A Mathematical Model of the Skype VoIP Congestion Control Algorithm

Luca De Cicco, S. Mascolo, V. Palmisano

Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari

47th IEEE Conference on Decision and Control
Cancun, Mexico

December 9th 2008

Outline

Introduction

- The Internet was designed for bulk data transfer, not for delivery of multimedia contents that are packet loss tolerant, but delay-sensitive
- Even though several multimedia CC transport protocols have been proposed (TFRC, DCCP, ARC, RAP), commercial applications use proprietary congestion control algorithms (if any) over UDP
- Skype is the leading Voice over IP application, generating an enormous amount of UDP flows transported in the Internet
- It is important to address the issues and the properties of the traffic generated by Skype VoIP application
- Mathematical models play a major role in understanding the fundamental properties, such as the stability of large scale and complex communication systems

State of the art

- VoIP traffic is generally considered as CBR, which is not the case of Skype
- Skype Audio/Video flows employ some sort of congestion control algorithm [WWIC07, NOSSDAV08]
- The Skype VoIP CC exhibits remarkable unfairness *wrt* TCP concurrent flows and a very slow reaction to sudden bandwidth variations (transient lasts $\sim 40\,\mathrm{s}$)
- Aim of the work: proposing and verifying a mathematical model that is able to explain the observed behaviour of Skype VoIP generated sending rate
- Modeling is complicated by the following factors:
    - Skype is a closed source application (source code is not available for inspection)
    - Skype uses AES encryption algorithm to crypt packets
    - It is reasonable to assume that the controller implements switching dynamics due to if-then-else statements

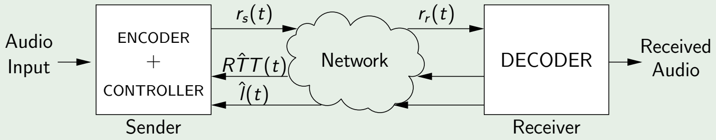
Schematic of the considered system

End-to-end congestion control algorithms typically react to congestion when:

- packet losses occur (loss based congestion control)
- the latency of the connection grows (delay based congestion control)

Schematic of the system

Therefore we consider *loss ratio* $\hat{l}(t)$ and *round trip time* $\hat{RTT}(t)$, as candidate inputs of the controller



Experiments to Investigate the Skype Congestion Control

The *Skype Measurement Lab*

Testbed

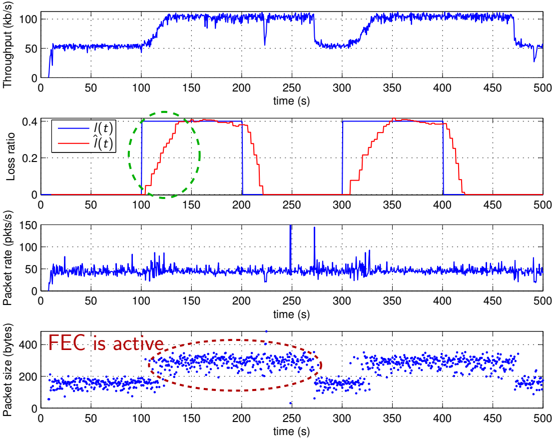
- We employ a network emulator to vary:
  - link *packet loss ratio* (PLR) $l(t)$
  - link *capacity* $c(t)$
  - link *Round Trip Time* $RTT(t)$
- The emulator allows to log *throughput*, *loss rate,* and *packet size* produced by each flow
- We are also able to log information (RTT, PLR, jitter) provided by Skype

Experiments to investigate Skype VoIP cc

- Skype over a variable loss ratio link
- Skype over a variable RTT link
- Skype over a variable available bandwidth link

Skype Audio over a variable loss rate link

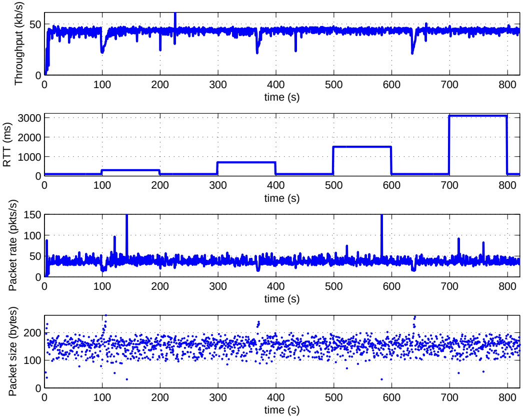
Loss ratio: square wave, period 200 s, max value 0.4, min value 0



1. When the imposed $l(t)$ increases, the sending rate increases, indicating that *the congestion control algorithm is not loss based*
2. The loss ratio $\hat{l}(t)$ as measured by Skype (in red) is a *low pass filtered* version of the signal $l(t)$ (blue)
3. The packet sizes evolution increases when $l(t)$ increases, indicating that Skype employs a Forward Error Correction (FEC) technique to counteract persistent packet losses

Skype Audio over a variable RTT link

Variable RTT: variations happen each 100 s in the range $[200, 3000]ms$



- The generated sending rate does not vary significantly when RTT varies, so that we can conclude that the *CC algorithm is not delay based*
- The packet size evolution indicates the FEC action is not activated

The Skype VoIP Congestion Control Model

Proposed Model

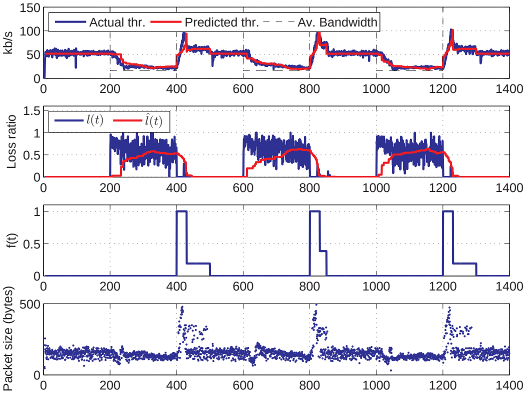
- We make the hypothesis that the encoder employed by Skype is multi-rate so that it can choose among N levels of encoding $L = \{L_1, L_2, \ldots, L_N\}$

Proposed model

$$r_s(t) = (1 - \hat{l}(t)) \cdot (1 + f(t)) L_{i(t)} \qquad (1)$$

- $\hat{l}(t)$ is the loss ratio as measured by Skype
- $f(t)$ FEC action, $f(t) \in [0, 1]$ ($f(t) = 0$ when FEC is off, $f(t) = 1$ when the FEC action is at maximum)
- $L_{i(t)}$ is the encoding level selected at time t

Validation of the proposed model



- The sending rate predicted by eq. (1) follows the rate produced by Skype
- FEC is not active during congestion (after a bandwidth drop occurs)
- FEC is activated after the available bandwidth increases to its maximum value (packet size doubles) to counteract packet losses in the "probing phase"

The long transient time exhibited by the CC algorithm is due to the filtering of the loss ratio

A Skype flow accessing a bottleneck

Setting: available bandwidth $b(t)$, drop-tail queue length q_M, fw path delay T_1, bw path delay T_2

Queue length model $q(t)$

$$\dot{q} = \begin{cases} 0 & q = 0, r \leq b \text{ or } q = q_M, r \geq b \\ r - b & \text{otherwise} \end{cases} \tag{2}$$

where r is the queue input rate. The overflow rate is given by:

$$o = \begin{cases} r - b & q = q_M, r > b \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Measured loss ratio $\hat{l}(t)$ model

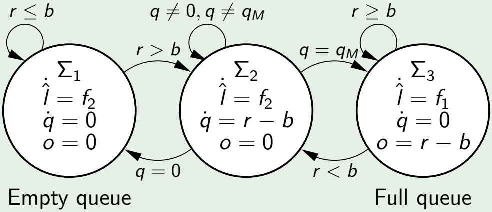
Recalling that $l = o/r$ and employing a first order filter with time constant τ ($\tau \cong 11\,\text{s}$) we have $\dot{\hat{l}} = -\frac{1}{\tau}\hat{l} + \frac{1}{\tau}\frac{o_{T_2}}{r_{T_2}}$. Considering (1) and (3) we obtain:

$$\dot{\hat{l}} = \begin{cases} f_1 = \frac{1}{\tau} - \frac{\hat{l}}{\tau} - \frac{b_{T_2}}{\tau(1-\hat{l}_{T_2})(1+f_{T_2})L_{T_2}} & q = q_M,\ r > b \\ f_2 = -\frac{1}{\tau}\hat{l} & \text{otherwise} \end{cases} \tag{4}$$

Skype accessing a bottleneck: hybrid model

It is simple to show that the following hybrid automaton models the considered system (equations (2) and (4)):

Hybrid Automaton $\mathcal{H}$



Characterizing the equilibria of $\mathcal{H}$

- Characterization of the equilibrium states of $\mathcal{H}$
- Steady state characterization of a Skype Audio flow accessing a bottleneck with capacity b
- The time delay is neglected since $\tau \gg RTT$

Characterization of the steady state

Proposition 1: *Let us consider the equilibrium inputs b^*, L^* and f^* the automaton $\mathcal{H}$ is characterized by the following equilibria that are asymptotically stable:*

$$\hat{l}^* = 1 - \sqrt{\frac{b^*}{L^*(1+f^*)}}\,;\; q^* = q_M \quad \text{iff} \quad b^* < (1+f^*)L^* \tag{5}$$

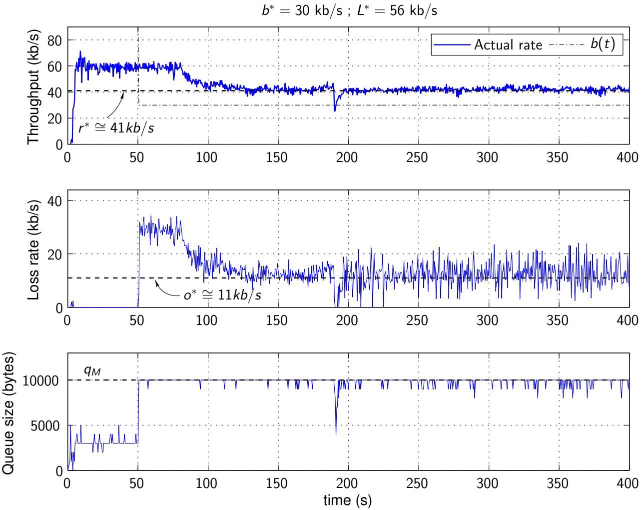
$$\hat{l}^* = 0\,;\; q^* = 0 \quad \text{iff} \quad b^* \geq (1+f^*)L^* \tag{6}$$

Proposition 2: *The controller employed by Skype is not able to avoid congestion unless $b^* > L^*(1+f^*)$.*

Dim. We are under the hypothesis of Prop 1 $\Rightarrow$ (5) is asymptotically stable. At steady state it results $o^* = (1-\hat{l}^*)L^*(1+f^*) - b^*$, thus substituting (5) in the equation just written we have:

$$o^* = \sqrt{b^* L^*(1+f^*)} - b^* > 0$$

Verification of the proposition



- Available bandwidth drop at $t = 50$, $b^* = 30\,\text{kb/s}$
- The experiment shows that the steady state values for $r(t)$ and $o(t)$ recover the ones predicted by the model

Conclusions

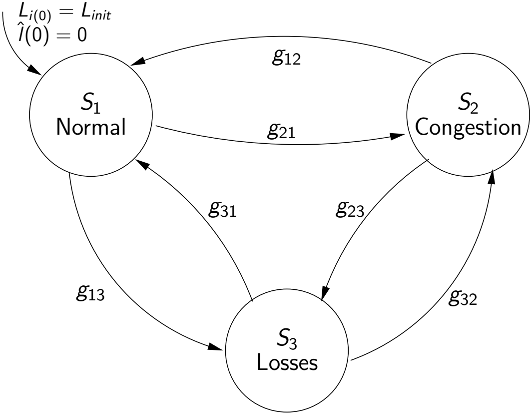
- We have proposed a mathematical model of the Skype closed loop congestion control
- Main contributions are:
  - Characterization of the equilibrium states of the system
  - Skype is not able to cope with congestion efficiently (a finite error is present and the queue is full)
- The model has been validated using both network experiments and NS-2 simulations

Thank you for the attention

Sunset in Cancun, Dec 2008

Backup slides

The Skype Hybrid Automaton



- Three different dynamics depending on the state. The guard conditions g_{ij} are still not identified
- S_1- Normal: no congestion occurs, no persistent loss due to link are present, rate is constant
- S_2 - Congestion: the link is congested, $r_s(t)$ varies according eq (1)
- S_3 - Losses: no congestion occurs, losses due to the link are present (FEC action is activated in this state)

Equilibrium of $\mathcal{H}$

Lemma 1: The system Σ_3 (full queue) has a unique asymptotically stable equilibrium:

$$\hat{l}^* = 1 - \sqrt{\frac{b^*}{L^*(1+f^*)}}\,;\ q^* = q_M \tag{7}$$

when $b^* < (1+f^*)L^*$.

Lemma 2: The system Σ_1 (empty queue) has a unique asymptotically stable equilibrium :

$$\hat{l}^* = 0\,;\ q^* = 0 \tag{8}$$

when $b^* > (1+f^*)L^*$

Lemma 3: The hybrid automaton $\mathcal{H}$ has a sink state Σ_3 if $b^* < (1+f^*)L^*$ and a sink state Σ_1 if $b^* > (1+f^*)L^*$