CONSERVATIVE CASCADES: AN INVARIANT OF INTERNET TRAFFIC

Steve Uhlig

Department of Computer Science and Engineering
Université catholique de Louvain, Louvain-la-neuve, Belgium
E-mail: suh@info.ucl.ac.be
URL: `http://www.info.ucl.ac.be/~suh/`

ABSTRACT

Conservative cascades are among the class of multiscaling processes that accomodate the two competing objectives of deterministic and random cascades. This paper discusses in what sense the cascade model can be seen as an invariant of the Internet traffic. It explains why conservative cascades model the time-invariant properties of the Internet traffic at timescales up to hundreds of milliseconds. Furthermore, it show the limitations of conservative cascades to model part of the time-dependent scaling properties of Internet traffic.

1. INTRODUCTION

The last decade has been a very fruitful period in important developments in network traffic modeling, uncovering a broad spectrum of scaling behaviors [1]. Self-similarity [2], multiscaling and multifractal behavior [3], and finally cascades [4, 5] have been studied and convincingly matched to real traffic.

Several papers have applied the cascade model to Internet traffic [3, 4, 5]. However, no work had focused yet on understanding the limits of the applicability of the cascade model. In this paper, we confirm the rationale for this model. Our main contribution is to show that the cascade model is an invariant of the traffic for timescales ranging between a few milliseconds and a few hundreds of milliseconds, timescales controlled by the behavior of TCP. By invariant, we mean that the generality of the model allows to capture the behavior of the Internet traffic at these timescales. However, invariant also means insensitivity to potentially important properties of the traffic. In this paper, we explain why conservative cascades is an invariant of Internet traffic, but we also show that it fails to capture other important properties of the traffic, like multifractality and second-order non-stationarity that depend on the conditions that the traffic undergoes when crossing the network topology.

2. SCALING, MULTISCALING AND CASCADES

Scaling behavior is a broad term connected with the absence of a particular characteristic scale controlling the process under study. Among scaling processes, *self-similarity* of parameter H ($0 < H < 1$) is defined by

$$\{X(t), t \in \mathbb{R}\} \stackrel{d}{=} \{c^H X(t/c), t \in \mathbb{R}\}, \forall c > 0, \quad (1)$$

where $\stackrel{d}{=}$ denotes equivalence in finite distributions. A property of self-similar processes concerns the scaling of their moments

$$E|X(t)|^q = E|X(1)|^q \, |t|^{qH} . \quad (2)$$

This latter property expresses the behavior of moment q of the process as a power law of time, defining a high-order scaling which is linear in H.

Multiscaling is a relaxation of strict scaling where the linearity relation among moments does not hold anymore but can vary for each moment q [5]:

- self-similarity : $E|d(j,k)|^q = C_q(2^j)^{qH}$,
- multiscaling : $E|d(j,k)|^q = C_q(2^j)^{H(q)}$,
- infinitely divisible cascade : $E|d(j,k)|^q = C_q e^{H(q)n(2^j)}$.

In the equations above, $d(j,k)$ represent the increments of the process at time k and for timescale j (increasing values of j refer to larger timescales). Strict self-similarity implies that all moments behave like a power-law of the timescale j, where the power-law is linear with respect to H. Multiscaling on the other hand allows the power-law relationship between moments to depend on the moment q. For instance, a special type of multiscaling is multifractality where the moments tend to diverge relative to one another for the smallest timescales ($H(q)$ increases with q), indicating local irregularities in the process.

Another type of departure from monoscaling is infinitely divisible cascades (IDC) [5] that generalize multifractality. One has no power-law relationship between scales and moments anymore but there are two quantities of interest $H(q)$ and $n(a)$ linked by the folowing relationship :

$$\ln S(q,j) = -H(q)\, n(a) + K_q \quad (3)$$

$$= \frac{H(q)}{H(p)} \ln S(p,j) + K_{p,q} \quad (4)$$

where K_q and $K_{p,q}$ are constants, $a = 2^j$ and $S(q,j)$ is the partition function of the process at timescale j for moment q. The difference between multiscaling and IDC can be seen through their respective definition, multiscaling being defined by

$$E|d(j,k)|^q \propto exp(H(q)\ \ln(2^j)) \quad (5)$$

while IDC by

$$E|d(j,k)|^q \propto exp(H(q)\, n(2^j)). \quad (6)$$

The noticeable difference lies within the power-law dependence of the timescale for multiscaling while this is not forcibly true for IDC. However, IDC still maintain the similarity with self-similarity and multiscaling, namely the separability of the moment structure in the moment order and the timescale. Hence, it is still possible to find the rule guiding the behavior of the process through formula 4. IDC is thus a generalization of multiscaling when $n(2^j)$ can be any function of the timescale j, not only a logarithmic function of the timescale. In addition, multifractality can be seen as the limiting object generated by the cascade, decribing the local singularities of the multiplicative process.

The function $H(q)$ describes the generating function of the kernel of the generator of the cascade (one step of the cascade), while $n(a)$ renders the depth of the cascade, i.e. how many times the generator is applied at this timescale. There are many distributions that can participate in an IDC: Gaussian, Poisson and compound Poisson, Gamma, α-stable,... Estimating the two quantities $H(q)$ and $n(a)$ is possible but identifying the exact distribution of the cascade is still an open issue.

The connexion between IDC and network traffic is conceptually simple, as seen through the notion of a *conservative cascade* [4]. A *conservative cascade* is a cascade that accomodates the two competing objectives of deterministic and random cascades: 1) preservation of the total mass of the process at each step of the cascade and 2) randomness of the distribution of the mass among the subintervals. These two properties are intuitively valid for TCP traffic, where there is a random distribution of the mass of the data to be sent over any TCP flow and a deterministic breaking of the TCP segments within a TCP flow due to the deterministic behavior of TCP. The distribution of the packets is hence a mix between the deterministic way with which the TCP protocol distributes the mass of the traffic within a flow, and the randomness induced by the behavior of the network and its users.

3. INTERNET TRAFFIC ANALYSIS

In this paper, we rely on the Auckland-IV data set [6]. This Internet traffic trace is a 45 days long trace captured at the University of Auckland between February and April 2001. The deviation of the timestamps of the trace from Universal Time are less than one microsecond, but we only use a one millisecond precision in this paper. We use in this paper 15 hours of this trace, during which 1,629,069 incoming TCP flows (1,613,976 outgoing) and more than 13 GBytes of incoming traffic (10 GBytes outgoing) were seen. The variable we study is the total amount of bytes seen for every one millisecond time interval. Note that we rely on wavelet-based estimators but due to space limitations we do not present them but refer to [7]. The smallest timescale j shown by the wavelet analysis of this section is octave 1 corresponding to a timescale of 2 ms, each successive octave representing a timescale two times larger.

We first study the partition function, as shown on top of Figure 1. The partition function indicates that the high-order moments contain valuable information of the dynamics of the process since the $\ln(S(q,j))$ are far larger for large values of q. Such a high-order moments scaling justifies the applicability of the models described in section 2. Three significant regions appear on the graphs: the first for octaves smaller than 8, the second between octaves 8 and 12, and the third for octaves larger than 12. The octaves between 8 and 12 correspond to timescales between 256 ms and a few seconds, where the cascade model starts not to be valid anymore. TCP breaks the information into segments that are sent over the network, and in order to adjust its sending rate TCP relies on an estimation of the bandwidth available on the path. TCP adjusts its sending rate on timescales of the order of the typical delay between the time it sends some segment and the time this segment is acknowledged by the destination host. A timescale of a few hundred milliseconds is hence typical of a few times this two-way delay where the cascade behavior of TCP is not the one leading the traffic dynamics. The incoming traffic for octaves smaller than 8 exhibits a multifractal behavior with the moments of the partition function that diverge with respect to one another, while outgoing traffic seems to contain only simple multiscaling. Octaves larger than 12 seem closer to self-similarity than multiscaling for both traffic directions. Our numerical estimations confirm the graphical impression that the outgoing traffic is not multifractal while the incoming traffic is, for octaves smaller than 8.

When trying to detect IDC behavior, the linear regression indicates that the outgoing traffic is consistent with the IDC for octaves up to 8 while the incoming traffic is very rarely consistent with the IDC model. The reason for this behavior is that outgoing traffic leaves the local network while the incoming comes from the rest of the Internet, hence incoming traffic has suffered from the perturbations of the cross-traffic along the path to attain the local network. To understand this effect of crossing the topology, we use a graphical tool introduced in [8] that displays in three dimensions the evolution with time of the logscale diagram. The *logscale diagram* [7] computes the sample variance of the wavelet coefficients at each timescale. The logscale diagram thus shows how the variance of the process gets distributed among the various timescales. On the middle part of Figure 1, we display the evolution in time of the logscale diagram computed over subintervals of the whole trace for both traffic directions. The graph of the outgoing traffic presents a neat and homogeneous logscale diagram that does not change much with time. The graph of the incoming traffic on the other hand not only has a more variable graph at the large timescales (large octaves) but seem to have a breakpoint near octave 8. The behavior of the logscale diagram at the small timescales explains the multifractality of the incoming traffic for octaves smaller than 8. Multifractality implies local irregularities of the process for small timescales, and the logscale diagram for the incoming traffic is more flat than the one of the outgoing traffic. This means that the relative variability of the small timescales for incoming traffic is more important than the one of the outgoing traffic, hence there are more irregularities in the incoming traffic at the small timescales. The changes in the network conditions on the topology that the incoming traffic has to cross explain why only the incoming traffic exhibits this behavior.

Now let us deal with the parameters of the cascade $H(q)$ and $n(a)$. We do not provide figures for the estimation of $H(q)$ due to space limitations. Both traffic directions have almost the same value of $\widehat{H}(q) \in [1.5,2]$ but with large 95 % confidence intervals that prevent us to draw any conclusion. Yet we can say that the two traffic directions have a similar estimated $H(q)$, which was expected since the network conditions undergone by the traffic should not change this part of the cascade (the kernel of the generator) that depends on how TCP breaks the data into segments. This also means that it should be possible to diagnose the behavior of TCP by tracking the $H(q)$ part of the cascade. A misbehaving TCP source (or a group thereof) could be detected by comparing the kernel of its (their) generator with the typical behavior of a TCP source. The estimation of $n(a)$ illustrates once more the difference between the two traffic directions (bottom of Figure 1). While the outgoing traffic shows a relatively stationary behavior, the incoming traffic on the other hand exhibits large fluctuations, consistently with the behavior of the partition function and the logscale diagram above. Furthermore, only octave 1 exhibits this behavior indicating that this non-stationarity is not due to the behavior of the cascade itself but to changes in the network conditions. The parameters controling the cascade are thus time-invariant, and the traffic properties not captured by the cascade model are caused by the time-dependent network conditions that the incoming traffic undergoes on the network topology.

4. CONCLUSION

This paper explained why the conservative cascade model can be considered as an invariant of Internet traffic. We showed that timescales between a few milliseconds and a few hundred of milliseconds are consistent with a conservative cascade model. We explained from a networking perspective why this model is sound at these timescales. We also showed the limitations of this model, by comparing traffic leaving a local network from traffic entering a local network. We showed that the incoming traffic, undergoing the variable network conditions, had multifractal and second-order non-stationary behavior that was not well captured by the cascade model.

5. REFERENCES

[1] P. Abry, R. Baraniuk, P. Flandrin, R. Riedi, and D. Veitch, "The multiscale nature of network traffic: Discovery, analysis, and modelling," *IEEE Signal Processing Magazine*, May 2002.

[2] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of ethernet traffic (extended version)," IEEE/ACM *Transactions on Networking*, 1994.

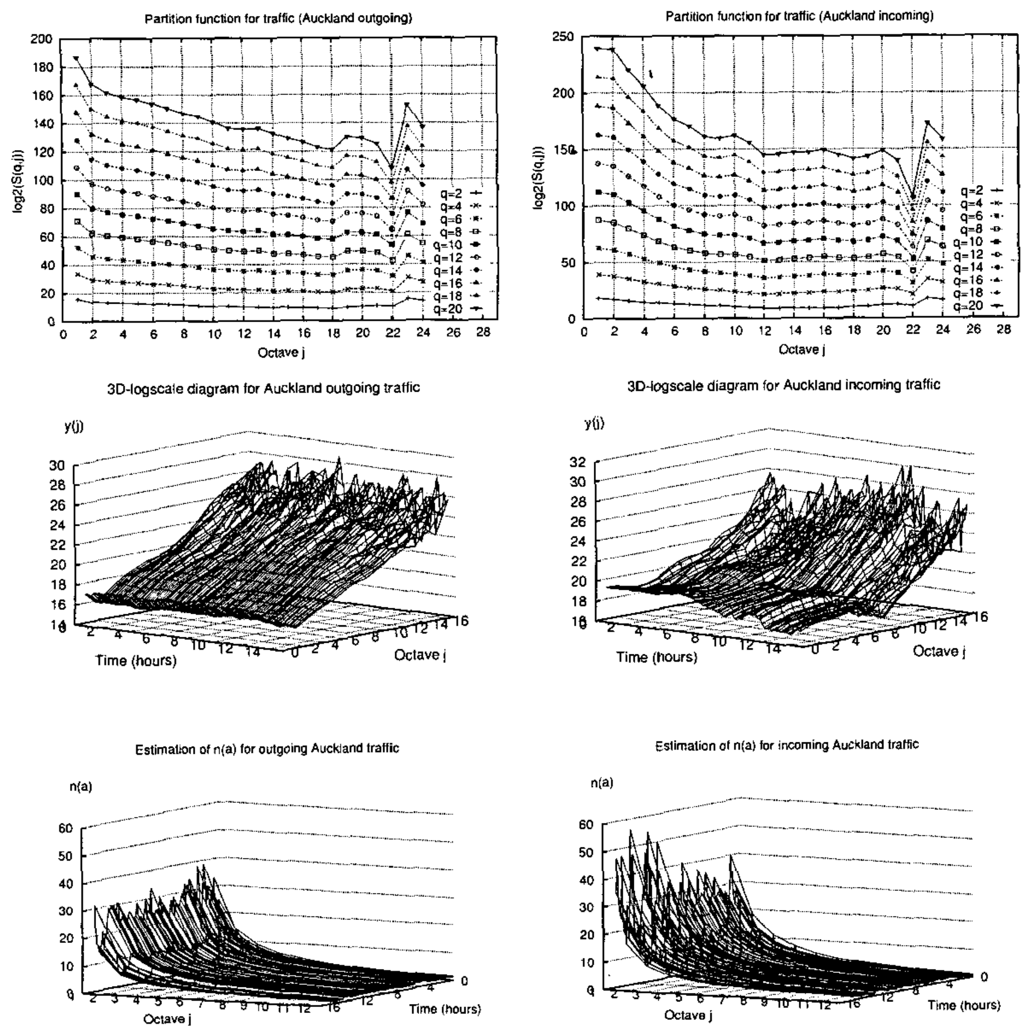

Fig. 1. Partition function (top), logscale diagram (middle) and n(a) parameter of the cascade (bottom).

[3] A. Feldmann, A. Gilbert, and W. Willinger, "Data networks as cascades: Investigating the multifractal nature of internet WAN traffic," in *Proc. of ACM SIGCOMM'98*, 1998.

[4] A. Gilbert, W. Willinger, and A. Feldmann, "Scaling analysis of conservative cascades, with applications to network traffic," *IEEE Trans. on Information Theory*, vol. 45, no. 3, pp. 971–992, 1999.

[5] D. Veitch, P. Abry, P. Flandrin, and P. Chainais, "Infinitely divisible cascade analysis of network traffic data," in *Proc. of ICASSP*, June 2000.

[6] J. Org, M. Ian, and G. Brownlee, "The auckland data set: an access link observed," in *Proceedings of the 14th ITC Specialist Seminar*, April 2001.

[7] P. Abry, P. Flandrin, M. Taqqu, and D. Veitch, "Wavelets for the analysis, estimation, and synthesis of scaling data," *In Park and Willinger (eds) "Self-Similar Network Traffic and Performance Evaluation"*, 2000.

[8] S. Uhlig, O. Bonaventure, and C. Rapier, "3D-LD : a graphical wavelet-based method for analyzing scaling processes," in *Proc. of the 15th ITC Specialist Seminar*, July 2002.