Modelling Internet End-to-End Loss Behaviors: A Congestion Control Perspective

Vinh Bui, Weiping Zhu, Ruhul Sarker
The University of New South Wales, Australia
Email. {v.bui,w.zhu,r.sarker}@adfa.edu.au

***Abstract*— This paper proposes a new approach to modelling and controlling Internet end-to-end loss behaviours. Rather than select the model structure from the loss observations as being done previously, we construct a new loss model based on the TCP congestion control mechanisms. Thus, the model can explicitly reflect the correlation between end-to-end loss observations and network flow level activities. Besides simulation, the model has been tested in both wired and wireless Internet environments. The result shows that, unless the losses due to the transmission errors are excessive e.g. in some lossy wireless channels, the model can correctly capture end-to-end loss behaviours not only in terms of average rates but also in terms of loss patterns i.e. loss and good runlengths. This implies a good connection between the model structure and network flow level activities, which makes the model attractive for assisting network traffic control.**

***Index Terms*— Internet Loss Modelling, Network Traffic Control, Markov Model, Markov Decision Processes.**

I. Introduction

Recently, Internet end-to-end traffic control has drawn considerable attentions from the research community due to the limitation of the TCP/IP best-effort mechanism in providing Quality of Service (QoS) [4], [6]. In end-to-end traffic control, end hosts rely on end-to-end packet delays and/or losses to adjust traffic intensity. Therefore, understanding the dynamics of loss and delay behaviours, especially at packet flow level, is essential to the success of any end-to-end approach.

A good way to understand the dynamics of loss and delay behaviours is to mathematically model them. As a result, numerous studies have been carried out on Internet loss and delay modelling, which approach the problem mainly from the Markov modelling point of view due to the temporal correlation between consecutive packets traversing the same path [3], [5], [8], [9], [13], [15], [16], [17], [20], [21].

Although the previously mentioned models have successfully described the loss behaviours to some degree, it is difficult to find a connection between the structures of those models and the network states, which cause the losses. Meanwhile, most end-to-end traffic control mechanisms rely on appropriate models to infer the network internal states from end-to-end observations. Hence, the capability of the mentioned loss models to assist end-to-end control is limited. In addition, since the structure of many existing loss models is empirically selected based on loss observations, different model structures are needed to reflect different network conditions leaving an open question about robustness of those models.

Therefore, in this paper, we propose a new Markov-based loss model to overcome the limitation of the previous models. We construct the model based on the TCP congestion control mechanisms. Thus, the model not only can describe end-to-end loss behaviours but also link them to network flow level activities. We extend the work presented in [2] by providing a comprehensive verification of the proposed model using both simulation with NS-2 [10] and Internet trace-based analysis. In addition, we also discuss about the application of the proposed model in assisting Internet end-to-end control.

To highlight the proposed approach, we underline that: (i) it introduces a new Markov-based model of capture Internet end-to-end loss behaviours; (ii) it provides a connection between the model structure and the network internal states/activities, which cause the losses; this connection is useful for assisting traffic control; (iii) it verifies the model in both simulation and over real heterogeneous networks. The obtained results indicate that the proposed model correctly describes loss behaviours of the Internet wired paths. It also works properly on the wireless paths if the loss rate due to the poor wireless channel quality is relatively low. However, the model fails to reflect the loss behaviours of the wireless paths when the losses resulted from the low channel quality are excessive. The reason is the model structure is not designed to reflect the transmission errors on the wireless channels.

The remaining of the paper is organized as follows. In section II, related work is discussed in relation with the proposed model. Section III introduces the new loss model in details. Performance of the proposed model is verified in section IV. The application of the proposed model in assisting end-to-end traffic control is discussed in section V. Finally, section VI provides concluding remarks.

II. Related Work

Network loss behaviours are characterised by loss rates, loss runlength and good runlength distributions. A loss runlength is the number of packets consecutively lost. In contrast, a good runlength, also known as inter-loss distance, is the number of packets consecutively received. An accurate loss model must be able to correctly reflect the loss behaviours in loss rates, loss runlength and good runlength distributions. In addition, if the model can connect the observed loss behaviours to internal network states/activities, it will be more useful for traffic engineering tasks e.g. network congestion control.

This paper is based on “Modelling Internet End-to-End Loss Behaviours: A New Approach”, by Vinh Bui and Weiping Zhu, which appeared in the Proceeding of the First Asia International Conference on Modelling & Simulation, AMS ’07, Phuket, Thailand, 27-30 March 2007.

Gilbert [5] was the first who used a 1^{st}-order 2-state Markov chain to model loss behaviours of a communication channel. The 2-state structure of the Gilbert model reflects the physically observed blocked and unblocked states of the channel. Although this simple model works relatively well at physical level, it fails to capture inter-loss behaviours at packet level, where the losses can be bursty implying a high-order correlation between them [9]. The failure of the Gilbert model also indicates that at packet level, a communication channel may have more than just blocked and unblocked states.

The work of Gilbert was followed by [21], [16], which tried different approaches to better describe the loss behaviours. Yajnik et al. [21] increased the complexity of the model by extending its order to 2^k, which allows the model to capture the temporal correlation between consecutive losses. However, an increase in the model complexity is not always followed by an increase in the model accuracy. Unlike [21], Sanneck et al. [16] dealt with the problem differently. Instead of using a 2^k-order chain, they introduced a so-called extended Gilbert model, which uses separate states to explicitly represent consecutive losses. As a result, the extended model can describe bursty losses more appropriately. However, this approach is too observation-oriented, which mainly aims at capturing but not explaining the observations. Moreover, since only consecutive losses are explicitly modelled, this model may not be able to correctly capture inter-loss behaviours.

Subsequently, Salamatian and Vaton [15] followed by [8], [13], [14], [17], [20] proposed the use of Hidden Markov Models (HMMs) to capture loss behaviours of an Internet path/channel. Compared to the previous models, the Hidden Markov based models can describe the loss behaviours more appropriately due to their double stochastic nature. Nonetheless, the structures of those HMMs were also chosen only to fit the observations, which could prevent them from explaining and subsequently avoiding the losses. In addition, the parameter estimation process of HMMs is usually computational expensive, which makes them less attractive for many time-sensitive traffic control tasks e.g. routing and scheduling.

A common approach of the previous work when selecting the model structure is to be based on the observations. We, on the other hands, construct the proposed model based on the TCP congestion control mechanisms. From our observations, Internet end-to-end loss behaviours are considerably influenced by the TCP congestion control process. Hence, by constructing the model structure from the states of the TCP congestion control mechanisms, we are able to establish a connection between the model and the network internal states. In addition, since the new model structure is observation independent, the proposed model is potentially more robust.

III. Proposed Loss Model

A. Model Construction

As TCP flows account for more than 75% of Internet IP traffic [18], Internet flow level loss characteristics are significantly influenced by the dynamics of TCP. Therefore, we construct the model with three separate states 0,1,2 respectively corresponding to congestion, slow start, and congestion avoidance stage of the TCP congestion control mechanisms [1]. Transition probabilities from one to another state reflect the dynamics of a TCP flow. The Markovness unveils the temporal correlation between the stages. A generic model structure is presented in Fig. 1. For the reference purpose, the connection between the model states and the stages of the TCP congestion control mechanisms is depicted in Fig. 2.

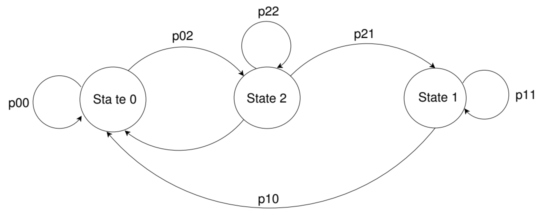

Fig. 1. Generic loss model structure for an end-to-end Internet path.

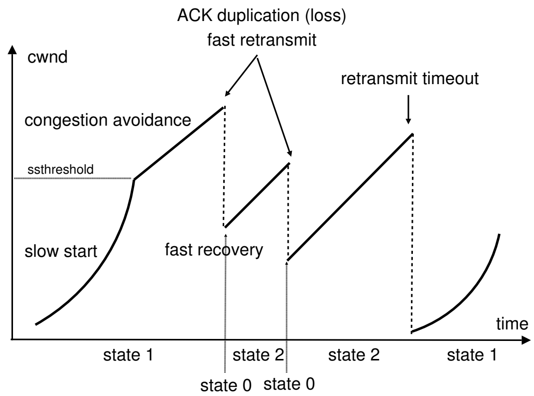

Fig. 2. TCP congestion control in relation with the proposed model.

When a TCP connection is established or restarted, the model is in State 1. No packet loss would occur at this stage since slow start algorithm is applied to exponentially increase the transmission rate to occupy the available bandwidth. After the transmission rate reached the *ssthreshold* [1], congestion avoidance algorithm is used to futher increase the transmission rate. Here, a packet loss could happen with high probability as the transmission rate keeps increasing slowly. In accordance, the model should move from State 1 to State 2. However, the transition from slow start stage to congestion avoidance stage is not directly observable from the view point of end-to-end loss behaviours. Consequently, the model would not make any transition at this moment. Once a packet loss occurs indicating the connection has reached congestion stage, the model correspondingly moves from State 1 to State 0. Due to fast retransmit and recovery mechanisms [1], which quickly recover lost packets detected by a duplication of ACKs, the connection does not make a full back off immediately. In place of a full back-off, it tries comming back to congestion avoidance stage by reducing the transmission rate normally by a half of the current value for every lost segment [1]. As a result, the connection may avoid suffering consecutive losses, which would lead to a full back-off. The transmission rate is then increased again by congestion avoidance mechanism. To reflect these dynamics, we have State 2 in the model structure. Everytime a packet is received, the model moves from State 0 to State 2 and may move back to State 0 if a packet loss occurs. The model keeps bouncing between State 0 and State 2 until a retransmit timeout causes the connection to fully back off. In this case, the transmission rate is quickly decreased towards the initial value and the model moves from State 2 to State 1 accordingly. As the model never moves directly from State 1 to State 2 as well as from State 0 to State 1, those transition probabilities associated with that moves are equal to 0 and not shown in the model structure (see Fig. 1).

In order to reflect the correlation between stages of a TCP connection, we use the model of 2^{nd}-order since the 1^{st}-order model cannot appropriately describe the correlation. In a 1^{st}-order Markov model, the next state of the model depends only on the current state. Meanwhile, the next stage of a TCP connection sometimes depends on both the current and the previous stage. For example, if the current stage of a TCP connection is the congestion avoidance then its next stage also depends on whether the previous stage of the connection was the slow start or the congestion. The model of a higher order is not required since the 2^{nd}-order model can appropriately capture the correlation between the stages.

B. Model Parameters

As the proposed model is 2^{nd}-order Markov-based, it is characterized by the initial state distribution matrix Π and the 2^{nd}-order state transition matrix P.

Π is a [3x1] probability matrix,

$$\Pi = (\pi_0 \ \pi_1 \ \pi_2)$$

in which $\pi_i = Pr(S_0 = i)$ denotes the probability of being in state i at time step 0.

P is a [3x9] probability matrix,

$$P = \begin{array}{c} \\ 00 \\ 01 \\ 02 \\ 10 \\ 11 \\ 12 \\ 20 \\ 21 \\ 22 \end{array} \begin{array}{c} \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{pmatrix} p_{000} & & p_{002} \\ & & \\ p_{020} & & p_{022} \\ p_{100} & & p_{102} \\ p_{110} & p_{111} & \\ & & \\ p_{200} & & p_{202} \\ p_{210} & p_{211} & \\ p_{220} & p_{221} & p_{222} \end{pmatrix} \end{array}$$

in which $p_{xyz} = Pr(S_{t+1} = z | S_t = y, S_{t-1} = x)$ denotes the probability of being in state z at time step $(t+1)$ given the model was in state y at time step t and state x at time step $(t-1)$. Since not all state transitions are possible, those probabilities associated with the invalid moves are equal 0.

To different degree the state transition probabilities reflect the interaction between the TCP flow and background traffic, which results in the observed loss behaviours. Consequently, they could reveal some characteristics of the background traffic such as burstiness and intensity. In order to better understand each transition probability, we will study them with the model.

Initially, the model is in State 1 and will remain in this state with probability p_{111}. In practice, p_{111} is usually close to 1.0 (> 0.98 as observed in our numerical study) indicating a non-congested traffic condition. The model leaves State 1 for State 0 with probability p_{110} corresponding to a packet loss. Once the model is in State 0, it will stay in this state with probability p_{100} for the next step and p_{000} for futher steps. These probabilities in combination with p_{200} indicate the probability of consecutive packet losses, which reveals the intensity and burstiness of background traffic.

If the model leaves State 0 for State 2, it will stay in this state with probability p_{022} for the next step and p_{222} for further steps. A high p_{022} (> 0.7) often indicates a congested traffic condition. From State 2, the model may move to State 1 with probability p_{221} corresponding to a back-off event or return to State 0 with probability p_{220} if a packet loss occurs. The value of p_{221} is protocol specific, which indicates how well the protocol can resist against the losses. If the model moves to State 1 corresponding with TCP slow start, it is likely that the model will stay in this state for a long period. Hence, the probability p_{211} is usually high (> 0.98). Indeed, not a single probability alone is capable of providing a complete picture of the traffic condition unless they are used in combination.

The average loss rate after sending n packets can be calculated from state occupancy statistics [7] of the model. If we define $v_{ij}(n)$ to be a number of times state j is entered through time n, given that the model started in state i at time zero and $x_{ij}(n)$ is the following step function:

$$x_{ij}(n) = \begin{cases} 1: & \text{if the model is in state j after n steps} \\ 0: & \text{otherwise} \end{cases}$$

If the model is in state i at time 0 then:

$$v_{ij}(n) = \sum_{k=0}^{n} x_{ij}(k)$$

Consequently, the expectation of $v_{ij}(n)$ equals:

$$\overline{v}_{ij}(n) = \sum_{k=0}^{n} \overline{x}_{ij}(k) = \sum_{k=0}^{n} \phi_{ij}(k)$$

where:

$\phi_{ij}(k)$ is a k-step transition probability from state i to state j. Given that, the average loss rate of the model after sending n packets is:

$$\overline{r}_{loss}(n) = \frac{1}{3} \sum_{i=0}^{2} \frac{v_{i0}(n)}{n}$$

If n is sufficiently large, $\overline{r}_{loss}$ will converge to π_0, which is the steady state distribution probability of State 0.

Distributions of loss runlengths and good runlengths can also be obtained from the model parameters. Let $P_{loss}(k)$ denotes the probability of observing k consecutive losses. If L represents an event of a packet is lost and S represents an event of a packet is successfully received, we have:

$$P_{loss}(k) = P(S \underbrace{L \ldots LL}_{k} S)$$

If k=1, since both State 1 and State 2 of the model indicate a packet is successfully received, we have:

$$\begin{aligned} P_{loss}^{(k=1)} &= P(101) + P(201) + P(102) + P(202) \\ &= 0 + P(201) + 0 + P(202) \\ &= P(2|01)P(01) + P(2|02)P(02) \\ &= p_{102}\pi_{01} + p_{202}\pi_{02} \end{aligned}$$

If k=2,

$$\begin{aligned} P_{loss}^{(k=2)} &= P(1001) + P(2001) + P(1002) + P(2002) \\ &= 0 + P(2001)) + 0 + P(2002) \\ &= P(2|001)P(001) + P(2|002)P(002) \\ &= P(2|00)P(0|01)P(01) + P(2|00)P(0|02)P(02) \\ &= (p_{100}\pi_{01} + p_{200}\pi_{02})p_{002} \end{aligned}$$

If k=3,

$$\begin{aligned} P_{loss}^{(k=3)} &= P(10001) + P(20001) + P(10002) + P(20002) \\ &= 0 + P(20001) + 0 + P(20002) \\ &= P(2|0001)P(0001) + P(2|0002)P(0002) \\ &= P(2|00)P(0|00)P(001) + P(2|00)P(0|00)P(002) \\ &= (p_{100}\pi_{01} + p_{200}\pi_{02})p_{000}p_{002} \end{aligned}$$

In general, for $k > 3$:

$$P_{loss}^{(k>3)} = (p_{100}\pi_{01} + p_{200}\pi_{02})p_{000}^{k-2}p_{002}$$

π_{ij} can be obtained by solving the equation $\boldsymbol{\pi} = \boldsymbol{\pi}P^*$, where P^* is the double-state transition probability matrix with $p*_{vxyz} \equiv P(vx|yz) = P(v|yz)P(x|yz) = p_{vyz}p_{xyz}$.

Similarly, if we define:

$$P_{good}(k) = P(L\underbrace{S \dots SS}_{k}L)$$

to be the probability of receiving k consecutive packets, so this probability also can be obtained using a similar manner.

C. Model Parameters Estimation

We represent end-to-end loss events using L and S symbols where S denotes a successfully received packet and L denotes otherwise. The model state sequence can be obtained by mapping the loss events to the model states. State 0 corresponding to a lost packet is easy to decide. However, State 1 and State 2 are more tricky as both correspond to successfully received packets. Since transitions from State 1 to State 2 and vice versa are not directly observable from the end-to-end point of view, which success event belongs to which of these states has to be decided empirically. The hint is the holding time of State 1 is usually longer compared to the holding time of State 2. The model state sequence is obtained as follows:

All L symbols are mapped to State 0. All segments of consecutive S symbols lengths of which are equal or less than K and the first K symbols of the remaining are mapped to State 2. All the rest S symbols are mapped to State 1. K is an empirical constant, which indicates the resistance of TCP against the consecutive losses. In practice, K is usually

... SSSS ... SLSSLSLLSLLLLLSSSSSS ... SS ...

...1111 ... 1022020020000022222 1 ... 11 ...

Fig. 3. Example of conversion from loss series state sequence ($K = 5$)

less than or equal 5 for most TCP implementations. The state mapping with $k = 5$ is illustrated in Fig. 3.

Having the state sequence, the initial state distribution matrix Π can be subsequently estimated as follows:

$$\pi_i = \frac{F_i}{\sum_{i=0}^{2} F_i}$$

where:
F_i is a number of times the model is in state i ($0 \le i \le 2$).

Similarly, the state transition matrix P also can be obtained.

$$p_{xyz} = \frac{T_{xyz}}{\sum_{z=0}^{2} T_{xyz}}$$

where:
T_{xyz} is a number of transitions from state x to state y and from state x to state z ($0 \le x, y, z \le 2$).

We notice that, p_{x01} and p_{x12}, ($0 \le x \le 2$), all equal 0 as the model never moves directly from state 0 to state 1, and from state 1 directly to state 2.

IV. Model Verification

A. Simulation

We created a simulation network using NS-2 [10] as presented in Fig. 4. Background traffic was generated using FTP, CBR and exponentially ON/OFF cross traffic sources. The number of cross traffic sources, their intensity, activation time and connection pair are random parameters chosen from a specific range to guarantee the dynamics of the network (see TABLE I). Target flows are TCP (Reno) with 1500 bytes packets traversing from node $n0$ to node $n2$ and from node $n1$ to node $n3$ alongside with other flows. Link $n5$ to $n6$ implements RED queue. All other links use DropTail queue.

TABLE I
Parameters of background traffic sources.

	FTP	CBR	ON/OFF
Num of Source	1000	250	250
Activation	Rand[0-1000]	Rand[0-1000]	Rand[0-1000]
Packet size	1024 bytes	512 bytes	128 bytes
Rate (Mbps)	N/A	Rand[0.1-1]	Rand[0.1-1]
Send size (KB)	Rand[500-10000]	N/A	N/A
Burst time	N/A	N/A	500ms
Idle time	N/A	N/A	500ms

Next, we varied the parameter range to create different loss scenarios. For each scenario, the network was simulated for 500 seconds i.e. approximately 10000 packets per target flow. The loss trace obtained from the target flows was used to train the model. Afterwards, the network was simulated for 1000 seconds to verify the model. The following are results of 3 simulation scenarios presented in the form of average loss rate, initial state distribution matrix, state transition probability matrix, loss runlength and good runlength distributions.

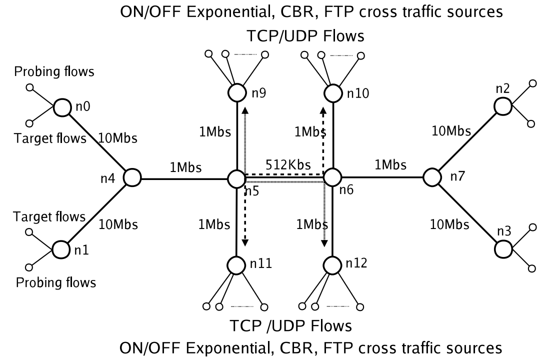

Fig. 4. Simulation network.

1) Scenario 1: We simulated the network in a low intensity cross traffic condition with the average loss rate of 0.45%. In this condition, the target flow is relatively stationary. The average loss rate obtained from the model is 0.42%, which is accurate. The distributions of good runlengths and loss runlengths of the model in comparision with those obtained from the simulation is also match as presented in Fig. 5.

$$\Pi = \begin{pmatrix} 0.004289 & 0.985573 & 0.010138 \end{pmatrix}$$

$$P = \begin{array}{cc} & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{array}{c} 00 \\ 01 \\ 02 \\ 10 \\ 11 \\ 12 \\ 20 \\ 21 \\ 22 \end{array} & \begin{pmatrix} 0.375000 & & 0.625000 \\ & & \\ 0.531915 & & 0.468085 \\ 0.138889 & & 0.861111 \\ 0.001426 & 0.998574 & \\ & & \\ 0.086207 & & 0.913793 \\ 0 & 1 & \\ 0.048485 & 0.224242 & 0.727273 \end{pmatrix} \end{array}$$

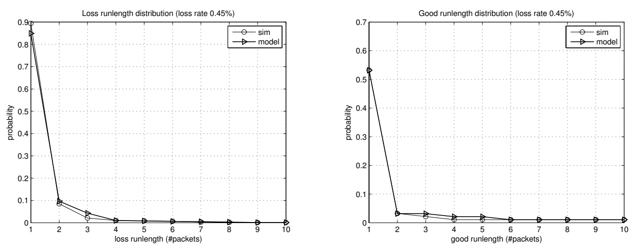

Fig. 5. Loss runlength and good runlength distributions.

2) Scenario 2: We increased the background traffic intensity to medium level, which resulted in the average loss rate of 2.40%. Consequently, we observed some burstiness in the background traffic, which indicates the traffic is non-stationary. The average loss rate calculated from the model is 2.38%. The comparison in terms of good runlengths and loss runlengths between the model and the simulation is illustrated in Fig. 6.

As illustrated, the model performs very well in this scenario as the average loss rate, good runlength distribution and loss runlength distribution obtained with the model are all match with those obtained from the simulation.

$$\Pi = \begin{pmatrix} 0.023808 & 0.932443 & 0.043749 \end{pmatrix}$$

$$P = \begin{array}{cc} & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{array}{c} 00 \\ 01 \\ 02 \\ 10 \\ 11 \\ 12 \\ 20 \\ 21 \\ 22 \end{array} & \begin{pmatrix} 0.421569 & & 0.578431 \\ & & \\ 0.353846 & & 0.646154 \\ 0.379452 & & 0.620548 \\ 0.006915 & 0.993085 & \\ & & \\ 0.360870 & & 0.63913 \\ 0.047945 & 0.952055 & \\ 0.071118 & 0.211901 & 0.716981 \end{pmatrix} \end{array}$$

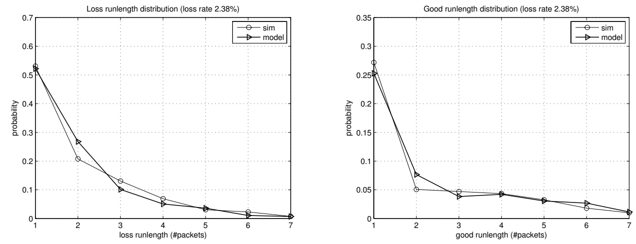

Fig. 6. Loss runlength and good runlength distributions.

3) Scenario 3: We simulated the network in a high intensity cross traffic condition with the average loss rate of 3.93%. We observed a high level of burstiness in the background traffic. The model average loss rate (4.06%), and the comparison in terms of good runlength and loss runlength distributions between the model and the simulation (Fig. 7) confirm a good performance of the proposed model in the simulated scenarios.

$$\Pi = \begin{pmatrix} 0.040632 & 0.883056 & 0.076312 \end{pmatrix}$$

$$P = \begin{array}{cc} & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{array}{c} 00 \\ 01 \\ 02 \\ 10 \\ 11 \\ 12 \\ 20 \\ 21 \\ 22 \end{array} & \begin{pmatrix} 0.456140 & & 0.543860 \\ & & \\ 0.266129 & & 0.733871 \\ 0.455224 & & 0.544776 \\ 0.013028 & 0.986972 & \\ & & \\ 0.552632 & & 0.447368 \\ 0.007408 & 0.992592 & \\ 0.074419 & 0.209302 & 0.716279 \end{pmatrix} \end{array}$$

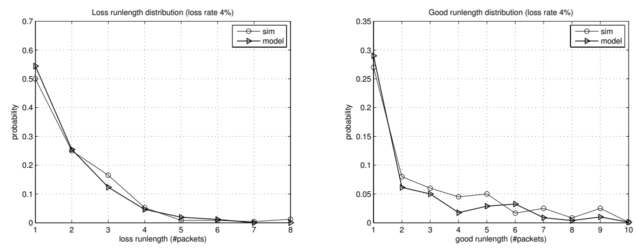

Fig. 7. Loss runlength and good runlength distributions.

B. Trace-based Analysis

We verified the model using Internet traces extracted from the Gigabit Ethernet connection entering UMASS [19]. We compared the model in terms of good runlength and loss runlength distributions against two well-known loss models, which are Gilbert model and 3-state HMM. The result presented in Fig. 8 shows that the model outperforms both Gilbert model and HMM especially in terms of good runlength distribution. As flows in the trace are TCP where the losses are strongly correlated, the Gilbert model shows the poorest performance in capturing good runlength behaviours compared to HMM and the proposed model.

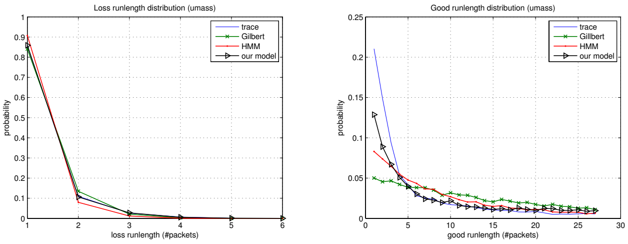

Fig. 8. Loss runlength and good runlength distributions.

We also verified the proposed model using real Internet traces obtained by means of an Active Measurement approach with D-ITG (*Distributed Internet Traffic Generator*) [11] in various heterogeneous network paths. In particular, we studied the model performance in real Internet environment with different network access technologies (Ethernet, IEEE 802.11, ADSL), operating systems (Linux and Windows), end-users devices (Workstation, Laptop, Palmtop), and packet sizes (64, 256, 512, 1024). Therefore, it allows us to assess the impact of several network parameters on the model performance.

Due to space limitations we can not present here all the results we obtained. Therefore, we present those related to 6 representative paths, data sets characteristics of which are summarized in TABLE II. These data sets have been collected using TCP, Linux operating system, and Laptop computers.

TABLE II

CHARACTERISTICS OF THE DATA SETS.

Name	Sender/Rcv.	Pkt size	Loss rate
Eth2Adsl64	Ether/ADLS	64	4.67%
Eth2Adsl256	Ether/ADLS	256	3.13%
Eth2Adsl1024	Ether/ADLS	1024	39.3%
Eth2Wifi64	Ether/802.11b	64	0.41%
Eth2Wifi512	Ether/802.11b	512	0.72%
Eth2Wifi1024	Ether/802.11b	1024	16.9%

In order to evaluate the robustness of the model, the data sets were chosen to exhibit significantly different loss behaviours i.e. different loss rates, loss runlength and good runlength distributions. Each data set was divided into 2 equal parts. The first half was used to trained the model and the second half is used to evaluate the model performance. For each scenario, the evaluation results are presented in forms of average loss rate, loss runlength and good runlength distribution. TABLE III summarizes the result in terms of loss rates.

As presented, the loss rates predicted by the model are reasonably accurate for Eth2Adsl traces. Since the losses on wired paths e.g. Eth2Adsl are mainly due to network congestions, the model can appropriately describe them thanks to its structure, which is constructed from TCP congestion control mechanism. However, the model did not predict accurately the loss rates for 2 (*Eth2Wifi512 and Eth2Wifi1024*) of the 3 wireless paths. The reason is that the losses on those wireless paths are due to both network congestions and data transmission errors. When the transmission errors rate is high, the model is no longer capable of describing the loss behaviours since it is not designed to model the combined losses. We can see that the loss rates predicted by the model are lower than the actual loss rates.

TABLE III

PREDICTED LOSS RATE.

Name	Trace loss rate	Predicted loss rate
Eth2Adsl64	4.67%	4.12%
Eth2Adsl256	3.13%	2.93%
Eth2Adsl1024	39.3%	39.7%
Eth2Wifi64	0.41%	0.40%
Eth2Wifi512	0.72%	0.55%
Eth2Wifi1024	16.9%	12.1%

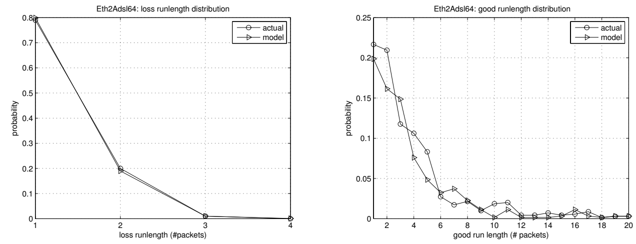

Fig. 9. Eth2Adsl64: Loss runlength and good runlength distributions.

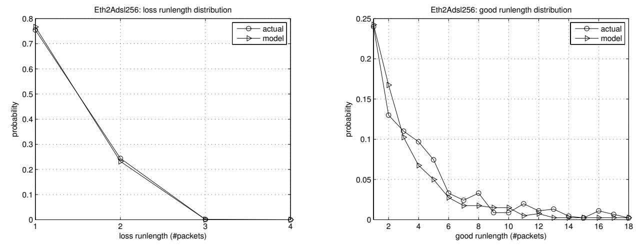

Fig. 10. Eth2Adsl256: Loss runlength and good runlength distributions.

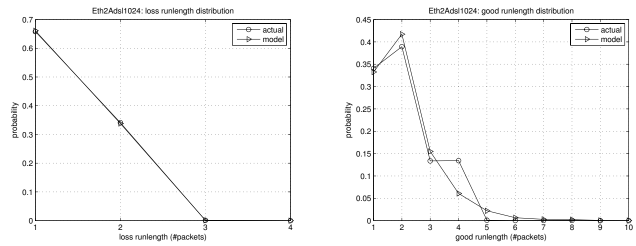

Fig. 11. Eth2Adsl1024: Loss runlength and good runlength distributions.

Apart from loss rates, the model performance is assessed based on ability to capture the loss patterns i.e. loss runlength and good runlength distributions. In wired environments e.g.

Ethernet and ADSL, the model performs well under different loss scenarios. The loss runlength and good runlength distributions of Eth2Adsl data sets in comparion with those produced by the model are depicted in Figs.(9, 10, and 11). As illustrated, the model appropriately captures the loss patterns of those data sets. However, as mentioned above, the model did not work properly for some wireless paths such as *Eth2Wifi512* and *Eth2Wifi1024*. Figs.(13,14) subsequently depict the loss runlength and good runlength distributions of *Eth2Wifi512* and *Eth2Wifi1024* set in comparison with those of the model. For *Eth2Wifi64* set, the model works relatively well (see Fig. 12) since the transmission losses are not significant.

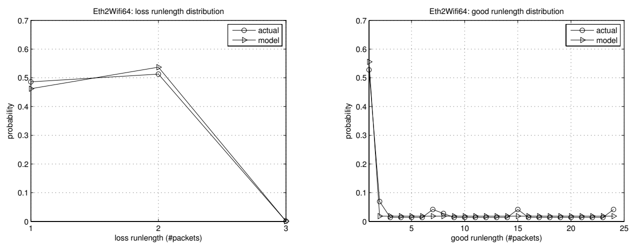

Fig. 12. Eth2Wifi64: Loss runlength and good runlength distributions.

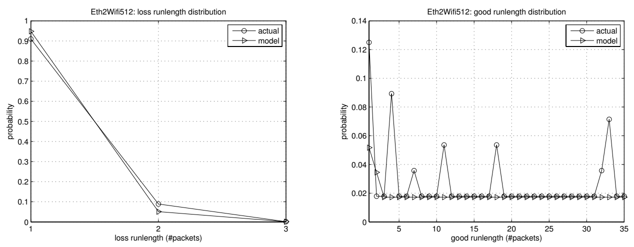

Fig. 13. Eth2Wifi512: Loss runlength and good runlength distributions.

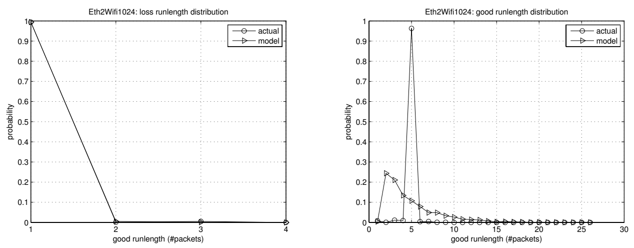

Fig. 14. Eth2Wifi1024: Loss runlength and good runlength distributions.

V. Model-based Traffic Control

Apart from the accuracy and robustness, an important advantage of the proposed loss model is the ability to link the loss observations to the path congestion states, which cause the losses. Thus, by observing end-to-end loss behaviours, the congestion states of the path can be inferred using the model. Once the congestion states are obtained, actions can be taken to improve the path performance and reliability. To be more specific, since each congestion state is associated with a certain packet loss probability, by applying a data transmission strategy with an adaptive transmission rate, one can obtain an optimal rate/loss ratio. Furthermore, if multiple disjoint paths are available, traffic can also be distributed among the paths to achieve high transmission rates and low loss rates at the same time. In this section, we show how the rate adaptive problem with a single path and the traffic distribution problem with multiple paths can be formulated using the proposed loss model under the framework of Markov Decision Processes.

A Markov Decision Process (MDP) [12] is a controlled Markov chain defined by a four-component tuple $\{S, A, T, R\}$ where $S = \{s\}$ is the state space of the chain; $A = \{a\}$ is a set of actions, which can be taken by the decision maker in each state to control the dynamics of the chain; $T = T(s, a, s')$ is the state transition probability function, which yields a probability the chain currently in state s will move to state s' in the next step as a result action a; $R = R(s, a, s')$ is the reward function, which specifies the value given to the decision maker if the chain moved from state s to state s' by taking action a. Solving a control problem under the framework of Markov Decision Processes means to find a policy, which dictates the action to take in each state to maximize the total (average) rewards accumulated over a decision horizon.

In contexts of the rate adaptive problem, the decision maker adjusts data transmission rates by deciding at decision epochs whether packets must be transmitted or delayed depending on the path congestion states. Subsequently, the four-component tuple $\{S, A, P, R\}$ of the problem is defined as follows:

- $S = \{0, 1, 2\}$ is the state space of the model, which reflects the path congestion states.
- $A = \{0, 1\}$ is a set of 2 actions where 1 means to transmit packets, and 0 means not to transmit.
- $T = T(s, a, s'); a \in A; s, s' \in S$ is the state transition probability function, which yields the value of:

$$T(s, a, s') = \begin{cases} 1.0 : & \text{if } a = 0, s' = s \\ 0.0 : & \text{if } a = 0, s' \neq s \\ p_{ss'} : & \text{if } a = 1 \end{cases}$$

  where $p_{ss'}$ is the probability the model currently in state s will move to state s' in the next step.
- $R = R(s, a, s')$ is the reward function, which should be able to reflect the optimal rate/loss ratio objective. We define $R(s, a, s')$ as follows:

$$R(s, a, s') = \begin{cases} +1.0 : & \text{if } a = 1, s' = 1, 2, \forall s \in S \\ -3.0 : & \text{if } a = 1, s' = 0, \forall s \in S \\ -1.0 : & \text{if } a = 0, \forall s, s' \in S \end{cases}$$

  In brief, a reward of $+1.0$ is given to the decision maker if a packet is successfully transmitted ($s' = 1, 2$). In contrast, if the packet is lost ($s' = 0$), a penalty of -3.0 is given. The action of delaying a packet costs the decision maker a value of -1.0. The reward and penalty values can be adjusted depending on rate/loss requirements.

In a similar manner, the traffic distribution problem is formulated as a MDP. The major difference between the two formulations is the state and action space. In the traffic distribution problem, several paths are available for data transmission. Hence, the state space is the combinations of all path states i.e. a combined state is the vector of all path states. At decision epochs, the decision maker selects a path to transmit data among the available paths depending on the congestion state of each path. Subsequently, the four-component tuple $\{S, A, P, R\}$ of the problem is defined as follows:

- $S = \{(s_1, s_2 \ldots, s_K), s_i = \{0, 1, 2\}\}$ is the combined state space, which has 3^K states. K is the number of transmission paths.
- $A = \{0, 1 \ldots K\}$ is a set of $K + 1$ actions where $1 \ldots K$ means to transmit packets on the $1^{st} \ldots K^{th}$ path, and 0 means to delay packets transmission.
- $T = T(s, a, s'); a \in A; s, s' \in S$ is the state transition probability function, which yields the value of:

$$T(s, a, s') = \begin{cases} 1.0: & \text{if } a = 0, s' = s \\ 0.0: & \text{if } a = 0, s' \neq s \\ p_{ss'}: & \text{if } a = 1 \end{cases}$$

  where $p_{ss'}$ is the probability the combined chain currently in state s will move to state s' in the next step.
- $R = R(s, a, s')$ is the reward function, which should be able to reflect the optimal rate/loss ratio objective. In specific, a reward of $+1.0$ is given to the decision maker if a packet is successfully transmitted. In contrast, if the packet is lost, a penalty of -3.0 is given. The action of delaying a packet costs the decision maker a penalty value of -1.0. The reward and penalty values can be adjusted depending on rate/loss requirements.

Having defined the four-component tuple, an optimal transmission policy can be found using the standard MDP solving tools such as dynamic programming and linear programming. Due to space constrains, the implementation details, results and discussions will be presented in another paper.

VI. Conclusion and Future Work

In this paper, we have proposed a new Markov-based model to capture Internet end-to-end loss behaviours. Unlike the previous loss models, the proposed model is constructed based on the TCP congestion control mechanisms. As a result, the model can appropriately describe flow level loss behaviours and at the same time connect them to network flow level activities. Simulation and Internet trace-based analysis have shown that the proposed model can correctly capture important loss behaviours e.g. average loss rates, loss runlength and good runlength distributions under heterogeneous network conditions without explicitly modelling them. This fact indicates that the model structure is accurate and it is appropriately connected to flow level network activities. Since the model is 2^{nd}-order, it partially explains why TCP traffic exhibit self-similar properties, which are governed by long-tailed distributions. In future work, we will investigate the implementation of the model in assisting application-level traffic control.

References

[1] M. Allman, V. Paxson, and W. Stevens. (1999, Apr.) Tcp congestion control. rfc2581.txt. [Online]. Available: http://www.ietf.org/rfc/

[2] V. Q. Bui and W. Zhu, "Modelling internet end-to-end loss behaviours: A new approach," in *Proc. of Asia Modelling Symposium 2007*, Phuket, Thailand, Mar. 2007, pp. 94–99.

[3] E. O. Elliott, "Estimates of error-rate for codes on burts-noise channels," *Bell System Technical Journal*, vol. 42, pp. 1977–1997, Sept. 1963.

[4] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the internet," *IEEE/ACM Trans. Networking*, vol. 7, no. 4, pp. 458–472, Aug. 1999.

[5] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell System Technical Journal*, vol. 39, pp. 1253–1265, Sept. 1960.

[6] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, "A clean slate 4d approach to network control and management," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 5, pp. 41–54, Dec. 2005.

[7] R. A. Howard, *Dynamic Probabilistic Systems Volume 1: Markov Models*. New York, USA: Wiley, 1971.

[8] G. Iannello and al, "End-to-end packet-channel bayesian model applied to heterogeneous wireless networks," in *Proc. GLOBECOM*, USA, 2005.

[9] W. Jiang and H. Schulzrinne, "Modeling of packet loss and delay and their effect on real-time multimedia service quality," presented at the Proc. the 10th International Workshop on Network and Operating System Support for Digital Audio and Video, USA, June 2000.

[10] NS-2. (2006, July) Network simulatror 2. [Online]. Available: http://nsnam.isi.edu/nsnam/index.php

[11] A. Pescapé and al. (2006, Nov.) Traffic project. [Online]. Available: http://www.grid.unina.it/Traffic/

[12] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. USA: John Wiley and Sons, 1994.

[13] P. S. Rossi, F. Palmieri, and G. Iannello, "Hmm-based monitoring of packet channels," in *Proc. of the 7th IEEE International Conference on High Speed Networks and Multimedia Communications (HSNMC)*, Toulouse, France, June 2004, pp. 144–154.

[14] P. S. Rossi, G. Romano, F. Palmieri, and G. Iannello, "A hidden markov model for internet channels," in *Proc. of the 3rd IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Darmstadt, Germany, Dec. 2003, pp. 50–53.

[15] K. Salamatian and S. Vaton, "Hidden markov modeling for network communication channels," in *Proc. ACM SIGMETRICS 2001*, USA, June 2001, pp. 92–101.

[16] H. Sanneck and G. Carle, "A framework model for packet loss metrics based on loss runlengths," in *Proc. of the SPIE/ACM SIGMM Multimedia Computing and Networking Conference*, USA, Jan. 2000, pp. 177–187.

[17] S. Tao and R. Guerin, "On-line estimation of internet path performance: An application perspective," in *Proc. IEEE INFOCOM*, vol. 3, HK, Mar. 2004, pp. 1774–1785.

[18] K. Thompson and et al., "Wide-area internet traffic patterns and characteristics," *IEEE Network*, vol. 11, no. 6, pp. 10–23, Nov. 1997.

[19] UMASS. (2005) Network trace gateway-2. [Online]. Available: http://prisms.cs.umass.edu/repository/data/ds/gwlink/1/trace.tgz

[20] W. Wei, B. Wang, and D. Towsley, "Continuous-time hidden markov models for network performance evaluation," *Elsevier Performance Evaluation*, vol. 49, no. 1-4, pp. 129–146, Sept. 2002.

[21] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modelling of the temporal dependence in packet loss," in *Proc. IEEE INFOCOM*, vol. 1, USA, Mar. 1999, pp. 345–352.

Vinh Bui is currently a Ph.D candidate at the School of Information Technology and Electrical Engineering, Univeristy of New South Wales, Australia. He received his BE and MS degrees in Electrical Engineering and Information Technology from Hanoi University of Technology and from Univeristy of New South Wales respectively in 1995 and 2001. His research interests include distributed systems, computer and communication networks and embedded systems.

Weiping Zhu received B.Eng and M.Eng from Hunan University and China University of Mining and Technology in 1982 and 1984, respectively. He received Ph.D from University of New South Wales, Australia in 1992. He is currently a senior lecturer at the School of Information Technology and Electrical Engineering, University of New South Wales, Australia. His research interests covers distributed systems, computer networks and network tomography.

Ruhul Sarker obtained his Ph.D in Operations Research (1991) from DalTech (former TUNS), Dalhousie University, Halifax, Canada. He is currently a senior lecturer in Operations Research at the School of Information Technology and Electrical Engineering, University of New South Wales (UNSW), ADFA Campus, Canberra, Australia. His research interests include applied mathematical modelling, optimization and evolutionary computation.