International Journal on Recent and Innovation Trends in Computing and Communication
Volume: 3 Issue: 4
ISSN: 2321-8169
2437 - 2439

Network Bandwidth Simulation using Monte Carlo Principle

Bhushan Gehi
Student of IInd Year M.E Computer Science,
K.J. Somaiya College of Engineering,
Vidyavihar, Mumbai.
bhushan.gehi@gmail.com

***Abstract* –** Network bandwidth plays an important role in evaluating any application's performance running on a computer. Objective of this paper is to implement a Network bandwidth simulator. This simulator would capture various types of data patterns seen in a typical network and replicate similar patterns in the simulated data feed.

Monte Carlo simulation principle would be used in to implement the data simulator.

*****

I. Introduction

A typical bandwidth usage pattern of a network would be as shown below in fig.1. As can be seen in the graph, we would mostly have certain hours of the network wherein the bandwidth usage is high. On the contrary we might also have certain hours in which it is not used at all or least used.

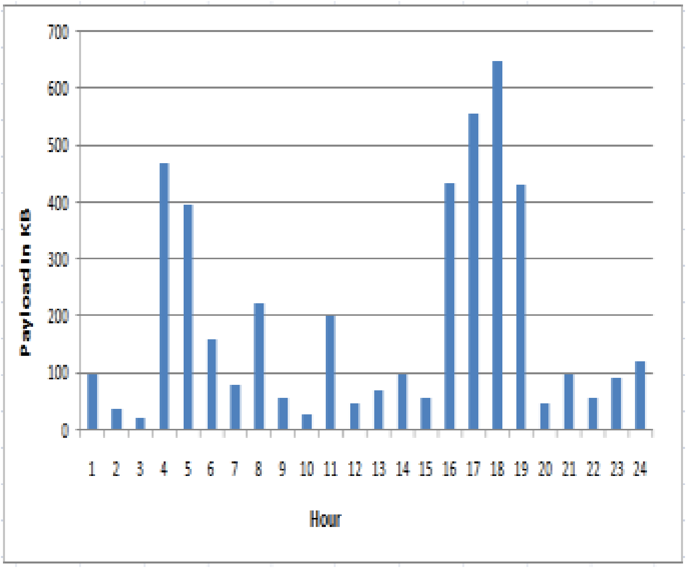

Fig. 1.1 Typical usage pattern of a network

In above usage pattern there are certain payloads which are periodic or have a certain cycle, like database refresh cycles, software update cycles, certain files being copied from remote servers, etc. However we also have random events of day to day work cycle which cause payloads to occur.

Monte Carlo simulation is a computerized mathematical technique that allows people to account for risk in quantitative analysis and decision making. Monte Carlo simulation furnishes the decision-maker with a range of possible outcomes and the probabilities they will occur for any choice of action. It shows the extreme possibilities—the outcomes of going for broke and for the most conservative decision—along with all possible consequences for middle-of-the-road decisions.

Monte Carlo simulation performs risk analysis by building models of possible results by substituting a range of values—a probability distribution—for any factor that has inherent uncertainty. It then calculates results over and over, each time using a different set of random values from the probability functions. Depending upon the number of uncertainties and the ranges specified for them, a Monte Carlo simulation could involve thousands or tens of thousands of recalculations before it is complete. Monte Carlo simulation produces distributions of possible outcome values. By using probability distributions, variables can have different probabilities of different outcomes occurring. Probability distributions are a much more realistic way of describing uncertainty in variables of a risk analysis. Monte Carlo methods vary, but tend to follow a particular pattern:

- Define a domain of possible inputs.
- Generate inputs randomly from a probability distribution over the domain.
- Perform a deterministic computation on the inputs.
- Aggregate the results.

We would follow above principle to implement our data simulator.

II. IMPLEMENTING THE SIMULATOR

To start implementing data simulator, we start by analyzing a sample of actual data set captured from live network. Below shown is a subset of actual dataset used for our study:

Table 2.1 Subset of sample data

Node	Date	Weekday	Hour	Payload (KB)
Node1	4/2/2014	Tuesday	0	2432
Node2	4/2/2014	Tuesday	2	424
Node3	6/2/2014	Thursday	3	535
Node3	6/2/2014	Thursday	4	466
Node1	4/2/2014	Tuesday	2	7987
Node6	4/2/2014	Tuesday	9	977
Node1	6/2/2014	Thursday	20	677
Node1	6/2/2014	Thursday	19	5578
Node1	6/2/2014	Thursday	17	858
Node5	6/2/2014	Thursday	8	568
Node6	4/2/2014	Tuesday	11	8688

In above sample data, we have bandwidth payload captured across various nodes of the network, for different days of the week and summarized at hour level. If we analyze above data in details, it is found that every payload event is defined by 2 types of events:

- Cyclic Payload
- Randomized Payload

Cyclic payloads are certain payloads which are periodic or have a certain cycle, like database refresh cycles, software update cycles, certain files being copied from remote servers, etc. Randomized payload are instantaneous payload events not characterized by any cyclic event like any user opening certain websites or download a audio file, etc.

We also implement our data simulator by above 2 types of payload. We identity the cyclic nature events from our sample dataset and introduce such cyclic events in our data simulator. If a particular event doesn't follow any cyclic nature, we use randomization to generate payload for that event. Same can be seen in the Code snippet shown in Fig 2.1.

```
private int getPayLoad(NodeData newNode) {

    Random r = new Random();

    int cyclicPayload = getCyclicPayload(newNode);

    if(cyclicPayload != -1)
        return cyclicPayload;

    if(newNode.getWeekday() < 5)
        // return payload between 0 and 200
        return r.nextInt(200);
    else
        // return payload between 0 and 100
        return r.nextInt(100);
}


private int getCyclicPayload(NodeData newNode) {
    // Below are the cyclicPayload for respective days
    Random r = new Random();
    int hour = newNode.getHour();
    int day = newNode.getWeekday();

    // Monday system logon dependencies update
    if(day ==1 && (hour > 9 && hour <12))
        // return payload between 200 and 500
        return r.nextInt(300)+200;
    //Working hour weekday range 100 - 300
    if(day <5 && (hour > 9 && hour <18))
        // return payload between 100 and 300
        return r.nextInt(200)+100;
    if(hour == 2)
        // OS and DB update everyday between 2-3 am
        return r.nextInt(500)+500;
    if(day >5 && (hour > 20 && hour <22))
        // Weekly system update on weekends
        return r.nextInt(1000)+500;
    return -1;
}
```

Fig.2.1 Code snippet of Monte Carlo simulator

As shown in above snippet, we use randomization principle from Monte Carlo simulation and then overlay cyclic or periodic nature to calculate the payload value.

Also, as can be seen above, for both type of data events we have defined certain payload range within which value will be picked. For example: Every Monday between 9 to 12 am we logon server dependencies are updated, hence it falls under cyclic event. This dependencies cause an average of 300 KB payload, hence payload for this hour will be between 200 and 500.

III. Testing and Result Analysis

Below shown is a sample dataset simulated using our custom data generator.

Table 3.1 Output of Feed Simulator

Node	WeekDay	Hour	Payload
1	0	0	19
1	0	1	158
1	0	2	796
1	0	3	56
1	0	4	116
1	0	5	37
1	0	6	146
1	0	7	98
1	0	8	178
1	0	9	170
1	0	10	106
1	0	11	281
1	0	12	107
1	0	13	158
1	0	14	196
1	0	15	270
1	0	16	272
1	0	17	140
1	0	18	179
1	0	19	131
1	0	20	181
1	0	21	178
1	0	22	58

Below shown is a comparison of actual data Vs the data generated by the simulator.

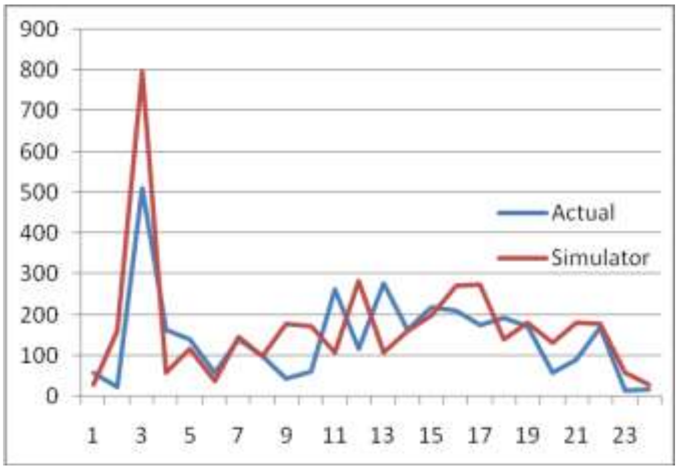

Fig. 3.1 Actual Dataset Vs Simulator Generated Data

IV. Conclusion and Future Work

As seen from Fig 3.1, the simulator algorithm does a fair job in generating values for most of the instances. However there might be few time instances where it could be improved. For such instances, there could be few more network parameters which should be captured. Also, simulation algorithm for events of network which are cyclic in nature should be continuously updated as per changing nature of cyclic events in network bandwidth.

V. References

[1] Keleher, P. ; Bhattacharjee, B. ; Sussman, A. Decentralized, accurate, and low-cost network bandwidth prediction. Dept. of Computer Science, Univ. of Maryland, College Park, MD, USA. INFOCOM, 2011 Proceedings IEEE

[2]Alaknantha Eswaradass, Xian-He Sun, Ming Wu. Network Bandwidth Predictor (NBP): A System for Online Network performance Forecasting. Department of Computer Science Illinois Institute of Technology Chicago, Illinois 60616, USA

[3] "Monte Carlo Simulation" http://www.palisade.com/risk/monte_carlo_simulation.asp

[4] "Network packet" http://en.wikipedia.org/wiki/Network_packet