A Novel Self-Similarity ($S^2$) Traffic Filter to Enhance the Success of E-Business

WILFRED W. K. LIN, ALLAN K.Y. WONG, THARAM S. DILLON and WONG HOK LEUNG

1Department of Computing, Hong Kong Polytechnic University, Hong Kong SAR, Fax: (852) 27740842
2Faculty of Information Technology, University of Technology, Sydney, Australia

Abstract: The novel self-similarity ($S^2$) filter is proposed to identify self-similar Internet traffic patterns on the fly. The previous real-time traffic pattern detector (RTPD) does not have this identification capability. The “RTPD + $S^2$” combination is the enhanced RTPD or ERTPD. Since the longest ERPD execution time with $S^2$ is only 1455 clock cycles, it is still highly suitable for real-time applications. For example, it can be incorporated as part of a dynamic buffer tuner so that the latter can use the detected traffic pattern to fine-tune the control process. In this way $S^2$ contributes to enhance the success of e-business by shortening the client/server service roundtrip time more effectively. This is achieved because it supports more accurate dynamic buffer tuning.

Key-Words: $S^2$ filter, e-business, Internet traffic, ERTPD, dynamic buffer tuning

1 Introduction

Although mobile or electronic businesses (e-business) may take many forms, they fit the general framework of Internet-based distributed computing. In light of this framework clients and e-business centres (i.e. e-shops) interact in both wireless and wireline fashions. Wireless interaction permits client mobility, and the communication cell, which supports this interaction, is referred to as the smart space [Garlan02]. A mobile client continues to interact on the go with the same e-shop while moving from one smart space to another. Wireline interaction requires cable based communication. Figure 1 captures the essence of an e-shop operation, which is exemplified by those that are selling shoes in Mainland China. A shoes e-shop consists of several basic elements: a virtual show room, shoes manufacturing at the back, a remote order service (ROS), and a list of friendly collaborators that can supply the needed commodities in a timely fashion. If a customer cannot find the right pair of shoes in the current e-shop’s virtual show room, she/he provides the specification for the e-shop to custom-make it or finds it from its collaborators. The key to success for any e-shop is a fast response to make customers happy.

A significant threat to fast service response (i.e. short service roundtrip time (RTT) in the client/server interaction) is widespread retransmission requests by clients. If the communication channel in Figure 1 has an error probability $\rho$, the average number of trials (ANT) to get a successful transmission is defined by $ANT = \sum_{j=1}^{\infty} jP_j \approx \frac{1}{(1-\rho)}$, where $P_j$ the probability of success at the $j^{th}$ trial. One factor that enlarges $\rho$ and thus AN is the e-shop buffer overflow due to the unpredictable nature of the incoming request streams that merge (i.e. represented $\oplus$ in Figure 1) at the service access point (SAP) [Wong02]. The unpredictability is caused by the changing traffic patterns of the request traffic (i.e. inter-arrival time or IAT) streams. Any sudden burst of short IATs could cause surges in the queue length leading to overflow of its buffer. In fact, postmortems or off-line analyses show that Internet traffic traces can be short-range dependence (SRD) (e.g. Markovian type) or long-range dependence (e.g. self-similar or heavy-tailed) [Molnar99]. One way to eliminate buffer overflow at the server level (e.g. e-shop) effectively is to make the buffer length always cover the queue length adaptively [Dillon99]. This approach is called dynamic buffer tuning [Wong02], which needs to be achieved statistically. However, using any preconceived mathematical model (e.g. Poisson) would fail for the Internet traffic pattern changes continuously [Paxson95].

So far, there is little published experience for on-line traffic pattern detection and analysis. The aim is to sample traffic on the fly and decide if the current IAT aggregate is SRD and LRD. The real-time traffic pattern detector (RTPD) proposed in [Lin04b] can effectively differentiate SRD from LRD, but it cannot identify whether the LRD

Figure 1. Client/server interaction of an e-shop
pattern is self-similar or heavy-tailed. This identification, however, is useful because self-similar and heavy-tailed patterns produce different degrees of instability in the dynamic buffer tuning process [Lin04a] and therefore require different compensation measures. For this reason the novel self-similarity ($S^2$) filter, which works with the “continuous aggregate based (CAB)” mechanism, is proposed in this paper. The “RTPD + $S^2$” combination is called the **enhanced RTPD** or simply (ERTPD). If the dynamic buffer tuner in an e-shop includes ERTPD as support, then it can leverage the traffic pattern identified by the latter to compensate and adjust the tuning process on the fly.

### 2 Related Work

The Internet’s sheer size and heterogeneity naturally involve many different protocols in client/server interactions [Lewandowski98]. As a result the Internet traffic follows the power law [Medina00], and over time the traffic may change suddenly from LRD (long-range dependence) to SRD (short-range dependence) or vice versa [Willinger03]. Using the Hurst (H) effect as the yardstick [Taquq03], LRD is within the range of $0.5 < H < 1$ and $0 < H < 0.5$ for SRD [Molnar99].

If $X^m = \{X^m_i : i \geq 1\}$ is a time series aggregate of size $m$ in a stochastic process $X$, its autocorrelation function (ACF) is $r^m(l) = \sum_{i=1}^{m} X_i X_{i+l}$. $r^m$ is the autocorrelation of $X^m$ and $l$ the aggregate level. The ACF of LRD traffic is non-summable, $\sum_{i=1}^{\infty} r^m(k) \approx \infty$, but in contrast the SRD’s is summable, $\sum_{i=1}^{\infty} r^m(k) < \infty$.

The previous RTPD differentiates SRD from LRD in a stationary process. Its core is the traditional R/S (rescaled adjusted statistics) approach, which is popular for off-line applications. The R/S is enhanced into a real-time “$M^3RT + R/S + filtration$” RTPD. The $M^3RT$ module is a micro Convergence Algorithm (CA) or MCA implementation [Wong01], which predicts the mean of any waveform quickly and accurately on the fly. It operates as a logical object to be invoked for service anytime and anywhere by message passing. In effect, the MCA changes the traditional R/S to its enhanced form (i.e. E-R/S). The filtration process activates the appropriate filter to identify the exact traffic pattern. For example, the **modified QQ-plot filter** identifies heavy-tailed distributions. The $S^2$ filter lets the filtration process identify self-similarity for the first time.

The CA operation, which is based on the **Central Limit Theorem**, is represented by equations: (2.1) and (2.2). The estimated mean $M_j$ in the $i^{th}$ prediction cycle is based on the fixed $F$ (flush limit) number of data samples. The cycle time therefore depends on the delay for collecting the $F$ samples. It was confirmed previously that $M_j$ has the fastest convergence for $F=14$ [Wong01]. The other parameters include: a) $M_{i-1}$ is the feedback of the last predicted mean to the current $M_i$ prediction cycle, b) $m^j_i$ is the $j^{th}$ data item sampled in the current $i^{th}$ $M_j$ cycle for $j = 1,2,3,...,(F-1)$, and c) $M_0$ is the first data sample when the MCA had first started. In the E-R/S, $M_j$ replaces $\overline{X}$ and yields $W_i = \sum_{m=1}^{i} (X^m_m - M_j)$. This replacement makes the E-R/S more suitable for real-time applications than the traditional R/S approach because the number of data items (e.g. IAT) to calculate $W_i$ is now predictable (i.e. $F = 14$). In an ERTPD implementation, E-R/S, $M^3RT$, and the invoked filter module are running in parallel. The overall ERTPD execution time depends on the module that has the longest execution. **Intel’s VTune Performance Analyzer** [VTune] recorded the following average execution times in clock cycles: 981 for E-R/S, 250 for $M^3RT$, and 520 for the **modified QQ-plot filter**. In the verification exercise the novel $S^2$ filter needed an average of 1455 clock cycles to execute. Therefore, the overall ERTPD execution time is affected by $S^2$ because it has the longest execution time compared to the other components such as $M^3RT$.

### 3 The Novel Self-similarity Filter

The self-similarity ($S^2$) filter identifies self-similar patterns on the fly. LRD traffic has at least two fractal components: heavy-tailed and self-similar. The
The self-similar nature of many fractal point processes comes from the heavy-tailed distributions, for example, the FRP (Fractal Renewal Process) inter-arrival times. The heavy-tailed property, however, is not a necessary condition for self-similarity because at least the FSNDPP (Fractal-Noise-Driven Poisson Process) does not possess any heavy-tailed property at all. The theoretical foundation for the novel $S^2$ filter is the “asymptotically second-order self-similarity” concept, which is simply called hereafter the statistical $2^{nd}$ OSS (or $S^2$ OSS ). This concept is associated with a sufficiently large aggregate level or log $l$ in a stochastic process $X$. If $X^m = \{X_i^m : I \geq 1\}$ is an aggregate in $X$ of size $m$, having $S^2$ OSS for a large enough $m$ means that the associated autocorrelation function (ACF), namely $r^m(l)$ (for $X^m$) is proportional to $l^{-(2-2H)}$.

The $S^2$ OSS is LRD because the ACF is non-summable, namely, $r^m(l) = \sum_{i=1}^{l} r^m = \infty$. The condition “$r^m(l) \propto l^{-(2-2H)}$” for a large $m$” is mathematically equivalent to the slowly decaying variance property. For this property the variance of the mean for sample size $m$ decays more slowly than $m$ (i.e. $\text{Var}(X^m) \propto m^{-\beta}$). A $2^{nd}$ OSS process $X$ and $0.5 < H < 1$ the value of $\beta = 2 - 2H$ should apply [Molnar99].

The equations (3.1) and (3.2) summarize the $S^2$ OSS property and they hold even for the weaker condition shown by equation (3.3). The slowly decaying variance property becomes conspicuous if a log-log plot is produced for equation (3.1) as shown by equation (3.4).

$\log(\text{Var}(X^m)) = \log(\text{Var}(X)) - \beta \log(m)$. (3.4)

As an implementation issue the sampling of data points for an aggregate should start with a statistically reasonable value, for example, $m = 30$. If the sampled data points do not show stationarity, then another 30 samples should be collected so that the next round of stationarity test should be based on $2m$ amount of data points. This process, which is called the “continuous aggregate based (CAB)” mechanism, repeats until the stationarity is confirmed. The stationarity/Gaussianity test is based on the “kurtosis and skewness (KS)” technique. The kurtosis and skewness values together indicate if an aggregate is stationary. The R/S and $S^2$ elements in the ERTPD mechanisms only work for stationary conditions. The normal bell/Gaussian curve, which represents an ideal stationary process, has kurtosis and skewness equal to 3 and 0 respectively. This ideal [3,0] pair is difficult to obtain statistically, but previous empirical experience [Lin05b] shows that stationarity is valid for the [ 0.0, 6.0 ] combinations. Skewness measures a distribution’s symmetry. A distribution skews to the right for positive skewness and to the left for a negative one. Higher positive kurtosis than 3 means a more “peaked” distribution while a negative kurtosis indicates a “flatter” one. Skewness is given as $\frac{\sum_{i=1}^{N}(x_i - \bar{x})^3}{(N-1)s_d^3}$, where $\bar{x}$ is the mean, $s_d$ the standard deviation and $N$ the number of data points in the aggregate, and kurtosis is $\frac{\sum_{i=1}^{N}(x_i - \bar{x})^4}{(N-1)s_d^4}$.

Figure 2. The “continuous aggregate based (CAB)” approach

The CAB process to fit the straight line with a slope $-\beta$ for each aggregate $X^m$ is based on the slowly decaying variance property. It starts only when the aggregate has satisfied the Gaussianity/stationarity test [Sarlotham01].
$S^2$ finds $\beta$ (equation (3.4)) by linear regression, which has its quality indicated by the coefficient of determination or $R^2$ between 0 and 1 [Jain92]. The higher the $R^2$ value the better quality is the fit. It uses the predefined threshold, $Th_{R^2}$ (e.g. 0.85 or 85%) to reject the possibility of a self-similar pattern in $X^m$ for $R^2 < Th_{R^2}$. The CAB mechanism is illustrated in Figure 2, where it works for many aggregates $X^m_{Ag-i}$. Assuming: a) P1, P2, and P3 are the log-log plots with respect to equation (3.4), b) they yield the corresponding $\beta$ values: $\beta_1$ for P1 with $R^2 = 0.82$, $\beta_2$ for P2 with $R^2 = 0.98$, and $\beta_3$ for P3 with $R^2 = 0.95$, c) $Ag = I$ is the aggregate level, and d) $Th_{R^2} = 0.9$, then both P2 and P3 indicate self-similar traffic but not P1 (for $R^2 < Th_{R^2}$). If P2 and P3 yield different $\beta$ values, they represent two different self-similar patterns. Therefore the data segment that includes both P2 and P3 is multifractal [Willinger03]. In contrast, if the $\beta$ values for P2 and P3 are the same, the data segment made up of P2 and P3 is monofractal (i.e. single dimension).

4 Experimental Results

The aim of the simulation experiments is to verify that the proposed $S^2$ filter, which is driven by the CAB mechanism, can indeed detect and identify self-similar Internet traffic patterns on the fly. The set up for the experiment is similar to Figure 1, but “traffic analysis” is implemented as the ERTPD mechanism. The traffic between the client and the SAP is either simulated or by using pre-collected IAT traffic traces. The self-similar patterns were mainly simulated by the Kramer tool [Kramer]. The VTune tool [VTune] was also used to measure the $S^2$ execution times under different conditions. The timing analysis with VTune is important for evaluating the $S^2$ fitness in time-critical applications.

In fact, the VTune results indicate that the $S^2$ filter needs an average 1.455 clock cycles to execute. For a platform running at 100 mega hertz this means a physical time of 1455/(100*10^3) or 14.55 micro seconds. This short execution time makes $S^2$ highly suitable for real-time applications. Table 1 shows one of the results produced by the $S^2$ filter. In this case the filter identifies the self-similar traffic patterns correctly with at least 90% confidence (i.e. the threshold $Th_{R^2}$ was set at 90%). The data segment made up of the first five aggregates is basically monofractal. The aggregates, 6 and 7 are rejected because they do not satisfy the $Th_{R^2}$ threshold. The collection of every aggregate (from 1 to 7) follows the following steps: a) start with the basic aggregate size of $m=30$ data points, b) carry out the Gaussianity test with the KS technique, c) if Gaussianity does not exists then collect $m=30$ more data points and go back to the previous step else find $\beta$. In fact, the experimental results for different real-life Internet traffic traces indicate that the Internet traffic pattern indeed changes over time. This empirical fact is demonstrated here by using the Sony trace [WongTrace]. The relevant experiments indicate that SRD and LRD data segments intertwine/interleave in the Sony trace as shown in Table 2. The basic aggregate size for these experiments was $m=32$, and the percentages of SRD and LRD segments are 81.07% and 17.95% respectively. Table 3 shows that the percentages of self-similar and heavy-tailed patterns for the LRD segments are respectively 76.77% and 85.42%. Out of the 439 LRD segments 289 of them (i.e. 65.83%) show both self-similar and heavy-tailed characteristics. This is normal because self-similarity usually comes from heavy-tailedness, even though the latter is not a necessary condition for the former.

$$S^2 = \sum_{i=1}^{n} (X_i - \bar{X})^2$$

Where $X_i$ are the data points, $\bar{X}$ is the mean of the data points, and $n$ is the number of data points. $S^2$ is a measure of the variability of the data points around the mean. The coefficient of determination $R^2$ is calculated as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}$$

Where $Y_i$ are the observed data points, $\hat{Y}_i$ are the predicted data points, and $\bar{Y}$ is the mean of the observed data points. $R^2$ ranges from 0 to 1, with 1 indicating a perfect fit.

### Table 1. The $S^2$ filter identifies self-similar traffic correctly

<table>
<thead>
<tr>
<th>$\beta$ slope</th>
<th>H (Hurst value), $H = (1 - \beta^2)$ (coefficient of determination)</th>
<th>$R^2$ (aggregates in sequence)</th>
<th>kurtosis</th>
<th>skewness</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6 583</td>
<td>0.671 0.936 (95.6%)</td>
<td>1 0.59 7045 1.18 0.0861</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6 809</td>
<td>0.660 0.975 (97.5%)</td>
<td>2 -0.56 218 0.79 8282</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6 425</td>
<td>0.679 0.977 (97.7%)</td>
<td>3 0.40 215 1.27 7175</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6 473</td>
<td>0.677 0.972 (97.2%)</td>
<td>4 0.38 386 0.86 1215</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4 685</td>
<td>0.766 0.939 (95.9%)</td>
<td>5 -0.58 417 0.89 2037</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3 762</td>
<td>0.812 0.885 (88.5%)</td>
<td>6 -1.01 033 0.44 6756</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1 978</td>
<td>0.901 0.605 (60.5%)</td>
<td>7 -1.16 043 0.38 8599</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 2. $S^2$ results for the Sony trace in light of stationarity, SRD and LRD

<table>
<thead>
<tr>
<th>Total no. of continuous aggregates</th>
<th>Stationarity (%)</th>
<th>SRD and LRD patterns intertwined</th>
</tr>
</thead>
<tbody>
<tr>
<td>2548</td>
<td>2446 (96.00%)</td>
<td>SRD (%) 1983 (81.07%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LRD (%) 439 (17.95%)</td>
</tr>
</tbody>
</table>

Differentiation between self-similar
and heavy-tailed traffic patterns

<table>
<thead>
<tr>
<th>and heavy-tailed traffic patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total number of LRD aggregates (Table 2)</strong></td>
</tr>
<tr>
<td><strong>Self-similar (SS) (%)</strong></td>
</tr>
<tr>
<td><strong>Heavy-tailed (HT) (%)</strong></td>
</tr>
<tr>
<td><strong>Both self-similar and heavy-tailed (overlapped) (%)</strong></td>
</tr>
<tr>
<td><strong>P(SS[HT]) ; both self-similar and heavy-tailed (overlapped)</strong></td>
</tr>
<tr>
<td><strong>P(SS[HT]) ; self-similar but not heavy-tailed</strong></td>
</tr>
</tbody>
</table>

Table 3. $S^2$ results for the Sony trace in light of SRD and LRD differentiation

4.1 Experimenting with the FLC Dynamic Buffer Tuner

The FLC (Fuzzy Logic Controller) is one of the few dynamic buffer tuners in the field [Lin05b]. Figure 3 is one of the FLC designs (i.e. the FLC[4x4] design), and its essence consists of the following:

a) It uses the rate of change of the server queue length, $\frac{dQ}{dt}$ for derivative control.

b) It uses the “queue length over buffer length (QOB)” ratio for proportional control.

c) The dot marks the QOB reference (i.e. $QOB_R$) of 0.8, and X marks the inert “don’t care” state.

d) The control decisions, which depend on the current QOB and $dQ/dt$ values, include: Addition (buffer elongation) or “+”, Subtraction (buffer shrinkage) or “-”, and “don’t care”.

e) The linguistic variables for the FLC are as follows:


ii) For the current $dQ/dt$: NL for Negative and Larger than the given threshold, NS for Negative and Smaller than the given threshold, PS for Positive and Smaller than the given threshold, and PL for Positive and Larger than the given threshold.

With the linguistic variables fuzzy rules for dynamic buffer tuning can be formulated such as follows ($L_{new}$ and $L_{old}$ denote the adjusted buffer length and the old buffer length respectively; ICM is the buffer adjustment):

Rule 1: If (QOB is MG) AND ($dQ/dt$ is PL) Then Action is “+”(Addition) AND $L_{new} = L_{old} + ICM$

Rule 2: If (QOB is ML) AND ($dQ/dt$ is NL) Then Action is “-”(Subtraction) AND $L_{new} = L_{old} - ICM$

Rule 3: If (QOB is L) AND ($dQ/dt$ is NS) Then Action is “X”(Don’t care) AND $L_{new} = L_{old}$

<table>
<thead>
<tr>
<th>QOB</th>
<th>dQ/dt</th>
<th>NL</th>
<th>NS</th>
<th>PS</th>
<th>PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>ML</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>0.8</td>
<td>L</td>
<td>X</td>
<td>X</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>G</td>
<td>-</td>
<td>X</td>
<td>K</td>
<td>+</td>
</tr>
<tr>
<td>MG</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3. The FLC[4x4] design

Figure 4. The calibrated MD versus GP relationship

In this set of experiments the $\frac{dQ}{dt}$ control was adaptively tuned by a gradient percentage (GP) to produce the minimum mean deviation from $QOB_R$ in the dynamic buffer tuning process. The tuning is controlled by the traffic pattern identified by the novel ERTPD mechanism. The mean deviation (MD) is the average deviation from $QOB_R$ in different experiments for the same type of request traffic (e.g. self-similar or Poisson). By a trial and error process the GP versus MD relationship was traced and calibrated as shown in Figure 4. The calibration for the self-similar traffic pattern was made possible by the novel $S^2$ filter. For example, if self-similar traffic is detected the FLC tuner has to use $GP = 8\%$ to produce the same minimum MD of 0.03 as for the Poisson condition it needs only $GP = 5\%$. It is worthwhile to tune the GP value adaptively because a high GP value means longer computation delay.
Figure 5 compares the FLC and “ERTPD + FLC[4x4]” performance for a self-similar trace

Figure 5. Comparing FLC and “ERTPD + FLC[4x4]” performance for a self-similar trace

5 Conclusion

The novel self-similarity ($S^2$) filter proposed in this paper enables self-similar Internet traffic patterns to be identified on the fly. It enhances the previous real-time traffic pattern detector (RTPD), which does not have this capability, to the new ERTPD package. Since the ERTPD execution time is short, it is highly suitable for real-time applications. For example, when it is combined with the FLC dynamic tuner it makes the latter’s tuning process more accurate. As a result there are less oscillations and lower mean deviations (MD) from the reference $QOB_r$ ratio. This is translated into shorter service roundtrip time and more efficient buffer memory usage. The experimental results indicate that $S^2$ indeed identifies self-similar patterns correctly. The next step in the research is to validate ERTPD package and thus $S^2$ in real e-shops.

6. Acknowledgement

The authors thank the Hong Kong PolyU for the A-PG51 research grant.

References:


