proposed an Artificial Neural Network (ANN) predictive mechanism for network bandwidth availability estimation. The proposed ANN prediction approach has been tested on long-term real-world network traffic traces and with NWS performance monitoring data. To verify the efficiency of the proposed prediction system, experiments are then conducted to compare the prediction errors of the proposed ANN approach with that of the well-known network prediction system NWS on the AUCKLAND II and IV traces. We made predictions by building individual neural networks for each type of network traffic and integrating to obtain the overall predictions. We also categorized the noise and performed predictions after noise reduction. Finally, we investigated the feasibility of the ANN method in providing runtime prediction and compared the performance of the ANN approach and NWS with monitoring data collected by NWS itself. Experimental results indicate that the ANN approach can accurately capture complicated network traffic pattern efficiently. It exhibits a noticeably improved performance over that of NWS. In addition, the ANN prediction mechanism can learn a communication pattern, if one exists, and can provide an even better performance results, as illustrated with different network applications. That demonstrates the potential of ANN in supporting application-level performance predictions. The ANN prediction only takes 1.5 milliseconds in our experiments. When the number of training samples is 3000, the training cost is less than 20 second. The proposed ANN prediction mechanism is feasible and practical. It can use the NWS monitoring system for network bandwidth prediction or be incorporated into NWS or other performance systems for an improved/alternative network prediction. Our current work mainly focuses on predicting network performance with the ANN mechanism.

Consider a user requesting a file from some servers. The file has to be transferred to the user following the transport protocols like TCP. TCP attempts to dynamically and adaptively search for the maximum possible rate using techniques such as slow start or congestion avoidance, which often lead to network underutilization and low application throughput. Our proposed neural network model avoids these problems, thus enabling the transport protocol and applications to achieve higher throughput and react faster to changing network conditions. In the future, we plan to develop an ANN based bandwidth predictive system and integrate the predictive system with new performance measurement approaches [3, 8] to provide real-time on-line network performance prediction.

The ANN mechanism is a complement of the computing model proposed in [18] and will be used in GHS [19] for performance prediction and task scheduling.

Acknowledgments

This research was supported in part by national science foundation under NSF grant CNS-0406328, SCI-0504291, ANI-0123930, and EIA-0224377.

References

[1] I. Foster and C. Kesselman, “The Grid: Blueprint for a New Computing Infrastructure”, Second Edition, Morgan-Kaufman, 2003.

[2] K. Lai and M. Baker, "Measuring Bandwidth." *Proceedings of IEEE INFOCOM '99*, Mar. 1999.

[3] M. Jain and C. Dovrolis, “End-end Available Bandwidth: Measurement Methodology, dynamics, and relation with TCP Throughput”, *Networking, IEEE/ACM Transactions*, Vol. 11, Issue 4, pp. 537-549, Aug. 2003.

[4] K. C. Claffy and C. Dovrolis, “DOE SCiDAC Proposal: Bandwidth estimation: measurement methodologies and applications”, Jul. 2001.

[5] GGF Network Measurements Working Group, “A Hierarchy of Network Measurement Tool Properties”, “http://www-didc.lbl.gov/NMWG/NM-WG-tools.html”.

[6] R.S.Prasad, M.Murray and K.C.Claffy, “Bandwidth Estimation: metrics, measurement techniques, and tools”, *IEEE Network*, Nov. – Dec. 2003 issue.

[7] M. Jain and C. Dovrolis, “Pathload: A measurement tool for end-to-end available bandwidth”, ftp://ftp.ee.lbl.gov/pathchar/, Apr. 1997.

[8] N. Hu and P. Steenkiste, “Evaluation and characterization of available bandwidth probing techniques”, *IEEE journal on selected areas in communications*, Vol. 21, No. 6, Aug. 2003.

[9] F. Berman, R. Wolski, S. Figueira, J. Schopf and G. Shao, “Application-level scheduling on distributed heterogeneous networks,” in the Proc. of Supercomputing'96, Pittsburgh, PA, Nov. 1996.

[10] R. Wolski, N. T. Spring, and J. Hayes, “The network weather service: a distributed resource performance forecasting service for metacomputing,” *Journal of Future Generation Computing Systems*, Vol. 15, No. 5-6, pp. 757-768, Oct. 1999.

[11] D. Hammerstrom, “Neural Networks at Work”, *IEEE Spectrum*, pp. 26-32, Jun. 1993.

[12] D. Park et al., “Electrical Load Forecasting Using An Artificial Neural Network”, *IEEE Trans. On Power Systems*, Vol. 6, No.2, pp. 442-449, May 1991.

[13] A.Hiramatsu, “ATM Communications Network Control by Neural Networks”, *IEEE Trans on Neural Networks*, Vol. 1, No. 1, pp. 122-130, Mar. 1990.

[14] A.Hiramatsu, “Integration of ATM call Admission Control and Link Capacity Control by Distributed Neural Networks,” *IEEE. J. Select. Areas in Communications*, Vol.9, No. 7, pp. 1131-1138, Sep1991.

[15] Y. Liang, "Real-Time VBR Video Traffic Prediction for Dynamic Bandwidth Allocation," *IEEE Transactions on Systems, Man and Cybernetics, Part C*, Vol. 34, No. 2, pp. 32-47, 2004.

[16] H. Yousefi'zadeh, "Neural Network Modeling of Self-Similar Teletraffic Patterns," Invited Paper, *In Proceedings of the First Workshop on Fractals and Self-Similarity, The 8-th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Jul. 2002.

[17] Y. Qiao and P. Dinda, “Network Traffic Analysis, Classification, and Prediction”, Technical Report NWU-CS-02-11, Department of Computer Science, Northwestern University, Jan., 2003

[18] L. Gong, X.H. Sun, and E. Waston, “Performance Modeling and Prediction of Non-Dedicated Network Computing” in *IEEE Trans. on Computers*, Vol 51, No 9, pp. 1041-1055, Sep., 2002.

[19] X.H. Sun and M. Wu, “Grid Harvest Service: A System for Long-Term, Application-Level Task Scheduling,” in the *Proc. of 2003 IEEE International Parallel and Distributed Processing Symposium* (IPDPS 2003), Nice, France, Apr., 2003.

[20] Weka: The University of Waikato. Machine Learning Software in Java, “http://www.cs.waikato.ac.nz/ml/weka/”

[21] M..Minsky and S.A.Papert, “Perceptrons:AnIntroductiontoComputational Geometry”, MITPress, Cambridge, MA, expandededition, 1988/1969.

[22] The Internet Traffic Archive, “http://ita.ee.lbl.gov/index.html”

[23] NLANR Passive Measurement and Analysis, “http://pma.nlanr.net/PMA/”

[24] NLANR Measurement and Network Analysis NLANR PMA: Special Traces Archive, “http://pma.nlanr.net/Special/index.html”

[25] DAG documentation and support, “http://dag.cs.waikato.ac.nz/dag/dag-docs.html”

[26] Waikato Applied Network Dynamics group, “DAG”, http://dag.cs.waikato.ac.nz/

[27] NWS Time Series Query, “http://nws.cs.ucsb.edu/CGI/graphIt.cgi”