University of Glasgow

Modelling Packet Loss in RTP-Based Streaming Video for Residential Users

Martin Ellis[1], Dimitrios Pezaros[1], Theodore Kypraios[2], Colin Perkins[1]

[1] School of Computing Science, University of Glasgow

[2] School of Mathematical Sciences, University of Nottingham

http://martin-ellis.net

Cisco, San Jose: 2012/06/15

Motivation

- Models for packet loss are useful for simulation studies.
- Markov-chain models widely used in simulation for video techniques (e.g., FEC).
- However, the accuracy of these models hasn't been studied for streaming to residential Internet users (i.e., using DSL/Cable).

This talk:

1. Presents an evaluation study of the accuracy of Markov models for simulating packet loss in residential streaming;
2. Introduces a more accurate two-level model, to address the shortcomings of existing models.

Outline

Outline

Why Model?

Packet loss models allow simulation of loss patterns:

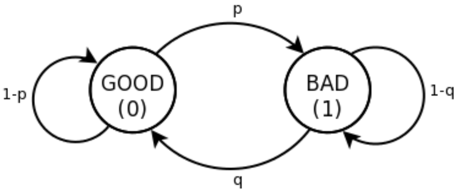
- evaluation of error recovery mechanisms
- determining impact of loss on coding schemes

A common way to do this is using Markov chain models:

- model parameters determine the probability of packet loss
- advantages: don't need large amounts of real trace data for simulation

With simulation, we can evaluate application performance before deployment, *provided that the simulation models are accurate*.

(Simple) Gilbert Model



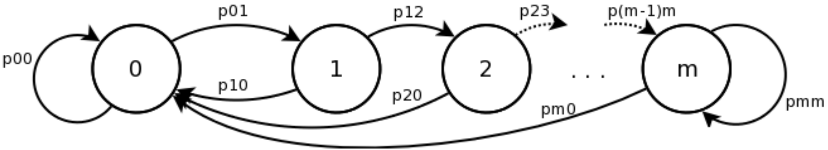
SGM has been widely used in multimedia performance evaluation (e.g., [1][2]).

It is has states directly representing received (0) and lost (1) packets. To estimate transition probabilities, we just count the number of transitions between states (i.e., recv/recv, recv/loss, loss/recv, loss/loss).

[1] Tao *et al.* Real-Time Monitoring of Video Quality in IP Networks. *IEEE/ACM Trans. Netw.*, 2008.

[2] Tournoux *et al.* On-the-Fly Erasure Coding for Real-Time Video Applications. *IEEE Trans. Multimedia*, 2011.

Extended Gilbert Model



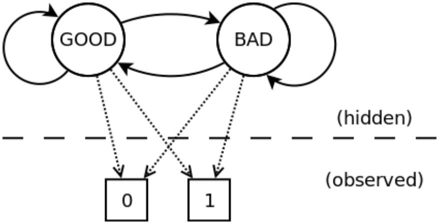
EGM aims to capture burstiness in packet loss, by increasing the number of states for packet loss to m [3].

Parameters estimated similarly to the SGM, by counting transitions.

[3] Jiang & Schulzrinne. Modeling of Packet Loss and Delay and Their Effect on Real-Time Multimedia Service Quality. *Proc. NOSSDAV*, 2000.

Hidden Markov Model

HMMs aim to capture transitions between "hidden states" [4] (i.e., bursty/non-bursty packet loss).



We look at the loss sequence, deriving with maximum likelihood:

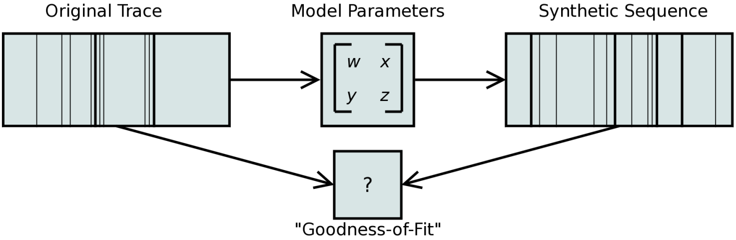
- transition probabilities between the unobserved "hidden" states
- the probability of packet loss within each of these states

Can increase the number of states, but this increases estimation time. We focus on 2- and 3-state HMMs.

[4] Silveira & de Souza e Silva. Modeling the short-term dynamics of packet losses. *Performance Evaluation Review*, 2006.

Outline

Evaluation procedure



We estimate model parameters from real packet loss traces, then simulate synthetic sequences and compare.

Packet loss data

RTP streaming of IPTV-like traffic from a well-connected server to residential Internet users:

- have ∼3800 traces (1–10 mins) from 14 links in the UK and Finland
- loss and delay observations for ∼230 million packets

- many have little or no loss (modelled easily by SGM, EGM, HMMs), but others show bursty, correlated loss
- here, we focus on the traces showing bursty loss (∼430 traces)

The dataset is described in [5], and is available for download at `http://martin-ellis.net/research/datasets`.

[5] Ellis *et al.* *End-to-End and Network-Internal Measurements of Real-Time Traffic to Residential Users*. In *Proc. ACM MMSys*, 2011.

Model evaluation : testing "goodness-of-fit"

We use two approaches for comparing sequences:

- (subjective) visual comparison of original/synthetic at individual trace level
- comparison of statistics from original trace vs. those from 1000 synthetic sequences (test whether original trace likely to have come from the model)

Model evaluation : parametric bootstrap

We use a variation on bootstrapping, a widely-used statistical technique [6][7].

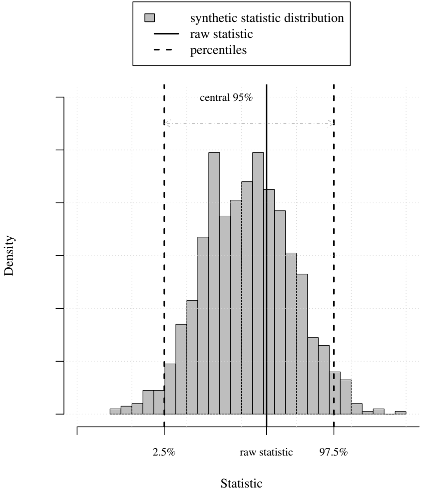
For each loss trace, we generate 1000 synthetic sequences using the model parameters, and calculate a set of statistics from each:

- mean loss rate
- percentiles of receive run-length distribution
- mean/median/max loss run-length

[6] Downey. Lognormal and Pareto distributions in the Internet. *Computer Communications*, 2005.

[7] Tariq *et al.* Poisson versus periodic path probing (or, does PASTA matter?). *Proc. ACM IMC*, 2005.

Goodness-of-Fit testing

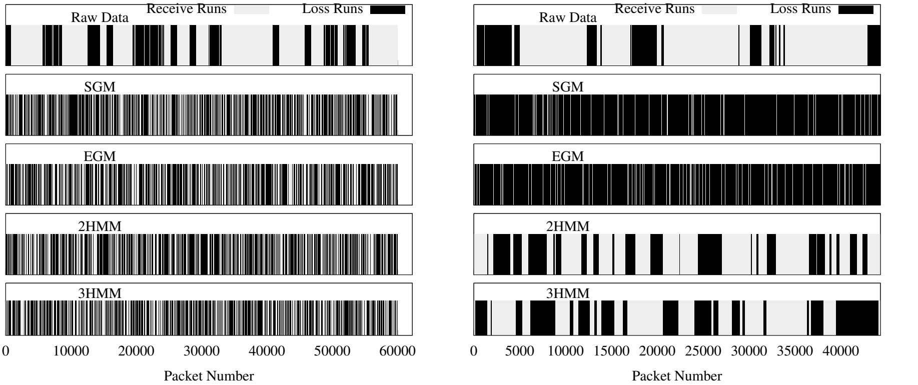


For each statistic (e.g., mean loss rate):

- calculate distribution of statistic from 1000 synthetic traces, and examine where raw statistic falls
- if raw stat falls within central 95%, no evidence of poor fit
- if raw stat falls outside central 95%, it suggests this value is unlikely to have been produced by the model → poor fit

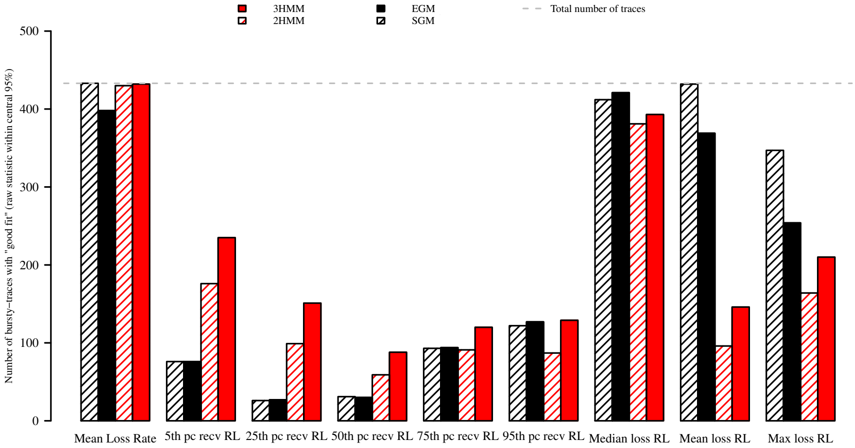
Outline

Example traces



For some traces, all models perform poorly; for others, the HMMs perform better.

Parametric bootstrap



Higher bars represent more traces with good “fit” (better model accuracy).

Summary : existing models not sufficient . . .

SGM and EGM models fail to capture the gaps between receive runs:

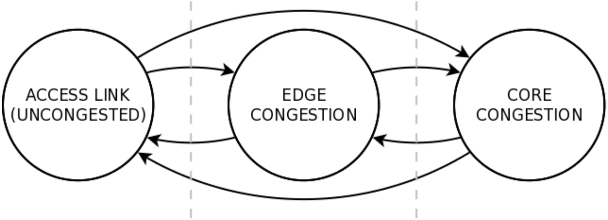
- instead, they just aim on getting the average loss rate or loss burst length

HMMs appear to be a little better, but can still be inaccurate:

- "hidden" states estimated by HMMs don't correspond to the underlying network states

Outline

Motivation



Traces show that there were clear "state changes" in loss patterns.

Idea: improve model by explicitly identifying states (with loss/delay data), and modelling each state separately.

- *uncongested*: low loss, low delay
- *edge congestion*: higher loss, high/variable delay (e.g., spikes)
- *core congestion*: higher loss, without the clear delay signals

Identifying state transitions

We use 2 simple classification algorithms, splitting the traces into 1-second windows and considering loss and delay per-window.

Packet loss:

$>$ 2 loss events or $>$ 2 loss bursts per-window $\implies$ "high loss"

Delay (loss/delay threshold *ld* classifier):

median delay $>$ 5ms $\implies$ "high delay"

Delay (loss/delay before loss *ldbl* classifier):

median delay before loss $>$ 2x median delay $\implies$ "high delay"

Identifying state transitions

We use 2 simple classification algorithms, splitting the traces into 1-second windows and considering loss and delay per-window.

Packet loss:

$>$ 2 loss events or $>$ 2 loss bursts per-window $\implies$ "high loss"

Delay (loss/delay threshold *ld* classifier):

median delay $>$ 5ms $\implies$ "high delay"

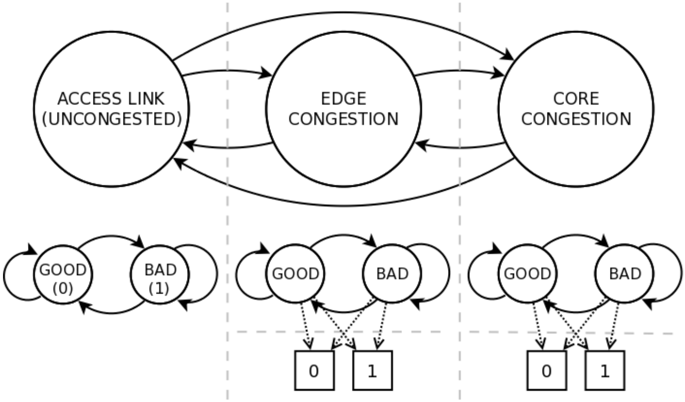
Delay (loss/delay before loss *ldbl* classifier):

median delay before loss $>$ 2x median delay $\implies$ "high delay"

State transitions:

- uncongested $\rightarrow$ edge congestion : "high loss" and "high delay"
- uncongested $\rightarrow$ core congestion : "high loss" and "low delay"
- edge/core congestion $\rightarrow$ uncongested : "low loss" and "low delay"

Combining classifiers with loss models

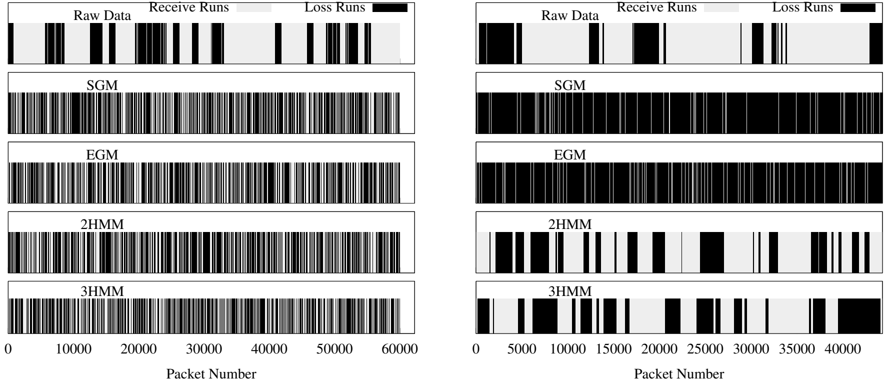


Different parameters for each "outer" state

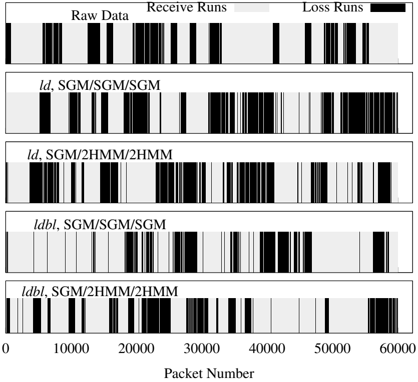
- can use different models too – HMMs better for congestive loss?

Outline

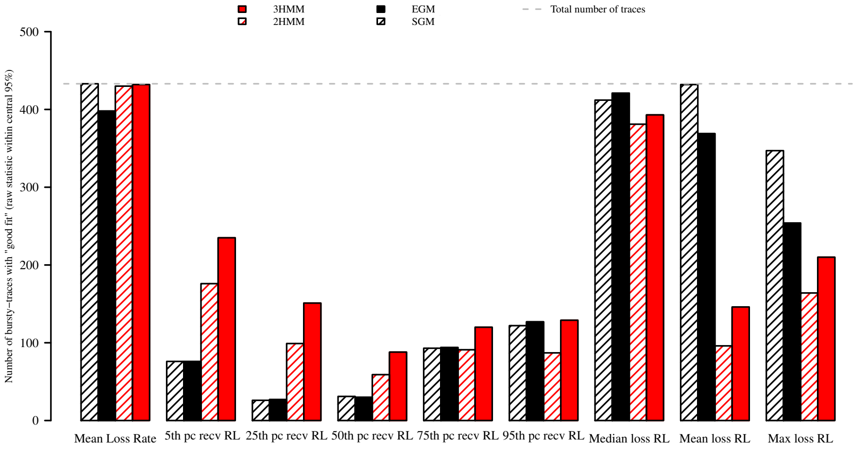
Example traces (recap of SGM, EGM, HMMs)



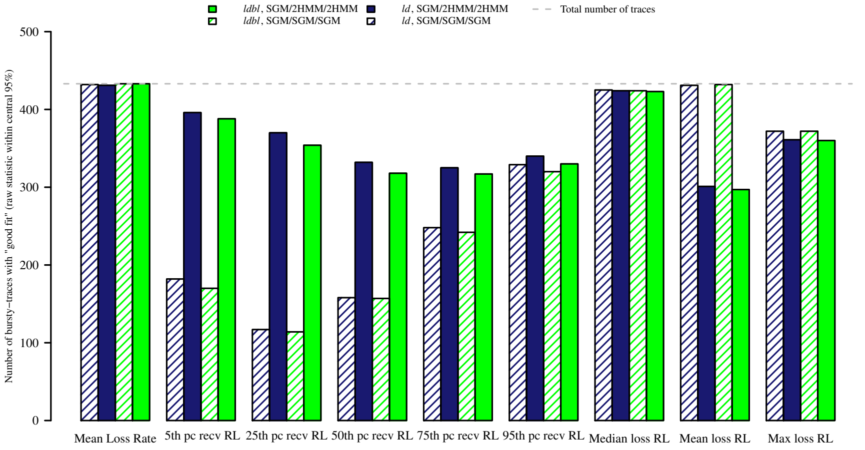
Example traces (two-level model)



Parametric bootstrap (recap of SGM, EGM, HMMs)



Parametric bootstrap (two-level model)



Summary : new models give more accurate performance

More "well-modelled" traces (in terms of all the statistics) than before:

- new models are more accurate
- using HMMs within congested states gives best performance

Outline

Future Directions

Applying models in evaluation studies:

- evaluation of FEC schemes, etc.
- network simulation

Applying models in real-time:

- use classification to do adaptation?
- applications in quality monitoring / anomaly detection?

Summary

In this work, we have:

- Shown that existing packet loss models (SGM, EGM, HMMs) perform poorly in bursty packet loss conditions seen on residential links.
- Designed a new two-level model to express changes in network state (using loss/delay data to classify performance).
- Demonstrated improved accuracy over the existing models.

ld classifier

```
if (state = "uncongested") then
    if (N > 2) or (M > 2) then            # "high loss"
        if (DQ~ > 5ms) then               # "elevated DQ"
            state ← "edge congestion"
        else
            state ← "core congestion"
        end if
    end if
else
    if (N ≤ 2) and (M ≤ 2) and (DQ~ ≤ 5ms) then
        state ← "uncongested"
    end if
end if
```

N	number of losses per window
M	number of loss bursts per window
$\widetilde{DQ}$	median queueing delay per window

ldbl classifier

if (state = "uncongested") **then**
  if $(N > 2)$ or $(M > 2)$ **then**   # "high loss"
    if $(\widetilde{DQ}_{BL} > 2\widetilde{DQ})$ **then**   # "elevated *DQ*"
      state ← "edge congestion"
    else
      state ← "core congestion"
    end if
    $\widetilde{DQ}_{UC} \leftarrow \widetilde{DQ}$
  end if
else if (state = "edge congestion") **then**
  if $(N \leq 2)$ and $(M \leq 2)$ and $(\widetilde{DQ} \leq k\widetilde{DQ}_{UC})$ **then**
    state ← "uncongested"
  end if
else if (state = "core congestion") **then**
  if $(N \leq 2)$ and $(M \leq 2)$ **then**
    state ← "uncongested"
  end if
end if

N	number of losses per window
M	number of loss bursts per window
$\widetilde{DQ}$	median queueing delay per window
$\widetilde{DQ}_{BL}$	median queueing delay (before loss) per window
$\widetilde{DQ}_{UC}$	median queueing delay in last uncongested window
k	threshold for delay "close to previous" $(k = 1.1)$