Summer Research

Pauline Dang

September 2022

1 Introduction

Network congestion refers to the reduction of service quality, packet loss, queuing delay, and poor connections. This paper looks at the different symptoms of network congestion and different approaches to characterizing, inferring, and predicting network congestion. Previous approaches have often define network congestion as an increase of RTT. However, the approach that was focused primarily on was the increase of jitter, which correlates to increased RTT as a more accurate identifier of congestion. Jitter is defined as the variation of RTT, and could be more insightful for more accurate congestion prediction.

2 Background Work

The first resource that was consulted heavily was Inferring Persistent Interdomain Congestion [1]. In this paper, they took the approach of inferring congestion based on increased RTT. Increased latency is one of the most classic symptoms of network congestion and is often a result of it. This approach found success in inferring recurrent congestion and some success at one of congestion. However, it has its own limitations in router queuing behavior, incompleteness, and unknown courses of congestion.

The other one was Jitterbug: A new framework for jitter-based congestion [2] inference where they proposed that approach of defining network congestion based off of jitter, jitter dispersion and latency times. They found great accuracy in being able to inferring not one recurrent congestion, but one-off congestion as well. While this was an interesting and insightful paper on inferring congestion, the focus of this internship is finding a way to infer network congestion in real time and even predict network congestion.

3 Jitterbug

The Jitterbug programs takes in an input of an RTT dataset. From that point it does through a series of filtering, tests, algorithms, and change point detection algorithms in order to infer congestion.

3.1 Signal Filtering

The initial RTT data set returns multiple latency times at the same time set, but at inconsistent time intervals. This is due to the fact that when measuring RTT using pings and probes, there is a high change of packet drops and many pings do not return at all. Furthermore, there can be too much data to filter through which makes it more difficult to see significant changes.

To combat this, the minimum latency of every 15-minute time interval is taken. this produces a min-RTT time series that the jitterbug program also analyzes. It gives the data uniformity and actually allows the program to run.

From that point, the program is able to compute the jitter based off of the raw RTT time series and the min RTT time series. This produces two data sets, jitter and j-min.

In order to actually calculate the jitter dispersion, the program puts the j-min time series through a Moving IQR and the Moving Average filters in order to compute jitter dispersion.

3.2 Interval Detection

The jitterbug program uses and Bayesian Change Point [3] algorithm in order to segment time intervals within the data. The Bayesian Change Point algorithm was the most effective at detecting the boundaries of intervals with RTT latency measurements in the data.

A standard Bayesian approach is based off of Adam and McKay's Bayesian Online Changepoint Detection [3]. It estimates the posterior distribution of the current run length in regards to how long it has been since the last changepoint. It compares the current data to the past data and updates and compares the changepoint data as it is received in real time. The algorithm can result in two possibilities: the current run time can continue as the value has not move drastically outside the current regime's parameters or the run length can drop to 0 as a new changepoint has been detected.

4 ARIMA

The ARIMA model is a time series prediction model that extends two other fundamental time series models, the Auto regressive model (AR) and Moving Average model (MA).

A time series is a representation of data where the dependent variable is time. For example, if there was a data set representing ice cream sales, there are a few different ways to represent sales. One version could have the x axis represent the temperature, with ice cream sales increasing as temperature increasing. One could look at what the temperature is tomorrow and find the correlating sales number on a graph like this. However, in a time series, the dependent variable would be time. In this case, if the time intervals is segmented into months, taken over years, it would likely show that ice cream sales peak in June and

July. Using this data for prediction is there the previously mentioned models come in.

The Auto Regressive Model is a linear combination of statistically significant time intervals and the value of these historical time intervals to make forecasts.

$$X_t = \beta_0 + \beta_1 X_{t-1} + \beta_2 X_{t-2} + ... + E_t \tag{1}$$

X_t represents the value at the next time interval, or the prediction. β_n represents the coefficient that the time lag, and is found using the partial auto correlation function. E_t represents the error and how much it was off is at that value.

5 Congestion Prediction

The jitterbug program was made available to the public via GitHub. With it it was provided an example data set of raw RTT and min RTT. the Jitterbug program was able to infer congestion from the data and was represented by a series on 1s and 0s that represents that congestion inferred and not inferred, respectively.

Using the min time series, I ran a time series program I wrote earlier in the summer. The following figure plots the example data of RTT in December 2017: Here, we can see that network congestion has some obvious recurrence to it on a daily basis. To see if my timer series and AR model code was capable of predicting latency times. To do so, I section off the data until December 15, for adequate historical data. From there I used my AR model to run on the remaining days and compared it to the actual data to test for accuracy.

In my first attempt I limited the historical data to the past 50 time intervals. As shown, there is some semblance of latency in the beginning, but overall, the prediction was widely inaccurate and this data cannot really be used.

For my second attempt, used used a wider parameter for the training data, which was that I used the past 100 time intervals. I again compared it to the actual data and is shown in the following figure:

This model was far more accurate to the actually data. We can see in the beginning especially, that it was extremely accurate. Due to the nature of time series, the accuracy decreased the further ahead it was trying to predict but there is a strong correlation between the two.

6 Conclusion

Throughout the course of my research, I studied a lot about inferring and predicting network congestion. My plan is to take the time series predicted data and run the jitterbug program to infer congestion. However some issues that arise with this approach are the following: Raw RTT: Due to time series needing uniform time intervals, it is impossible to predict the raw time series as the time in between the latency times are sporadic. However jitter is computed off of both raw data and filtered data. Computation time and resources: When running the time series code on the example data set, there was a huge strain on the computer I was running it on. To run the code with high accuracy and in real time would require much more advanced software. Despite the constraints, I think this could be a very lucrative approach to prediction traffic congestion. The jitter and jitter dispersion give much more insight to congestion as opposed to the traditional approach which is looking at just the latency times.

References

[1] A. Dhamdhere, D. D. Clark, A. Gamero-Garrido, M. Luckie, R. K. Mok, G. Akiwate, K. Gogia, V. Bajpai, A. C. Snoeren, and K. Claffy, "Inferring persistent interdomain congestion," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 1–15.

[2] E. Carisimo, R. K. Mok, D. D. Clark, and K. C. Claffy, "Jitterbug: A new framework for jitter-based congestion inference," in *International Conference on Passive and Active Network Measurement*. Springer, 2022, pp. 155–179.

[3] R. P. Adams and D. J. C. MacKay, "Bayesian online changepoint detection," 2007. [Online]. Available: https://arxiv.org/abs/0710.3742