

Evaluating a Congestion Management Architecture for SMS Gateways

A. Gonzalez Prieto, R. Stadler
KTH, Royal Institute of Technology, Stockholm, Sweden
{gonzalez, stadler}@imit.kth.se

Abstract

In the paper, we evaluate our congestion management architecture for SMS systems. In our architecture an SMS operator can dynamically define the maximum acceptable loss of messages of a non-guaranteed SMS service class, thereby controlling the trade-off between minimal message loss and maximum throughput in a SMS system. We study our proposal through simulations in different scenarios.

1. Introduction

The Short Message Service (SMS) permits users to send and receive text messages to/from their mobile phones. SMS was introduced in 1992 and, since then, has experienced a remarkable success: 45 billion messages are sent per month.

We consider two classes of SMS service. The first is the guaranteed service, which guarantees delivery, offering zero losses. The second, currently considered by service providers, is the non-guaranteed service. The rationale for service differentiation is the existence of different uses of SMS: person-to-person, bulk messaging, etc.

The SMS gateway (SMSG) is the functional block in the SMS architecture that interconnects the wireless network to others, like other mobile operator's network or TCP/IP networks. When receiving a message, the SMSG routes it to the appropriate outgoing port. The ports in a SMSG are software ports. Each of them has an associated queue. This permits the SMSG to cope with brief periods of congestion. However, long periods require control mechanisms.

In [1], we proposed a policy-based architecture for congestion management for SMS gateways. In this paper, we evaluate our architecture through simulations.

2. System Design

We use two mechanisms to address congestion in the SMSG. The first one is reducing the **acceptance rate** at the incoming port. The acceptance rate controls the admission of messages from an incoming port into the SMSG. We set the acceptance rate of incoming ports individually. The drawback of this mechanism is that the reduction of the acceptance rate affects the traffic to all outgoing ports, not only the congested ones. Therefore, it compromises the overall throughput of the gateway.

The second mechanism is **dropping** some of the non-guaranteed messages that are routed to a congested port. This mechanism permits having a higher overall throughput at the cost of introducing losses.

These mechanisms present a trade-off: throughput vs. losses.

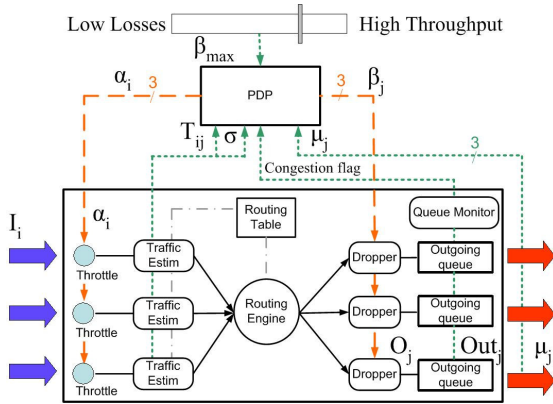


Figure 1: Functional Architecture for SMS Gateway Congestion Management

under congestion. β_{\max} takes values from 0% (no losses) to 100% (high throughput). The maximum allowable dropping policy provides the maximum throughput under the constraint that, at most, β_{\max} % of non-guaranteed messages is dropped.

Figure 1 presents the functional architecture of our proposal[1]. The **throttles** are responsible for enforcing the acceptance rates (α_i) at the incoming ports. The **traffic estimators** estimate the future (i) traffic matrix T_{ij} and (ii) percentage of guaranteed messages σ . The prediction is based on the average value in a moving window. The **droppers** enforce the dropping percentage of non-guaranteed messages (β_j). The **PDP** (policy decision point) is the main block of the architecture: it is responsible for adapting dynamically to congestion. It calculates the values for α_i and β_j that maximize the overall throughput for the steady state, keeping the drops below β_{\max} . To do so, it solves periodically an optimization problem whose objective function is: $G(x) = \sum_j \sum_i \alpha_i T_{ij} (1 - (1 - \sigma) \beta_j)$ subject to: (i) $\sum_i \alpha_i T_{ij} (1 - (1 - \sigma) \beta_j) \leq \mu_j \quad \forall j$, (ii) $1 \geq \beta_{\max} \geq \beta_j \geq 0 \quad \forall j$ and (iii) $\alpha_{\max} \geq \alpha_i \geq 0 \quad \forall i$. μ_j stands for the service rate of port j .

3. Evaluation through Simulation

We have evaluated our proposal through simulation. For this purpose, we have developed a SMS gateway simulator, including our congestion management proposal.

We have analyzed the system in three scenarios: service rate decrease in one port, service rate increase in one port, and traffic matrix change. The three scenarios share the following characteristics: (i) the port configuration consists of three incoming and three outgoing ports; (ii) Poisson sources inject traffic at 50m/s to each incoming port; (iii) the nominal service rate of each outgoing port is 50 m/s; the nominal acceptance rate of each incoming port is 50 m/s; (iv) 10% of the messages use the guaranteed service; (v) congestion is declared when the average queue size reaches 100 messages (Q); (vi) the capacity of the outgoing queues is 500 messages; (vii) during congestion, the PDP re-calculates the optimal configuration every second. In the figures below, throughputs are 1-second averages; losses are 10-seconds averages; queue sizes are instant values. We only present losses for the ports where they are not zero.

The SMSG operator has to specify the policy for the SMSG during congestion: give priority to (i) having low losses, or (ii) having high throughput. She does so using a sliding bar (figure 1). The performance metric to favor is indicated by the position of the slider. This qualitative priority is mapped to a quantitative parameter: the **maximum allowable dropping percentage** (β_{\max}). It indicates the maximum percentage of non-guaranteed messages that can be dropped

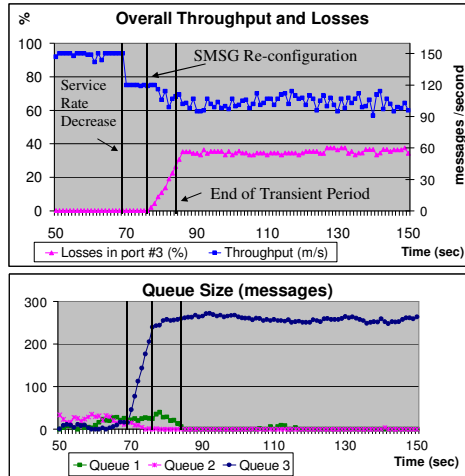


Figure 2: Service Rate Decrease (port 3)

the gateway, there is a short **transient period**. In this case, it is 9 seconds long.

3.2. Service Rate Increase in Port 3

In this scenario, we analyze the behavior of our system when the service rate of one outgoing port increases. The experiment (figure 3) starts with outgoing port 3 serving at 20 m/s, causing congestion. During this period, the PDP evaluates the policy and determines the optimal configuration. β_{\max} is 20%, and the traffic matrix is balanced. At time=70 s, outgoing port 3 recovers its nominal service rate, and starts serving at 50 m/s. The recovery follows a step function.

The system adapts to the new situation within 15 seconds. This depends on the time the system requires to correctly estimate the new service rate. In our experiments, it is estimated by averaging it in the last ten seconds. Longer averaging windows lead to longer transient periods, while shorter windows react more abruptly to brief service rate variations.

3.3. Traffic Matrix Change

The experiment (figure 4) starts with an unbalanced traffic matrix that does not cause congestion in any outgoing port (all serving at 50 m/s). β_{\max} is 20%. At time=70 s, the traffic matrix changes, following a step function. For each incoming port, 50% of the messages go to outgoing port 1, 40% to port 2, and 10% to port 3. This causes congestion in outgoing ports 1 and 2.

The system detects congestion in 9 seconds. Then, the transient period lasts for 39 seconds, which is longer than in the other experiments. However, this only has a marginal effect in the throughput. It falls to 110 m/s two seconds after the matrix change. From then on, it keeps stable with minor oscillations. The reason is that the traffic sent to port 2 (O_2) obtained with the new configuration is very close to μ_2 .

3.1. Service Rate Decrease in Port 3

In this scenario, we analyze the behavior of our system when the service rate of outgoing port 3 slows down. The experiment (figure 2) starts with all outgoing ports serving at 50 m/s. β_{\max} is 40%, and traffic matrix is unbalanced –uneven distribution of traffic to the outgoing ports. At time=70 s, outgoing port 3 slows down to 20 m/s following a step function, causing congestion.

It takes the system 6 seconds to detect congestion. This time depends on Q: higher values cause slower reaction, but they avoid reacting against short periods of congestion. After the system detects congestion and re-configures

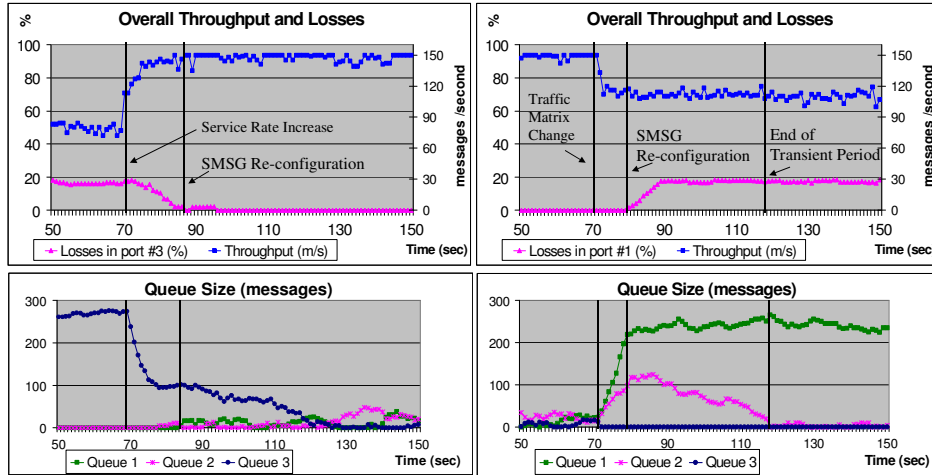


Figure 3: Service Rate Increase (port3)

Figure 4: Traffic Matrix Change

4. Discussion

We have evaluated our proposal through simulation in different scenarios. In all the experiments, the system **controls congestion** after the operator's policy: the outgoing queues never get full while keeping the constraint that, at most, $\beta_{\max}\%$ of non-guaranteed messages is dropped. The experiments we presented show that the system reacts fast to service rate, and traffic matrix variations.

The end of the **transient phase** is determined by all the outgoing queues getting empty. The duration of the transient phase depends on the size of the queues when congestion is detected. It also depends on β_{\max} , and the traffic matrix. It is longer for higher β_{\max} and unbalanced matrices. The reason is that, in both cases, we obtain a higher throughput. Therefore, it takes longer to empty the queues.

Additional experiments have confirmed that the overall throughput in steady state depends on β_{\max} . It is higher for higher β_{\max} . This **dependency is not linear**. The derivative of this function is higher for higher β_{\max} .

In a forthcoming paper, we will present the evaluation of a prototype of our architecture. It is implemented on a commercial SMSG: the EMG [1].

References

- [1] A. Gonzalez Prieto, R. Cosenza, and R. Stadler, "Policy-based Congestion Management for an SMS Gateway", IEEE 5th International Workshop on Policies for Distributed Systems and Networks (POLICY 2004), Yorktown Heights, New York, June 7-9, 2004
- [2] Nordic Messaging, www.nordicmessaging.se