NETWORK ADAPTIVE BEHAVIOUR AND ITS OPTIMAL PERFORMANCE

Akhigbe – Mudu Thursday Ehis

Department of Computer Science, Federal University of Agriculture Abeokuta. Nigeria.

ABSTRACT: Adaptation is a fundamental characteristic of living organisms since they attempt to maintain physiological equilibrium in the midst of changing environmental conditions. An approach to the design of adaptive systems is to consider the adaptive aspects of human or animal behavior and to develop systems which behave somewhat analogously. Hence, good performance of networks requires the design of effective control strategies that are capable of rapid adaptation to new circumstances. We designed a frame work algorithms to carry out real – time network traffic, capture the traffic through a sample run of this algorithms. The level of performance is measured by the mean value of temperatures using an ANOVA statistics and the measurements were used to display a performance graph. It is observed that network performance suffers starvation at various levels of environmental changes, and this affects Quality of Service. Our approach is adjudged an efficient approach for network adaptive measures.
KEYWORDS: adaptation, performance, optimal, behavior, environmental and temperature

1. INTRODUCTION

Adaptation is a fundamental characteristic of living organisms since they attempt to maintain physiological equilibrium in the midst of changing environmental conditions. All organisms have adaptations that help them survive and thrive. Some adaptations are structural (anatomical), behavioral or physiological. Anatomical adaptations are physical features of organisms like the fur on a bear or the shape of an animal. Behavioral adaptations are the things organisms do to survive, they can be inherited or learnt and it includes tools, language and swarming behavior. [Dob08]. Autonomic communications systems must demonstrate behavior that remains correct under a range of environmental conditions. An approach to the design of adaptive systems is then to consider the adaptive aspects of human or animal behavior and to develop systems which behave somewhat analogously. In order to gain confidence that a system will behave as intended, it is advantageous to have a formal description of the expected behavior that can be analyzed and tested for compliance with different stimuli [CZ10]. In this paper, we propose an adaptive systems perspective on a recently proposed formal analytical model of network performance. We suggest how this can be used to study the adaptive behavior of systems and to ensure that their adaptations maintain desired properties. If system parameters vary widely according to environmental changes, however, then the control system may exhibit satisfactory response for one environmental condition but may fail to provide satisfactory performance under other conditions. In such case, large variations of system parameters may cause instability. Generally, all mathematically tractable performance indexes have one serious drawback in common, though they specify the cost of system operation in terms of error and energy, they do not give us information about the transient response characteristics of the system [Oga86]. Thus a system that is designed to operate optimally may have undesirable transient characteristics or may even be unstable. On account of these factors, conventional analytical modeling of networks becomes a tedious task. Yet, the good performance of networks requires the design of effective control strategies that are capable of rapid adaptation to new circumstances. This implies not only a lot of processing time, but also a huge amount of memory to maintain the past records. We therefore, need to have an efficient learning paradigm that economizes both time and space of network management resources [G+11]. Human operator recognizes familiar inputs and can use his past learned experiences in order to react in an optimal manner. A learning system is a higher level system to human operator in any adaptive conditions, it is a system which is capable of recognizing the familiar features and patterns of a situation and which uses its past learned experiences in building an optimal fashion.

2. LITERATURE REVIEW

Adaptive system usually implies that the system is capable of accommodating unpredictable environmental changes, whether these changes arise within the system or external to it [Dob08]. The authors in [LLS03], presented a framework for specifying adaptive policies for the management of network services. Some issues in their work need to be addressed. The problem of adaptive management system rather than manual intervention to adapt network configuration is a big problem. Possible problems are that the management system may oscillates or fail to deliver a feasible network configuration. Policies themselves need to be managed and adaptation can itself be specified and enforced by other policies. [BLV98] proposed network adaptive TCP slow starter to provide a mechanism for performing the transfer of small files much more efficiently than is currently done with TCP. The work here relied on previous network performance history as well as the size of the data in an effort to speed up the file transfer. There are a number of things to be improved upon in this work. First of it all, is a lot of the parameters for the round trip times are untested. Secondly, many assumptions have been made for the purpose of simulation, going from well connected local domain to another across a WAN of unknown characteristics [W+06]. The rapid growth of wireless communications networks has put tremendous pressure on network's resources. As a result, efficient adaptive systems are required for effective resource management which constitutes a challenging task [MP13].

3. METHODOLOGY

We designed a framework algorithms called Packet Monitor Algorithms (see figure 1) to carry out a real time network management analysis. Traffic is captured through a sample run of this algorithm displayed in figure 2. The output window is then processed into a sequence of M traffic length, the sequence of length n (M) = 30 is again subdivided into three segments $(g_1, g_2.and.g_3)$ respectively. $n(g_1)=10, n(g_2)=10.and.n(g_3)=10$. Analysis of Variance (ANOVA) is used here to test the designed hypothesis, the data in figure 2 shows the data generated, as χ_i of units of performance rate when the process was operating at each of the three atmospheric temperature levels. Test of significant used is $\alpha = 0.05$. The level of performance is measured by the mean value and $\bar{X}_i$ indicates the observed traffic mean at level i where $i = 1, 2.and.3$ corresponding to temperatures $68^0 F, 72^0 F.and.76^0 F$ respectively. There is a certain amount of variations among these means, since sample means do not necessarily repeat when repeated samples are taken from a population, some variations can be expected .The question here is; Is this variations among the $\bar{X}_i$'s due to chance or is it due to a difference in the traffic rate at each temperature level? The null hypothesis that we test is: $(H_0 : \mu_{68} = \mu_{72} = \mu_{76})$. That is, the true performance mean is the same at each temperature tested. In other words, the temperature does not have a significant effect on the rate of performance. And the alternative to this is: $(H_a : \mu_{68} \neq \mu_{72} \neq \mu_{76})$, not all means are equal. Thus we shall reject the null hypothesis if the data show that one or more of the means is significantly different from the others. The decision to reject or fail to reject H_0 will be made using the F – distribution and the F – test statistics. That is, we will compare the calculated value of F, F^*, to a one tailed critical value of F obtained from statistical table of appendix G [Joh76].

3.1. Packet Monitor Algorithms

(A) Algorithm for Port Scanning

Step 1: Variable Declaration
Declare variables for storing IP Address and host name and set them to null
Step 2: Input
2.1 Enter value of Host name (or IP Address)
Step3: Scanning
3.1 Declare variable port = 0
3.2 Declare initial port = value.
3.3 Declare final port = value.
3.4 Check if the port is available between initial port and final port.
3.5 increment port by 1
3.6 Repeat step 3.4 up to final port.
Step4: Display
4.1 Display all the active ports in GUI format
(B) Algorithm for packet capturing
Step 1: Obtaining the list of network interfaces
1.1 Create a variable array of devices
1.2 Detect network interfaces present in user
1.3 Store the above list in devices variable.
Step 2: Displaying the list of network interfaces
Declare loop counter integer variable i and initialize to 0
While the value of i is less than the length of the array of devices, do Step 2.3
Print out the name and description of the captured Network Interface.
Step 3: Open the network interface.
Declare integer variable J and initialize to zero (J=0)
While J < length of array of devices, Goto Step 3.3 else Goto Step 3.7
Check if the network interface at Jth index number in devices array is selected. If yes goto Step 3.6 else goto Step 3.4
J=J+1 Goto Step 3.2

Open the selected network interface i.e. Network Interface at Jth index, then Goto Step 4
Display that the network interface has not yet been selected by the user. he user. Goto Step 8
Step 4: Capture packets from the network interface
Is the menu button of stop capture packet selected? If yes goto Step 3.8 else goto Step 4.2
Capture the upcoming single packet from the network

Display the captured packet by going to Step 5
Step 5: Display the captured packet to the user in proper GUI format.
Detect user' menu choice of the format in which captured packet's to be displayed

Analyze the packet. Display in Hexadecimal format
Goto Step 6 to save the packets to a temporary file
Go back to Step 4.1
Step 6: Save captured packets into a file
6.1 Create a temporary file say
6.2 Save captured packets into the opened file
6.3 Go back to Step 5.4
Step 7: Close all the open network interface
7.1 Delete the temporary file.
7.2 Close the network interface.
Step 8: End

Figure 1: packet monitor algorithm

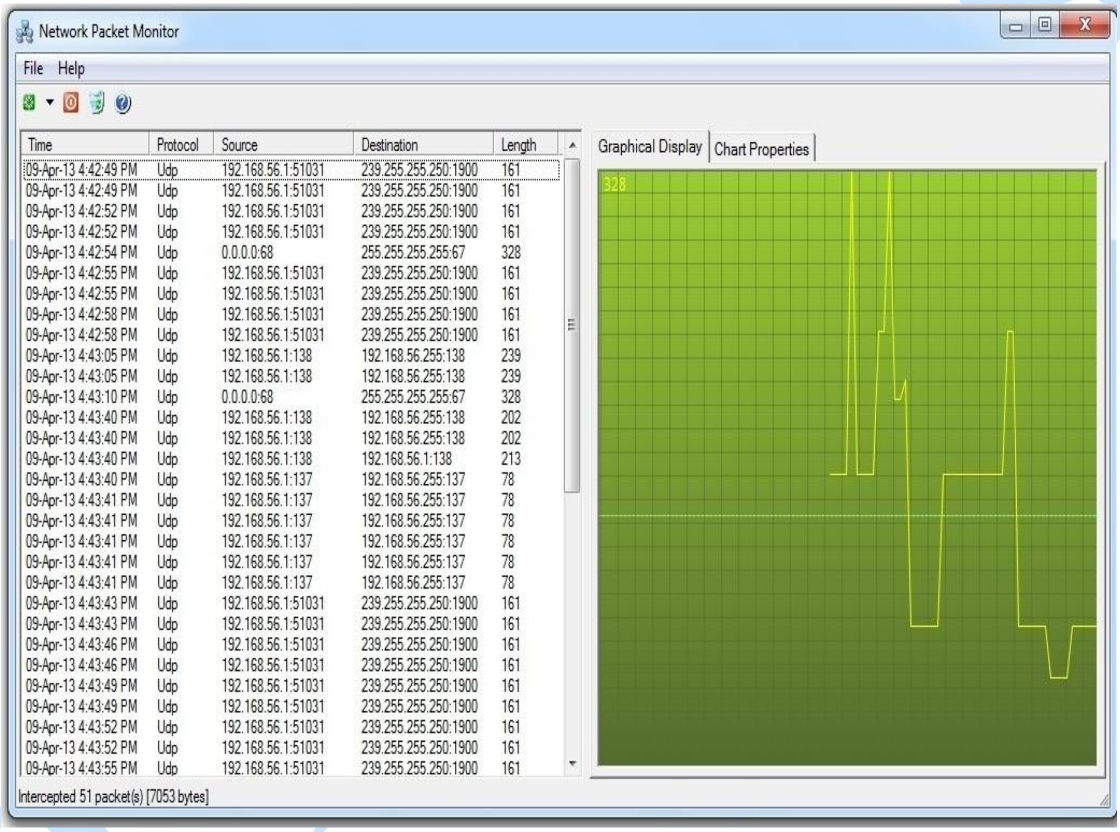


Figure 2: Output Window of Packet monitor Algorithms

4. PERFORMANCE IMPLEMENTATION

Temp level $(\bar{X}_i)$	Replicate (j)										Row (T_i)	Mean $(\bar{X}_i)$
	1	2	3	4	5	6	7	8	9	10		
68^0F	161	161	161	161	328	161	161	161	161	239		
72^0F	239	328	202	202	213	78	78	78	78	78		
76^0F	78	161	161	161	161	161	161	161	161	161		

Figure 3: Sequence of Data from the Output window

Before performing the calculations on these numbers, let's code the data by dividing each data shown in the preceding table by 20. Since ANOVA technique use only measures of variation in the decision making process, coding the original data by dividing by a convenient number will not change the results but will make the arithmetic considerably easier.

Temp (i)	Replication(j)										Row (T_i)	Mean $\bar{X}$
	1	2	3	4	5	6	7	8	9	10		
68^0F	8.05	8.05	8.05	8.05	16.4	8.05	8.05	8.05	8.05	11.95	92.75	9.28
72^0F	11.95	16.4	10.1	10.1	10.65	3.9	3.9	3.9	3.9	3.9	78.7	7.8
76^0F	3.9	8.05	8.05	8.05	8.05	8.05	8.05	8.05	8.05	8.05	76.35	7.6

Figure 4: A coded Sequence of Data from the Output Window

The analysis of variance (ANOVA) procedure will separate the variations among the entire set of data into two categories. This separation is accomplished by first working the numerator of the fraction that is used to define sample variance $S^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}$.

This numerator is called sum of squares:

$$\text{Sum of squares} = \sum_{i=1}^{n}(X_i - \bar{X})^2 \quad (1)$$

$$SS(total) = \sum_{i=1}^{n} X_i^2 - \frac{\left(\sum_{i=1}^{n} X_i\right)^2}{n} \quad (2)$$

The SS (total) for our illustration is now found using formula (2).

$$\sum_{i=1}^{n} X_i^2 = 2322.95 \quad \text{and} \quad \left(\sum_{i=1}^{n} X_i\right)^2 = 61{,}404.84$$

Thus $SS(total) = ?$ must now be separated into two parts, sum of squares, SS(temp), due to the temperature levels, and sum of squares, SS(error), due to experimental error of replication. The sum of SS (temp) and SS(error) = SS(total), hence this splitting is often referred to as partitioning.

$$SS(factor) = \frac{\sum(T_i^2)}{c} - \frac{\left(\sum_{i=1}^{n} X_i\right)^2}{n} \quad (3)$$

where T_i row totals and C are the number of replicates.

$$\sum(T_i)^2 = 20625.575$$

and

$$SS(temp) = \frac{\sum(T_i)^2}{c} - \frac{\left(\sum_{i=1}^{n} X_i\right)^2}{n} = 15.76$$

The sum of squares SS (error), which measures the variation within the rows, is found by use of:

$$SS(error) = \sum_{i=1}^{n} X_i^2 - \frac{\sum(T_i^2)}{c} \quad (4)$$

The SS(error) for our illustration is found by use of formula (4)
For convenience we shall use an ANOVA table to record the sum of squares and organize the rest of the calculations. The degree of freedom, df, associated with each of the three rows are determined as follows: $SS(error) = 260.4$

1. df(temp) is I less than the number of levels at which the factor is tested:

$$df(temp) = r - 1 \quad (5)$$

2. df(total) is 1 less than the total number of pieces of data:

$$df(total) = n - 1 \quad (6)$$

3. df(error) is the sum of the degree of freedom for all the rows. Each row has c-1 degrees of freedom; therefore:

$$df(error) = r(c-1) \quad (7)$$

The sums of squares and the degrees of freedom must both check. That is

SS(temp) + SS(error) = SS(total)
.and.
SS(temp)+ df(error) = df(total)

The mean square, MS (factor) for the factor being tested, MS (error) will be obtained by dividing the sum of squares value by corresponding number of degrees of freedom. The resulting table of values is shown in figure 5.

Source	SS	df	MS
Temp	15.76	2	7.9
Error	260.4	27	9.6
Total	275.8	29	

Figure 5: ANOVA table

df(temp) = 3 -1 = 2,
df (error) = r (c-1) = 27
and
df(total) = n -1 = 30 – 1 = 29

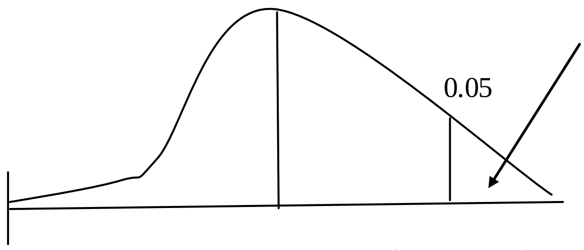
$$MS(temp) = \frac{SS(temp)}{df(temp)} = 7.9$$

MS(temp) by MS(error).

The hypothesis test is now completed, using the two mean squares as measures of variance. The calculated value of the test statistics F, F^* is found by dividing $MS(temp).by.MS(error)$.

$$F^* = \frac{MS(temp)}{MS(error)} = 0.78$$

The decision to reject or fail to reject H_0 will be made by comparing this calculated value of $F, F^* = 0.78$ to a one tailed critical value of “F” obtained from table 8a of appendix G [Joh76].
Fail to reject H_0 .



$$F(2,27,0.05) = 3.39$$

Figure 6: Showing the Critical Region

4.10 Performance Evaluation

The applicability of our approach to adaptive network system behavior has been tested under different temperature levels and these measurements are used to display a performance graph (see figure 7). The concept of a control system optimization comprises a selection of a performance index which will minimize some measures of a deviation from the ideal behavior. Our propose approach is adjudged good and efficient for network adaptive behavioral measures.

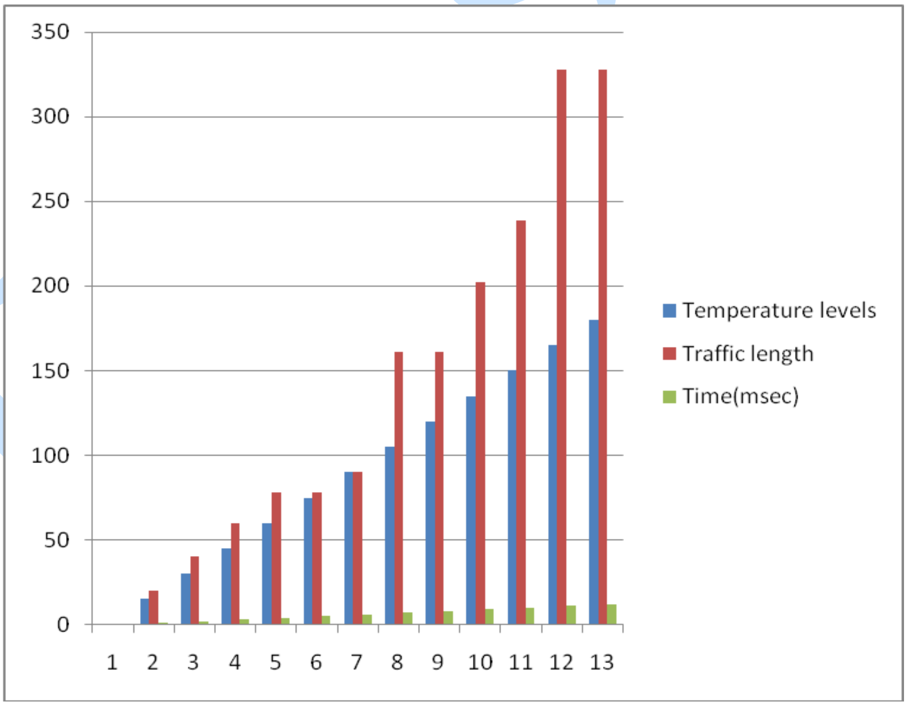


Figure 7: Network Adaptive Behavior

5. CONCLUSION AND FUTURE WORK

Adaptive system implies that the system is capable of accommodating unpredictable environmental changes, whether these changes arise within the system or external to it. This concept has a great deal of appeal to the system designers since a highly adaptive system, besides accommodating environmental, changes would also accommodate moderate engineering design errors or uncertainties and would compensate for the failure of minor system components, thereby increasing system reliability. In this paper, we designed frame work algorithms to carry out real – time traffic, capture the traffic through a sample run of these algorithms.

The level of performance is measured by the mean value of temperatures using an ANOVA statistics and the measurements are used to display a performance graph. The concept of a control system optimization comprises a selection of a performance index which will minimize some measures of a deviation from ideal behavior. A fail to reject H_0 decision is interpreted as the conclusion as there is no evidence of a difference due to the level of the tested factors. In other words, the temperature does not have a significant effect on the rate of network performance. In most control systems, small deviation in parameter values from their design values will not cause any problem in the normal operation of the system, provided these parameters are inside the loop. If system parameters vary widely according to environmental changes, the system may exhibit satisfactory response for one environmental condition but may fail to provide satisfactory performance under other conditions. Our propose approach is therefore adjudged an effective design for network adaptive behavioral measures. The solution to an optimal control problem may not exist if the system considered is not controllable and observable. Then it is necessary to know the condition under which a system is controllable and observable and this is a reference for future work.

REFERENCES

[BLV98] **Yogesh Bhumralkar, Jeng Lung, Pravin Varaiya** - *Network Adaptive TCP slow Start,* University of California, Berkeley, (1998).

[CZ10] **W. Chen, Z. Zhang** - *Globally Stable Adaptive Back stepping Fuzzy Control for Output Feedback Systems with unknown High-Frequency Gain Sign,* Fuzzy Sete and Systems, Vol. 161, No, 6, pp 821 – 836, (2010)

[Dob08] **Simon Dobson** - *An Adaptive System Perspective on Network Calculus, with Applications to Autonomic Control,* International Journal of Autonomous and Adaptive Communications System,(IJAACS) Volume 1, Number 3, (2008)

[G+11] **Joao V. Gomes, Pedro R. M. Inacio, Manela Pereira, Mario M. Freire, Paulo P. Monteiro** - *Identification of Peer – To – Peer VoIP Sessions Using Entropy and Codec Properties.* IEEE Transactions on Parallel and Distributed Systems, Volume x, No. x, (2011).

[Joh76] **Robert R. Johnson** - *Elementary Statistics, Duxbury* Press, North Scituate, Massachusette, Pages 449 – 470 (1976)

[LLS03] **Leorunas Lymberopoulos, Emil Lupu, Morris Sloman** - *An Adaptive Policy Based Framework for Network Services Management,* Journal of Network and Systems Management, Special Issues on Policy Based Management, 11:3, pages 277 – 303, (2003)

[MP13] **Lepora N. Martinez, T. Prescott** - *Active Touch for Robust Perception under Position Uncertainty,* IEEE Proceedings of ICRA (2013).

[Oga86] **Katsuhiko Ogata** - *Modern Control Engineering,* Prentice Hall of India, Private limited, New Delhi, Pages 750 – 796 (1986)

[W+06] **C. Wang, D. J. Hills, S. S. Ge, G. Chen** - *An ISS – Modular Approach for Adaptive Neural Control of Pure Feedback Systems,* Automatica Vol. 42, No. 50, pp. 723 – 731, (2006)