

Insights on the Configuration and Performances of SOME/IP Service Discovery

Jan Seyler

Daimler AG - Mercedes-Benz Cars

Nicolas Navet

University of Luxembourg

Loïc Fejoz

RealTime-At-Work (RTaW)

ABSTRACT

Scalable Service-Oriented Middleware on IP (SOME/IP) is a proposal aimed at providing service-oriented communication in vehicles. SOME/IP nodes are able to dynamically discover and subscribe to available services through the SOME/IP Service Discovery protocol (SOME/IP SD). In this context, a key performance criterion to achieve the required responsiveness is the subscription latency that is the time it takes for a client to subscribe to a service. In this paper we provide a recap of SOME/SD and list a number of assumptions based on what we can foresee about the use of SOME/IP in the automotive domain. Then, we identify the factors having an effect on the subscription latency, and, by sensitivity analysis, quantify their importance regarding the worst-case service subscription latency. The analysis and experiments in this study provide practical insights into how to best configure SOME/IP SD protocol.

CITATION: Seyler, J., Navet, N., and Fejoz, L., "Insights on the Configuration and Performances of SOME/IP Service Discovery," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 8(1):2015, doi:10.4271/2015-01-0197.

INTRODUCTION

Context of the Work

Scalable Service-Oriented Middleware on IP (SOME/IP) is a protocol on top of IP that provides service-oriented communication in vehicles. SOME/IP takes advantage of Ethernet's bandwidth, maximum frame size and IP multicast capabilities to support larger messages and reduce the overhead with regard to legacy protocols such as CAN and FlexRay. SOME/IP nodes are able to dynamically discover and subscribe to available services through the SOME/IP Service Discovery protocol (SOME/IP SD), with efficient routing strategies mixing unicast and multicast communications. Though it remains to be ascertained which use-cases would most benefit from SOME/IP, it opens the door to more flexibility in automotive communications, such as the ability to dynamically add new services or migrate existing services.

Problem Definition and Contribution of the Paper

SOME/IP SD induces overhead at run-time due to the exchange of signaling and state messages between nodes, which will affect the rest of the traffic. In addition configuring SOME/IP involves setting numerous parameters which have some impact on the service subscription delays and the additional network load. The specification documents, nor, to the best of our knowledge, any studies published in the literature so far, provide practical guidelines on how to best configure and deploy SOME/IP SD. This work explores this question and aims to give insights into how to configure SOME/IP SD so as to:

- ensure system responsiveness: guarantee the maximum time it takes for the stations to subscribe to all the services they require,
- Identify, amongst all the system characteristics and SOME/IP SD protocol parameters, the ones that have the largest impact on the subscription latency. The observations drawn for the experiments of this paper are intended to help the designer come up with adequate trade-offs between network load (*i.e.*, SOME/IP SD overhead) and delays in connecting to services.

These questions are studied experimentally on a configuration kept simple on purpose, so as to enable us identify the main factors of the subscription latencies.

Related Work

The existing work on SOME/IP SD is limited since the proposal is recent [2] and is still under development and test. In addition to the specification, there is a set of overview papers and presentations about the protocol and its rationale (see [3,4,5,6]). To the best of our knowledge, the first publication addressing the temporal performances of SOME/IP SD is [1]. In the latter study, a worst-case analysis of the subscription delay is proposed, applied to a case-study and validated against simulation results and data traces recorded on a test-bed. This paper builds on [1] and explores by sensitivity analysis the impact of the configuration of SOME/IP on the worst-case subscription latency.



OUTLINE OF SOME/IP SERVICE DISCOVERY

SOME/IP is a service oriented protocol for the automotive use in which services are offered by nodes and clients can subscribe to them. SOME/IP SD, the protocol part of SOME/IP in charge of service discovery and connection management, is described in the specification document [2].

In the following, a *client* refers to an entity wanting to subscribe to a single *service*, but there can be several clients and several services on the same ECU. It should be noted that there can be also several *instances* of the same service available on different ECUs.

When a service is available, that is not *down* in SOME/IP terminology, it has to be in one of the three following functioning phases: the *initial wait phase*, the *repetition phase*, and the *main phase*. A client that is active (*i.e.*, not down) has to be also in one of these phases (see Figure 1).

A service broadcasts *offer messages* on the network. These messages include a type, a service ID, an instance ID, a version, and a time-to-live fields. Upon the receipt of an *offer message*, interested clients will subscribe to the service. However, in order to reduce the subscription time, clients can also broadcast *find messages* to notify their need for a certain service and receive *offer messages* in response.

As this is described next, there are random waiting delays in the sequence of operations of SOME/IP SD so that not all services and clients start to operate at the same time. This helps to better spread over time the network and CPU load induced by the protocol.

Initial Wait Phase

Client behavior: a client enters the initial wait phase upon the request of the application layer. It remains in this phase for a time randomly chosen between $SdClientTimerInitialFindDelayMin$ and $SdClientTimerInitialFindDelayMax$ [2, §7.7.2]. In this phase the client remains silent. If an offer of the service that the client looks for is received in this phase, the client asks to subscribe to the service and then goes directly to its main phase, skipping the end of the initial wait phase and the repetition phase.

Service behavior: a service set to available by the application layer triggers the entry into the initial wait phase. Like the client, the service remains in this phase, without sending any message, for a time randomly chosen between $SdServerTimerInitialOfferDelayMin$ and $SdServerTimerInitialOfferDelayMax$ [2, §7.6.1]. But unlike the client, any find message received in this phase is ignored by the service (*i.e.*, the service does not buffer subscription requests) and the repetition phase cannot be skipped. This has the consequence that the duration of the initial wait phase of a service is a part of the subscription latency that cannot be shortened. This asymmetry in the behavior of clients and services is noteworthy and, as discussed later, will have to be considered when setting the protocol parameters.

Repetition Phase

Client behavior: the client sends a pre-defined number of *offer* messages ($SdClientTimerInitialFindRepetitionMax$) with an exponentially increasing waiting time between successive messages. The first message is sent upon the entry in the repetition phase, then the following messages are sent with a delay equal to $SdClientTimerInitialFindRepetitionBaseDelay \cdot 2^i$. When a client receives an *offer* message from the service in the repetition phase, the client requests subscription to the service and then goes directly to its main phase.

Service behavior: The service sends an *offer* message upon its entry in the repetition phase, then the following messages are sent with an exponential increasing delay equal to $SdServerTimerInitialOfferRepetitionBaseDelay \cdot 2^i$. After a predefined number of *offer* messages in the repetition phase ($SdServerTimerInitialOfferRepetitionMax$), the server enters the *main phase*. When the service receives a *find* message from a client, it waits during a random time from $SdServerTimerRequestResponseMinDelay$ to $SdServerTimerRequestResponseMaxDelay$ [cf. 2, §7.5.3] and sends a unicast *offer* message to the sender of the *find* message.

Main Phase

In this phase, corresponding to a stable operation mode unlike what happens at the startup of the system or after the transition to a new vehicle functioning mode, the network load induced by SOME/IP SD can be reduced.

In the main phase, a client does not send any *find* message, but will answer to server's offer message if needed. The service sends cyclically offer message with a period equal to $SdServerTimerOfferCyclicDelay$ (see [2], §7.6.3). It will also answer to *find* message as in the repetition phase.

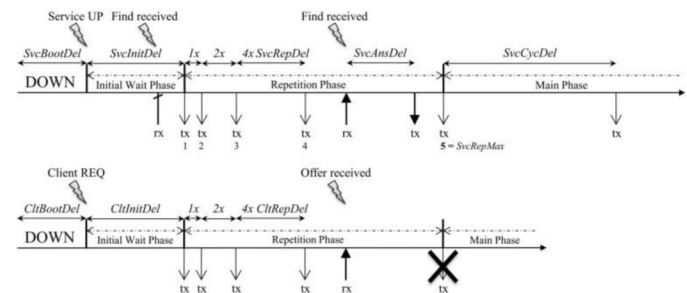


Figure 1. The startup phase of SOME/IP SD for a service and a client (figure from [2]).

Functioning Modes for Clients and Services

SOME/IP SD communication stacks allow configuring a client in two possible modes: the *request mode*, where it will send *find* messages in the repetition phase, or the *listen mode*, where it will only wait for *offer* messages. Similarly, a service can either be in offer mode, where it will send *offer* messages in the repetition phase and main phase, or in silent mode, where it will only respond to *find* messages from clients. The request mode for both the client and the server will be the focus of the paper hereafter, since it is in our understanding of the specification the normal functioning mode, and anyway the one leading to the lowest subscription latency.