SIP Bottleneck Analysis in IMS

Dmitriy Gorbunov
State University of Telecommunications
St.Petersburg, Russia
kmplx@inbox.ru

Abstract

Performance analysis is a key task for intellectual networks design. This paper shows a theoretical approach to analyze IMS-based UMTS network performance characteristics, such as bottleneck.

INDEX TERMS: IMS, SIP, PERFORMANCE ANALYSIS, BOTTLENECK.

I. INTRODUCTION

Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. These sessions can contain any combination of media (voice, data, video, audio files, anything), and can be modified at any time to add new parties or to change the nature of the session.

The main purpose of the research has been to analyze the key perfomance characteristics of the SIP operation in IP Multimedia Subsystem.

The performance analysis was based on the deep investigation of the call setup and termination procedures. The signaling flow, passing along routing chain, hits each functional entity (one of CSCFs or HSS) several times. On hit, the functional entity provides some service for the signalling flow and then forwards the traffic on. We counted the hit times of each entity and detected the bottle-necks on different scenarios. We concluded that which one node would be the bottleneck depends on how the traffic is distributed in the networks.

II. MAIN PART

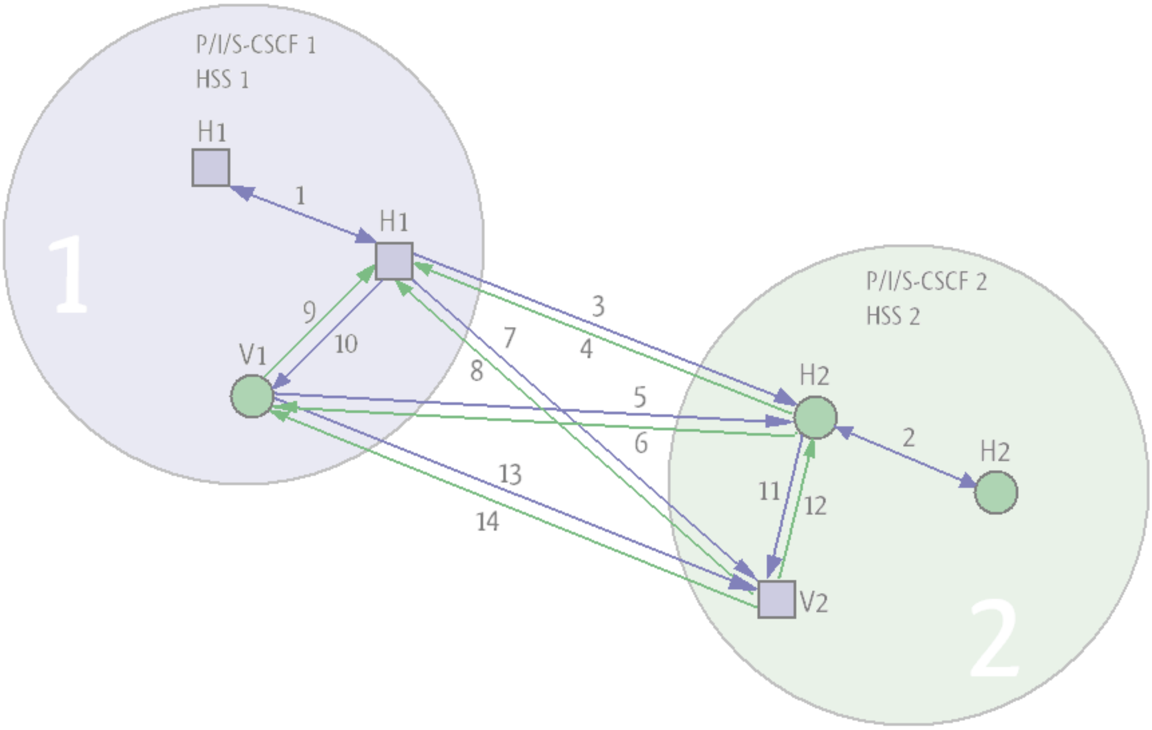
We know that for each call to be setup/released, the signalling traffic goes through each involved functional entities several times. In another word, each functional entity on the routing chain is hit by the signalling traffic flow several times. Once hit, the CSCF (or HSS) provides some service and then forwards the traffic on. If we consider the nodes involved as servers, then every server has a queue formed by different traffic flows waiting to get through. Which server will be the bottleneck is a critical problem that we are interested in. The analysis of the bottleneck will help us increase the capacity of the network, and thus increase the revenue.

If we assume no media update occurs, we will get that in a call set up/release procedure, P-CSCF is hit 11 times, S-CSCF is hit 11 times, I-CSCF is hit 12 times, and HSS is hit one time.

Assume we have two networks of two different operators. We assume there is one P-CSCF, S-CSCF and I-CSCF in each network, and they provide the service in a «first-come-first-served» (FCFS) order. Assume the mean service time of P-CSCF is $\overline{X}p$, mean service time of S-CSCF is $\overline{X}s$, mean service time of I-CSCF is $\overline{X}i$, and mean service time of HSS is $\overline{X}h$.

Then we will do the bottleneck analysis based on the assumptions; we will get all of the possible scenarios within the two networks as follows:

H1: non-roaming Network1 subscriber
H2: non-roaming Network2 subscriber.
V1: Network2 subscriber roaming at Network1
V2: Network1 subscriber roaming at Network2.



We know that for one call setup/release procedure, each P-CSCF and S-CSCF involved is hit 11 times, each I-CSCF involved is hit 12 times; Assume the total traffic load is λ (calls/time unit), traffic load for each scenario is λ_i, and $\lambda=\Sigma\lambda_i$

Then we will have the total hits of each node:

P-CSCF1:
$\Lambda_1 = (\lambda_3+\lambda_4+\lambda_5+\lambda_6+\lambda_7+\lambda_8+\lambda_{13}+\lambda_{14}) \times 11 + (\lambda_1+\lambda_9+\lambda_{10}) \times 22;$

P-CSCF2:
$\Lambda_2 = (\lambda_3+\lambda_4+\lambda_5+\lambda_6+\lambda_7+\lambda_8+\lambda_{13}+\lambda_{14}) \times 11 + (\lambda_2+\lambda_{11}+\lambda_{12}) \times 22;$

S-CSCF1:
$\Lambda_3 = (\lambda_3+\lambda_4+\lambda_9+\lambda_{10}+\lambda_{11}+\lambda_{12}+\lambda_{13}+\lambda_{14}) \times 11 + (\lambda_1+\lambda_7+\lambda_8) \times 22;$

S-CSCF2:
$\Lambda_4 = (\lambda_3+\lambda_4+\lambda_9+\lambda_{10}+\lambda_{11}+\lambda_{12}+\lambda_{13}+\lambda_{14}) \times 11 + (\lambda_2+\lambda_5+\lambda_6) \times 22;$

I-CSCF1:
$\Lambda_5 = (\lambda_1+\lambda_4+\lambda_7+\lambda_8+\lambda_9+\lambda_{11}+\lambda_{13}) \times 12;$

TABLE 1.
DESCRIPTION OF CALLS AND ROUTING CHAINS OF SIGNALLING TRAFFIC FLOW.

	Scenario	Description	Routing Chain
1	H1↔H1	The non-roaming Network1 subscribers call each other.	P-CSCF1↔S-CSCF1↔I-CSCF1↔HSS1↔S-CSCF1↔P-CSCF1
2	H2↔H2	The non-roaming Network2 subscribers call each other.	P-CSCF2↔S-CSCF2↔I-CSCF2↔HSS2↔S-CSCF2↔P-CSCF2
3	H1→H2	A non-roaming Network1 subscriber calls a non-roaming Network2 subscriber.	P-CSCF1↔S-CSCF1↔I-CSCF2↔HSS2↔S-CSCF2↔P-CSCF2
4	H2→H1	A non-roaming Network2 subscriber calls a non-roaming Network1 subscriber.	P-CSCF2↔S-CSCF2↔I-CSCF1↔HSS1↔S-CSCF1↔P-CSCF1
5	V1→H2	A Network2 subscriber roaming at Network1 calls a non-roaming Network2 subscriber	P-CSCF1↔S-CSCF2↔I-CSCF2↔HSS2↔S-CSCF2↔P-CSCF2
6	H2→V1	A non-roaming Network2 subscriber calls a Network2 subscriber roaming at Network1.	P-CSCF2↔S-CSCF2↔I-CSCF2↔HSS2↔S-CSCF2↔P-CSCF1
7	H1→V2	A non-roaming Network1 subscriber calls a Network1 subscriber roaming at Network2.	P-CSCF1↔S-CSCF1↔I-CSCF1↔HSS1↔S-CSCF1↔P-CSCF2
8	V2→H1	A Network1 subscriber roaming at Network2 calls a non-roaming Network1 subscriber.	P-CSCF2↔S-CSCF1↔I-CSCF1↔HSS1↔S-CSCF1↔P-CSCF1
9	V1→H1	A Network2 subscriber roaming at Network1 calls a non-roaming Network1 subscriber.	P-CSCF1↔S-CSCF2↔I-CSCF1↔HSS1↔S-CSCF1↔P-CSCF1
10	H1→V1	A non-roaming Network1 subscriber calls a Network2 subscriber roaming at Network1.	P-CSCF1↔S-CSCF1↔I-CSCF2↔HSS2↔S-CSCF2↔P-CSCF1
11	H2→V2	A non-roaming Network2 subscriber calls a Network1 subscriber roaming at Network2.	P-CSCF2↔S-CSCF2↔I-CSCF1↔HSS1↔S-CSCF1↔P-CSCF2
12	V2→H2	A Network1 subscriber roaming at Network2 calls a non-roaming Network2 subscriber.	P-CSCF2↔S-CSCF1↔I-CSCF2↔HSS2↔S-CSCF2↔P-CSCF2
13	V1→V2	A Network2 subscriber roaming at Network1 calls a Network1 subscriber roaming at Network2	P-CSCF1↔S-CSCF2↔I-CSCF1↔HSS1↔S-CSCF1↔P-CSCF2
14	V2→V1	A Network1 subscriber roaming at Network2 calls a Network2 subscriber roaming at Network1.	P-CSCF2↔S-CSCF1↔I-CSCF2↔HSS2↔S-CSCF2↔P-CSCF1

I-CSCF2:

$\Lambda_6 = (\lambda_2+\lambda_3+\lambda_5+\lambda_6+\lambda_7+\lambda_{10}+\lambda_{12}+\lambda_{14}) \times 12;$

HSS1:

$\Lambda_7 = (\lambda_1+\lambda_4+\lambda_7+\lambda_8+\lambda_9+\lambda_{11}+\lambda_{13});$

HSS2:

$\Lambda_8 == (\lambda_2+\lambda_3+\lambda_5+\lambda_6+\lambda_{10}+\lambda_{12}+\lambda_{14});$

And we will have the utilization of each entity:

P-CSCF1: $= \Lambda_1 \times \overline{X}p$,
P-CSCF2: $= \Lambda_2 \times \overline{X}p$,
S-CSCF1: $= \Lambda_3 \times \overline{X}s$,
S-CSCF1: $= \Lambda_4 \times \overline{X}s$,
I-CSCF1: $= \Lambda_5 \times \overline{X}i$,
I-CSCF2: $= \Lambda_6 \times \overline{X}i$,
HSS1: $= \Lambda_7 \times \overline{X}h$,
HSS2: $= \Lambda_8 \times \overline{X}h$,

If we assume $\overline{X}p = \overline{X}s = \overline{X}i = \overline{X}i$, we will get the bottleneck if we know the traffic load of each scenario. We get different node to be the bottleneck in different situations, and here are some examples:

	1	2	3	4	5	6
λ_1/λ	0.2	0.1	0.05	0.05	0.1	0.1
λ_2/λ	0.2	0.1	0.05	0.05	0.1	0.1
λ_3/λ	0.05	0.05	0.05	0.05	0.05	0.05
λ_4/λ	0.05	0.05	0.05	0.05	0.05	0.05
λ_5/λ	0.05	0.05	0.15	0.05	0.15	0.05
λ_6/λ	0.05	0.05	0.15	0.05	0.15	0.05
λ_7/λ	0.05	0.05	0.05	0.15	0.05	0.05
λ_8/λ	0.05	0.05	0.05	0.15	0.05	0.05
λ_9/λ	0.03	0.03	0.05	0.05	0.05	0.15
λ_{10}/λ	0.03	0.03	0.05	0.05	0.05	0.15
λ_{11}/λ	0.02	0.12	0.05	0.05	0.05	0.05
λ_{12}/λ	0.02	0.12	0.05	0.05	0.05	0.05
λ_{13}/λ	0.1	0.1	0.1	0.1	0.05	0.05
λ_{14}/λ	0.1	0.1	0.1	0.1	0.05	0.05
Bottleneck	P-CSCF1	P-CSCF2	S-CSCF1	S-CSCF2	S-CSCF1	P-CSCF1

III. CONCLUSION

We find that which one node will be the bottleneck depends on how the traffic is distributed in the networks. For example, when the traffic between non-roaming network1 subscribers and network1 visitors becomes heavier, P-CSCF1 is the bottleneck (situation 6), however if the traffic between non-roaming network2 subscribers and network2 visitors becomes heavier, P-CSCF2 is the bottleneck (situation2). If we increase the traffic load between non-roaming network1 subscribers and roaming network1 subscribers, S-CSCF1 is the bottleneck (situation5), etc.

And since there can be infinite possibilities of traffic distributions, theoretically each node may be the bottleneck in certain situations.