Abstract: With the growth of Internet-of-Things (IoT) technology and the automobile industry, various In-Vehicle Infotainment (IVI) services have been developed, in which users can exploit a variety of IVI devices, such as navigation systems, cameras, speakers, headrest displays and heated seats. A typical IVI system is based on the peer-to-peer model, in which the user will directly control each device. This tends to induce a large overhead and inconvenience to the user. To overcome the drawbacks of the peer-to-peer model, the centralized IVI (C-IVI) scheme was recently proposed in which an IVI master is employed to provide IVI services between users and devices. However, the centralized model gives lower performance, as the number of users and devices gets larger. To improve the performance of IVI services, in this paper, we propose an agent-based IVI (A-IVI) scheme. In the proposed A-IVI scheme, a new entity called ‘agent’ is introduced, based on the C-IVI model. Each IVI agent will be used to manage a group of devices and also to perform the communication with the IVI master, on behalf of the concerned devices. The proposed scheme can be used to provide scalability and perform enhancement. The IVI agents are also helpful for supporting a variety of constrained IVI devices, such as speakers or cameras, which may usually have too low power to perform IoT communications. The proposed A-IVI scheme is implemented by using the IoT messaging protocols. For performance comparison with the existing schemes, we performed testbed experimentations. From the results, we see that the proposed A-IVI scheme can provide better performance than the existing IVI systems in terms of transmission delays, throughput and master’s loads. It is expected that the proposed scheme may be used effectively for IVI systems with a large number of users/devices, as seen in public transportation, such as public trains or airplanes.

Keywords: Internet-of-Thing (IoT); In-Vehicle Infotainment (IVI); master; agents

1. Introduction

Nowadays, a variety of technologies on autonomous vehicles have been extensively developed, including autonomous driving technology [1,2]. Many software companies, such as Apple, Google, and Microsoft, are also conducting research and development to introduce autonomous vehicles. In order to realize perfect autonomous driving, connected technology that connects the car and all things is necessary. As such, the development of mobile communication technology is rapidly changing the paradigm of the automobile industry. In particular, In-Vehicle Infotainment (IVI) services have been noted as one of the key services in the automobile industry [3–6]. In the near future, people will be able to watch some virtual reality (VR) movies through the streaming service provided in the vehicle. The current IVI platforms tend to focus on the communication inside a vehicle, but interworking with an external network also needs to be considered, together with the Internet-of-Things (IoT) communication and security issues [7]. A wide variety of IoT services, such as smart healthcare,
may be integrated with the IVI system. For example, the status of the driver can be monitored by using bio-monitoring devices and delivered to IVI users [8].

The conventional IVI system is based on the peer-to-peer model, in which the vehicle driver (owner) will directly control all devices in the vehicle, such as navigation, black-box, and camera, as shown in Figure 1. This tends to induce large overhead and inconvenience to the user. Moreover, when many users in the vehicle want to use the IVI devices, it is difficult for other users, except the driver, to use and control those IVI devices. In particular, in the ‘car-sharing’ service, which is currently growing, someone other than the owner of the vehicle may become the driver. In this situation, there is a risk that a user who is not the owner of the vehicle may change the environment settings or system settings of the vehicle.

**Figure 1.** Conventional peer-to-peer In-Vehicle Infotainment (IVI) system.

To overcome the drawbacks of the peer-to-peer model, the centralized IVI (C-IVI) scheme was recently proposed [9], in which an IVI master is employed to provide IVI services between users and devices. However, the centralized model may give lower performance, as the number of users and devices gets larger. In addition, there are some limitations to reliably connecting and efficiently managing many heterogeneous IVI devices that were made by different vehicle manufacturers according to their own infotainment ecosystem. For example, in case of a brand-new autonomous car, there may be many users and a wide variety of IVI devices, such as smart phones, speakers, music/video players, heaters, air conditioners, navigation, lights, head-mounted displays, seats, and so on. If these devices are used by many users at the same time, a large amount of traffic will be generated, and this may induce performance degradation in the C-IVI system.

To address these problems, in this paper, we propose the Agent-based IVI (A-IVI) system. The A-IVI is designed to provide several advantages: (1) IVI agent manages a group of IVI devices (called a cluster) by clustering IVI devices according to the communication method. This can reduce the load given to the IVI master; (2) Even if the IVI device is not compatible with the IVI master, the IVI master can be accessed through the IVI agent that supports the communication method, so the vehicle owner can freely add or remove a new IVI device, as needed; (3) We can provide more efficient communication than the existing IVI system by applying the cluster-based Constrained Application Protocol (CoAP) to the IVI agent node [10]. By using multiple Uniform Resource Identifiers (URIs) of a cluster-based CoAP scheme, commands can be delivered to multiple IVI devices with one message. This gives some advantages from the viewpoint of network congestion and bandwidth usage in the low-power vehicle networks.

This paper is organized as follows. In Section 2, we summarize some representative IVI systems/services developed so far by industry and also describe the existing centralized IVI (C-IVI)
system. In Section 3, we describe the proposed Agent-based IVI (A-IVI) system in details. Section 4 describes the implementations of the proposed A-IVI system by using the IoT protocols. Section 5 discusses the testbed experimentations for performance comparisons with the existing IVI system models. Finally, Section 6 concludes this paper.

2. Related Works

2.1. IVI Systems/Services by Industry

Many automakers and information technology (IT) companies have so far developed a variety of IVI systems and services, as summarized in Table 1.

| Company         | IVI System       | Description                                             |
|-----------------|------------------|---------------------------------------------------------|
| Mercedes-Benz   | MBUX [11]        | AI assistant Predictive user experience based Touch screen/pad |
| BMW             | iDrive [12]      | Status monitoring Driver assistance and Emergency service Remote diagnosis service |
| Audi            | Audi Connect [13]| 4G/LTE wireless communication Google earth User pattern deep learning |
| Hyundai         | BlueLink [14]    | Remote control and Safe security Car management Navigation |
| Google          | Android Auto [15]| AI assistant (Google NOW) Navigation with learning features Automotive entertainment app |
| Apple           | Car Play [16]    | AI assistant (Siri) Voice message Dial control support |
| Microsoft       | Windows in the Car [17] | AI assistant (Cortana) MirrorLink |
| Samsung         | Digital Cockpit [18] | AI assistant (Bixby) Smart things Digital cluster and Circular UX |
| Naver           | AWAY [19]        | AI assistant (Clova) Driver friendly UX Streaming media service |

The most technologically prominent companies in the IVI system market are the IT companies, which are Google and Apple. Automakers only provide IVI systems with functions such as vehicle status monitoring and driver assistance, but IT companies quickly support the IVI system with entertainment elements through their existing mobile research experience.

Automakers (such as Mercedes-Benz, BMW, Audi, and Hyundai) basically provide an IVI system that provides a user-friendly user interface (UI), such as a touchpad and diagnostic system that can check the status of the car. In the case of Mercedes-Benz and BMW, simple AI assistant functions can be performed, but it is difficult to perform complex tasks due to the characteristics of vehicles that cannot use the Internet in a good environment.

Google Android Auto and Apple CarPlay are now installed in vehicles from many automakers around the world. When connected to a smartphone supporting its Operating System (OS), various
services (such as AI assistant, voice control, navigation, music, and video streaming) can be used. It is equipped with Google Now and Siri, which are AI assistants being developed in the mobile market.

These IVI systems provide many distinctive services with user-friendly interfaces for easy management. On the other hand, most of the current IVI systems are based on the peer-to-peer or C-IVI models, since there are small number of IVI devices, at most ten, in the system. Google [15] and Apple [16] partially support the C-IVI system.

2.2. Centralized IVI System (C-IVI)

To overcome some drawbacks of the conventional peer-to-peer IVI systems, the centralized IVI (C-IVI) system was proposed. In this work, the IoT-based Light-Weight Machine-to-Machine (LWM2M) framework [20] is used for communication between IVI master and IVI devices. For managing IoT devices, the application layer protocol between the LWM2M server and LWM2M client is used, which is the CoAP [21]. In particular, the IVI master was employed to manage the IVI devices more efficiently.

The C-IVI system consists of IVI master, IVI user, and IVI devices, as shown in Figure 2. The IVI user accesses the IVI master using a smartphone, and the protocol uses the HyperText Transfer Protocol (HTTP). The IVI master manages the IVI device based on the LWM2M framework, and the communication with the IVI device uses CoAP. The IVI master connects the IVI device and the IVI client and manages all IVI devices in the vehicle. IVI users can access and control the IVI device only through the IVI master.

The IVI master is a newly introduced entity that comprehensively manages and controls devices in the vehicle. Users who want to control or use the IVI device can access through the IVI master. For example, when a user wants to use a heated-seat, a command to turn on the heated-seat is transmitted to the IVI master through the user’s smartphone. The IVI master receiving the command transmits the command to the corresponding IVI device, and the device receiving the command executes the command.

In the C-IVI system, user types are classified into Car Owner, Temporary Owner, Private Client, and Public Client, as shown in Table 2. Car Owners have the characteristics of long-term use and ownership, and Temporary Owner may be a user with the car-sharing or car-rental services who has the ownership in the short-term. In addition, the clients can be classified into a Public Client with the short-term use (e.g., taxi client) and a Private Client with the long-term use (e.g., family members or friends).
The C-IVI system can provide better performance and more convenience for the users, compared to the peer-to-peer system model. However, this centralized model may give lower performance, as the number of users and devices gets larger. In this paper, we propose an agent-based IVI (A-IVI) scheme to improve the performance of IVI systems or services.

3. Proposed Agent-Based IVI (A-IVI) System

In the future, it is expected that a wide variety of IVI devices will be used in the IVI system. In this case, the overhead (or load) for communication and management given to the IVI master will get larger, as the number of devices increases in the vehicle. To address this issue, we propose a new ‘IVI agent’ over the C-IVI system. The IVI agent is used to manage a cluster (group) of IVI devices with the similar characteristics. Each IVI device must be registered with the IVI agent, and the IVI agent will communicate with the IVI master, on behalf of the concerned IVI devices. However, in a certain case, the IVI device can communicate directly with the IVI master, which is discussed in Section 3.3.

3.1. System Architecture for A-IVI

The A-IVI scheme proposed in this paper consists of the IVI user, IVI device, IVI agent, and IVI master. Figure 3 shows the A-IVI system architecture. The IVI agent is newly introduced for control and management of the IVI devices within the same cluster, such as sensors, devices, and contents.

![Figure 3. Agent based IVI (A-IVI) system architecture.](image)

As described in Section 2, IVI users (owners or clients) may have different rights to control IVI devices, as per the user type. The IVI user can access the IVI master with the user’s smartphone and transmit a control message or data request message. The IVI master manages the information of the IVI devices by communicating with the IVI agent. Basically, the IVI master receives requests or information from the IVI device through the IVI agent. However, when there is a lot of data usage such as mass transfer or video streaming, the IVI master may communicate directly with the IVI device, not through the IVI agent. The IVI device refers to all devices inside the vehicle, such as sensors, navigation to help driving, a black box, and air-conditioners, heated-seats, and speakers within the vehicle. Figure 4 shows a possible configuration of A-IVI system components, based on Figure 3, in the example of public transportation vehicles.
The IVI agent can also be used to support IVI devices that cannot support the IoT protocol (e.g., HTTP or CoAP) in IVI environment due to low power. As such, the proposed A-IVI system can support many heterogeneous IVI devices, with the help of the IVI agent. For example, an IVI agent may be used to support the low-power IVI devices that can use only Bluetooth, but not CoAP or HTTP. In this case, the IVI agent will communicate with the IVI master by using HTTP or CoAP, on behalf of those IVI devices.

In addition, the IVI agent may cluster (or group) a set of IVI devices, according to the communication method, and manage these devices in the same cluster. In the initial phase, each IVI device will be registered with an IVI agent, and the IVI device information will be delivered to the IVI master by the IVI agent. During the IVI service provisioning, the status information of IVI devices is also transmitted to the IVI agent periodically or by the request of the IVI agent. Such information will be further delivered to the IVI master. Accordingly, the IVI master does not need to directly manage the status and information of all IVI devices. Instead, the IVI master will process the aggregated device information that is provided by IVI agents. This will be helpful to reduce the processing load of the IVI master.

3.2. A-IVI Functions

IVI functions are divided into six functional blocks: registration, authentication management, device control, device monitoring, profile management, and contents delivery, as shown in Figure 5.

3.2. A-IVI Functions

IVI functions are divided into six functional blocks: registration, authentication management, device control, device monitoring, profile management, and contents delivery, as shown in Figure 5.

The IVI agent can also be used to support IVI devices that cannot support the IoT protocol (e.g., HTTP or CoAP) in IVI environment due to low power. As such, the proposed A-IVI system can support many heterogeneous IVI devices, with the help of the IVI agent. For example, an IVI agent may be used to support the low-power IVI devices that can use only Bluetooth, but not CoAP or HTTP. In this case, the IVI agent will communicate with the IVI master by using HTTP or CoAP, on behalf of those IVI devices.

In addition, the IVI agent may cluster (or group) a set of IVI devices, according to the communication method, and manage these devices in the same cluster. In the initial phase, each IVI device will be registered with an IVI agent, and the IVI device information will be delivered to the IVI master by the IVI agent. During the IVI service provisioning, the status information of IVI devices is also transmitted to the IVI agent periodically or by the request of the IVI agent. Such information will be further delivered to the IVI master. Accordingly, the IVI master does not need to directly manage the status and information of all IVI devices. Instead, the IVI master will process the aggregated device information that is provided by IVI agents. This will be helpful to reduce the processing load of the IVI master.

3.2. A-IVI Functions

IVI functions are divided into six functional blocks: registration, authentication management, device control, device monitoring, profile management, and contents delivery, as shown in Figure 5.

The IVI agent can also be used to support IVI devices that cannot support the IoT protocol (e.g., HTTP or CoAP) in IVI environment due to low power. As such, the proposed A-IVI system can support many heterogeneous IVI devices, with the help of the IVI agent. For example, an IVI agent may be used to support the low-power IVI devices that can use only Bluetooth, but not CoAP or HTTP. In this case, the IVI agent will communicate with the IVI master by using HTTP or CoAP, on behalf of those IVI devices.

In addition, the IVI agent may cluster (or group) a set of IVI devices, according to the communication method, and manage these devices in the same cluster. In the initial phase, each IVI device will be registered with an IVI agent, and the IVI device information will be delivered to the IVI master by the IVI agent. During the IVI service provisioning, the status information of IVI devices is also transmitted to the IVI agent periodically or by the request of the IVI agent. Such information will be further delivered to the IVI master. Accordingly, the IVI master does not need to directly manage the status and information of all IVI devices. Instead, the IVI master will process the aggregated device information that is provided by IVI agents. This will be helpful to reduce the processing load of the IVI master.

3.2. A-IVI Functions

IVI functions are divided into six functional blocks: registration, authentication management, device control, device monitoring, profile management, and contents delivery, as shown in Figure 5.

The IVI agent can also be used to support IVI devices that cannot support the IoT protocol (e.g., HTTP or CoAP) in IVI environment due to low power. As such, the proposed A-IVI system can support many heterogeneous IVI devices, with the help of the IVI agent. For example, an IVI agent may be used to support the low-power IVI devices that can use only Bluetooth, but not CoAP or HTTP. In this case, the IVI agent will communicate with the IVI master by using HTTP or CoAP, on behalf of those IVI devices.

In addition, the IVI agent may cluster (or group) a set of IVI devices, according to the communication method, and manage these devices in the same cluster. In the initial phase, each IVI device will be registered with an IVI agent, and the IVI device information will be delivered to the IVI master by the IVI agent. During the IVI service provisioning, the status information of IVI devices is also transmitted to the IVI agent periodically or by the request of the IVI agent. Such information will be further delivered to the IVI master. Accordingly, the IVI master does not need to directly manage the status and information of all IVI devices. Instead, the IVI master will process the aggregated device information that is provided by IVI agents. This will be helpful to reduce the processing load of the IVI master.

3.2. A-IVI Functions

IVI functions are divided into six functional blocks: registration, authentication management, device control, device monitoring, profile management, and contents delivery, as shown in Figure 5.

The IVI agent can also be used to support IVI devices that cannot support the IoT protocol (e.g., HTTP or CoAP) in IVI environment due to low power. As such, the proposed A-IVI system can support many heterogeneous IVI devices, with the help of the IVI agent. For example, an IVI agent may be used to support the low-power IVI devices that can use only Bluetooth, but not CoAP or HTTP. In this case, the IVI agent will communicate with the IVI master by using HTTP or CoAP, on behalf of those IVI devices.

In addition, the IVI agent may cluster (or group) a set of IVI devices, according to the communication method, and manage these devices in the same cluster. In the initial phase, each IVI device will be registered with an IVI agent, and the IVI device information will be delivered to the IVI master by the IVI agent. During the IVI service provisioning, the status information of IVI devices is also transmitted to the IVI agent periodically or by the request of the IVI agent. Such information will be further delivered to the IVI master. Accordingly, the IVI master does not need to directly manage the status and information of all IVI devices. Instead, the IVI master will process the aggregated device information that is provided by IVI agents. This will be helpful to reduce the processing load of the IVI master.

3.2. A-IVI Functions

IVI functions are divided into six functional blocks: registration, authentication management, device control, device monitoring, profile management, and contents delivery, as shown in Figure 5.
3.2.1. Registration

The registration process is required for the IVI master to manage the IVI devices and IVI users inside the vehicle. During the registration process, information and profiles of the IVI user and IVI device are stored in the IVI master. Car Owners do not need an initial registration process and can store information through an initialization step. The registration process is divided into user registration and device registration. In the case of user registration, a Private Client needs the permission of the Car Owner only at the first registration, and after that, it can automatically connect to the IVI master through the stored information. The Public Client requires the permission of the Car Owner at every registration, and the client information is deleted after a certain period of use. In the case of IVI device registration, the permission of the Car Owner is necessary, and after the IVI master establishes a connection with the IVI device, the information of the device is stored in the IVI master.

3.2.2. Authentication Management

The IVI master stores the authority information of IVI users, and thus allows only appropriate access to the user authority. The IVI master performs authority checks using the authentication key of the IVI user and prevents IVI user from arbitrarily controlling and attacking the device inside the vehicle. Table 3 shows the IVI services that can be used by the authority level in the A-IVI system.

| IVI Services                          | Level High | Level Medium | Level Low |
|---------------------------------------|------------|--------------|-----------|
| System settings                       |            |              |           |
| Device registration and deregistration| V          |              |           |
| Authority check                       |            | V            |           |
| Client registration and deregistration|            | V            |           |
| Usage of shared service               |            | V            |           |
| Usage of high-level personal service  |            | V            |           |
| Usage of medium-level personal service|            | V            |           |
| Usage of low-level personal service   |            |              | V         |

3.2.3. Device Control

Before using an IVI device, the IVI user must check the occupancy status of the device. It can check the status of IVI device through the IVI master. In the case of a device occupied by another user, whether or not it can be used depends on the authority level of the IVI user or the priority of the device. If the user can use the device, the IVI user can control the IVI device only through the IVI master.

3.2.4. Device Monitoring

In order for the IVI master to know the status of the IVI device, each device must send the latest status information to the IVI master. Device monitoring information may be periodically transmitted by each device or may be transmitted as a specific event occurs. Periodic monitoring is generated based on a timer, and event-based monitoring messages are generated when the device status changes. In some cases, the IVI master can first send a query message to the device.

3.2.5. Profile Management

To manage a stable IVI system, the IVI master stores and manages profile information such as metadata of registered IVI users and IVI devices. The profile information indicates information about the device, such as which service the device supports to the user, and whether the device can be accessed by multiple users simultaneously. The profile information may be referred to for IVI services.
3.2.6. Content Delivery

IVI provides the content delivery function for content exchange, such as multimedia data between IVI users and IVI devices through the IVI master. The content delivery function may include content delivery initialization and content delivery. The content delivery initialization is performed to check the authority level of the relevant user for the content delivery service. The error control operation can be performed to provide reliability for content transmission between the IVI device and the IVI master and between the IVI user and the IVI master. The IVI users can exchange large amounts of multimedia data with IVI devices. Content delivery may be different, depending on owner and client.

3.3. A-IVI Communication Mechanisms

3.3.1. Multiple URI Communication

The proposed A-IVI scheme uses the CoAP-based multiple URI communication mechanism so as to reduce traffic generated by numerous IVI devices. In the multiple URI mechanism, multiple device information can be transmitted or controlled in one message. The multiple URI communication is useful when the IVI master delivers control messages to multiple devices managed by a specific IVI agent, or when the IVI agent delivers multiple device status information managed by the IVI agent to the IVI master.

Figure 6 shows an operation flow of using the multiple URI mechanism. The multiple URI option is used to efficiently process messages in the following two situations. Firstly, multiple URI option is used to process the request message of the IVI user. The IVI user transmits various request messages for IVI device control to the IVI master. The IVI master does not immediately process the service request messages that are frequently used among these request messages. Instead, it will process those messages at a specific time. Then, messages to be delivered to IVI devices (managed by the same IVI agent) are transmitted as one packet including multiple URI. Secondly, the multiple URI option is used to process status messages that are periodically transmitted by the IVI device. The IVI device transmits a status report message to the IVI agent periodically, or when a change in status occurs. When the IVI agent periodically receives the message, it does not deliver the message to the IVI master. If a message indicating a change in status is received, it reports to the IVI master. The status report message is not transmitted immediately, but after a certain time has elapsed. All state change messages received during this period can be transmitted in one packet including the multiple URI option.

![Figure 6. Operation flows of multiple URI communication.](image-url)
3.3.2. Indirect and Direct Communications

The proposed A-IVI scheme supports both indirect communication and direct communication. In indirect communication, data will be exchanged between agents and users via the master, whereas in direct communication, data are exchanged directly between users and agents without using the master. Direct communication can be applied only for data transport, in particular for the real-time multimedia data transmission. Note that all control messages must be done by indirect communication using the IVI master, since the control operations are critical and thus they need the permission of the master. In this paper, we do not address the error control issue for message loss and recovery in the network, since the underlying protocol, such as CoAP, can provide its own error control mechanisms by using the CoAP CON (Confirmable) method [9,10].

Figure 7 shows the difference between indirect communication and direct communication. The use of indirect and direct communications may be determined by the type of data to be contained in the message. For example, we may use indirect communication for delivery of mission-critical or control data, since the associated information needs the permission of IVI master, whereas general user data, such as multimedia content, may be delivered by using the direct communication.

![Indirect and direct communications.](image)

**Figure 7.** Indirect and direct communications.

3.4. Operations for Registration and Data Communication

3.4.1. IVI User Registration

Figure 8 shows the operation flows of Car Owner registration. If there is no Car Owner information registered in the IVI master, the IVI master periodically broadcasts its Uniform Resource Locator (URL) information. The Car Owner who wants to register to the IVI master transmits a POST message including its information. After receiving the Car Owner’s information, the IVI master stores this information into its database and responds with an authentication key. The Car Owner transmits a message including the received authentication key to the IVI master. The IVI master confirms the received authentication key and sends the 200 OK response message so as to complete the Car Owner’s registration process.

![Operation flows for Car Owner registration.](image)

**Figure 8.** Operation flows for Car Owner registration.
Figure 9 shows the registration process of the general users, not the Car Owner. IVI user first transmits a POST message to the IVI master. Then, the IVI master will create a URL for permission and waits for permission from the Car Owner. The Car Owner transmits an approval message including the service level of the corresponding IVI user to the generated URL. After that, the IVI master transmits a message including the authentication key to the IVI user. The IVI user who received the authentication key can log in by sending a PUT message to the IVI master along with the received authentication key.

3.4.2. IVI Resource Registration

The IVI resources represent IVI agents, IVI devices. The IVI resource registration is divided into IVI agent registration and IVI device registration. Car Owners can add or remove IVI agents and IVI devices, as needed. For IVI device registration, the IVI agent registration must be done first.

Figure 10 shows the IVI agent registration process. The IVI master periodically broadcasts its information. If an IVI agent receives this broadcast message, the agent sends a PUT message to the IVI master. The IVI master then stores the IVI agent information into its database and informs the Car Owner that there is an IVI agent to register. The Car Owner sends a POST message for requesting IVI agent registration to the IVI master. The IVI master creates a resource for IVI agent management and sends a GET message for requesting a detailed specification of the IVI agent. When the IVI master receives the updated information of the corresponding IVI agent, the message is delivered to the Car Owner, and the registration process is completed.

Figure 11 shows the IVI device registration process. The IVI agent periodically broadcasts its information to the IVI devices, after registration with the IVI master. Each IVI device receives the broadcast message and tries to register with the IVI agent. The IVI agent then transmits the metadata of the IVI device to the IVI master. Upon approval of the Car Owner, the IVI master passes the device
registration permission to the IVI agent. Then, the IVI agent delivers the registered information to the IVI device, and the IVI device registration process is completed.

Figure 11. Operation flows for IVI device registration.

3.4.3. IVI Device Monitoring

The device monitoring function is used to collect the vehicle status or the sensing information. With the help of device monitoring function, the IVI master can maintain and manage the most recent profile information of IVI devices. Figure 12 shows the operation flows for IVI device monitoring. An IVI agent transmits request messages so as to obtain the status of IVI devices that are managed by the IVI master, immediately after being connected to the IVI master. The IVI device periodically transmits its information to the IVI agent. In addition, the information will be delivered to the IVI master. Note that this report can also be done, when a specific event (e.g., a status change) occurs. In this way, the IVI agent collects all the information of IVI devices within the same cluster. The collected information is periodically transmitted to the IVI master.

Figure 12. Operation flow for IVI device monitoring.
3.4.4. Indirect Data Communication

The indirect data communication between users and agents will be done by way of the IVI master. Figure 13 is the operation flow of the indirect communication of A-IVI system. Each IVI agent sends data to the master, and the master will deliver the data to the user. For this purpose, the IVI user first requests a token required for accessing the IVI agent from the IVI master. After receiving this request message, the IVI master checks the authentication information included in the request message and sends a response message along with the token value. Then, the user can transmit a command to the IVI device via the IVI master. The IVI master transmits the command to the IVI agent, and the IVI agent delivers the command to the IVI device. If there are many IVI devices attached to the IVI agent, the multiple URI communication will be performed, as described in Section 3.3.1.

Figure 13. Operation flow for indirect communication.

3.4.5. Direct Data Communication

When a large amount of data are exchanged between IVI master and IVI agents in the indirect data communication, the data processing delay may increase rapidly and the traffic will be concentrated on the IVI master. In this case, the IVI user can directly communicate with the IVI agent without using the IVI master, as shown in Figure 14.

Figure 14. Operation flows for direct communication.
4. Implementations

4.1. Testbed Configuration

In this section, we discuss the prototype implementation of the proposed A-IVI scheme. Figure 15 shows the testbed configuration, in which Raspberry Pi [22] was used as IVI user, IVI agent, and IVI device, and a desktop computer was used as the IVI master. In experimentations, the number of IVI users and IVI devices was increased by generating the corresponding processes in the testbed devices. The CoAP-based communication was implemented by using the iris framework [23] and the go-coap [24] open source libraries.

Figure 15. Testbed configuration for implementation of the proposed A-IVI scheme.

For various experimentations, the testbed network included three Wi-Fi access points (APs), which were connected via a star topology. We used an ASUS RT-AX56U as AP. The AP was connected to all IVI entities. The IVI master and IVI agent were connected with wires, whereas IVI device or IVI user can be connected by wireless media. In addition, all APs were connected by Local Area Network (LAN). Figure 16 shows an example of AP configuration, in which the bandwidth between APs was set to 5 Mbps. Two IVI agents in this testbed were connected to the APs. IVI users and IVI devices were also connected to APs using Wi-Fi.

Figure 16. Access point (AP) configuration for A-IVI testbed.
Figure 17 shows the IVI system entities for implementation. Figure 18 shows the protocol stack used for implementation of each entity in the proposed A-IVI scheme. The IVI user communicates with the other IVI entities using HTTP/TCP, whereas CoAP/UDP is used for communication among the IVI master, IVI agents, and IVI devices to support low-power devices and compatibility with the standard IoT frameworks. As shown in the figure, the IVI master must support both HTTP and CoAP to relay the IVI user’s requests to IVI devices. IVI agents must also support both HTTP and CoAP to perform direct communication with IVI users.

![IVI System Entities](image.png)

**Figure 17.** Implementation of IVI system entities.

![Protocol Stacks](image.png)

**Figure 18.** Protocol stacks for implementation of A-IVI entities.

### 4.2. IVI Agent Registration

Figure 19 shows the packet capturing result of the IVI agent registration process. The IP address “155.230.36.152” indicates the IVI master, “155.230.36.156” indicates an IVI agent, and “233.38.35.98” indicates an IVI user. In the figure, the packet number 344 shows that the IVI master broadcasts its information to find the IVI agents. Packet number 2540 shows that the IVI agent sends a POST message to the IVI master. The IVI master receiving the POST message creates a resource that the agent can access and informs the IVI agent, as shown in the packet number 2541. The IVI master that receives the PUT message, including the IVI user’s authorization information, requests the detailed information.
of the IVI agent with a GET message. The IVI agent sends a message including detailed information 
to the IVI master, and upon receiving the message, the IVI master sends a registration completion 
message to the IVI user (see the packet numbers of 2696–2698).

| No. | Time  | Source          | Destination         | Protocol | Length info |
|-----|-------|-----------------|---------------------|----------|-------------|
| 344 | 7.82   | 155.230.30.152 | 155.230.30.255     | CoAP     | 64 NOH, M3D:1, GET, TPN:81 56 89 56 01 94 86, /broadcast |
| 2580 | 22.48  | 155.230.30.152 | 155.230.30.152     | CoAP     | 160 COM, M3D:3, POST, TPN:81 56 89 56 01 94 86, End of Block #0, Agent (a) |
| 2582 | 22.48  | 155.230.30.152 | 155.230.30.152     | C2M      | 138 AX, M3D:3, 2.01 Created, TPN:81 56 89 56 01 94 86, End of Block #0, / |
| 2594 | 21.42   | 223.38.35.98   | 155.230.30.152     | HTTP     | 157 PUT /auth HTTP/1.1 (application/json) |
| 2596 | 20.45   | 155.230.30.152 | 155.230.30.152     | HTTP     | 75 COM, M3D:3, GET, TPN:81 56 89 56 01 94 86, agent-perfect89999 |
| 2597 | 23.47   | 155.230.30.152 | 155.230.30.152     | CoAP     | 149 AX, M3D:1, 2.05 Content, TPN:81 56 89 56 01 94 86, End of Block #0, / |
| 2598 | 22.48   | 155.230.30.152 | 223.38.35.98       | HTTP     | 246 HTTP/1.1 200 OK (application/json) |

Figure 19. Packet capturing results for IVI agent registration process.

4.3. IVI Device Registration

Figure 20 shows the packet capturing results during the IVI device registration process. Figure 20a shows 
the packet captures for the IVI master, and Figure 20b shows the packet captures for the IVI agent. 
Packet number 1508 means that the IVI device sends a POST message containing its information 
to the IVI agent. The IVI agent that received this message sends a POST message to the IVI master, 
as shown in the packet number 681 of Figure 20a and in the packet number 1509 of Figure 20b. The IVI 
master generates the resource URL of the device and transmits it to the IVI device through the IVI 
agent. The IVI master requests the detailed information to the device, after obtaining the permission 
from the Car Owner. At the packet number 1914, the IVI agent requests the information of the IVI 
device with a GET message, and the IVI device transmits the information of the IVI device to the IVI 
agent. The IVI agent then informs that the IVI device is registered in the IVI master with a 2.01 Created 
message, and the IVI master sends a registration completion message to the Car Owner.

Figure 20. Packet capturing results for IVI device registration process: (a) Packet capture for IVI master, 
(b) Packet capture for IVI agent.

4.4. Indirect and Direct Data Communications

Figure 21 shows the packet capturing results for indirect communication. Figure 21a shows the 
packets for the IVI master, and Figure 21b shows the packet captures for the IVI agent. The IVI user 
will request the access information of the IVI device with a GET message. The IVI master sends the 
access token of the IVI device, after checking the user’s authority. The IVI user transmits the command 
to be delivered to the IVI master by using the access token. This can be seen in the packet number 528 
in Figure 21a. Packet numbers 530 and 356 indicate that IVI master transmits the command received 
from the IVI user to the IVI agent. After receiving the command, the IVI agent delivers the command 
to the IVI device, and the IVI device sends the 2.04 message to the IVI agent. Then, the IVI agent 
transmits a 2.04 message to the IVI master.
to the IVI device, and the IVI device sends the 2.04 message to the IVI agent. Then, the IVI agent transmits a 2.04 message to the IVI master.

![Figure 21. Packet capturing results for indirect communication: (a) Packet capture in IVI master, (b) Packet capture in IVI agent.](image)

Figure 22 shows the packet capturing results for direct communication between IVI users and IVI agents without using the IVI master. In the red box of Figure 20, we can see that the IVI user requests the IVI device information directly to the IVI agent through the PUT message. The packet information of packet number 2444 confirms that the CoAP observe message was used.

4.5. Multiple URI Communication

Figure 23 shows the packet capturing results for multiple URI communication, in which the IVI agent (IP: 192.168.126.1) receives a status change message from three IVI devices. After a certain time has elapsed, the IVI agent transmits three device status change information to the IVI master (IP: 155.230.156.1), with a single message including multiple URIs. We can see that the multiple URI message that the IVI agent delivers to the IVI master includes three or more URI-path options starting...
with "/". In addition, we can see that the three options of type 100 are included together. Each field represents the length of data to be delivered to each URI.

155.230.36.152), with a single message including multiple URI's. We can see that the multiple URI message that the IVI agent delivers to the IVI master includes three or more URI-path options starting with "/". In addition, we can see that the three options of type 100 are included together. Each field represents the length of data to be delivered to each URI.

**Figure 23.** Packet capturing results for multiple URI communication.

5. Performance Analysis by Experimentation

This section describes the performance analysis for the existing and proposed IVI schemes based on real testbed experimentation. The performance analysis was conducted for three performance metrics: total transmission delay (TTD), throughput, and master load. We will analyze these performance metrics for different number of users and devices. In the basic experimental setup, IVI devices were registered with two IVI agents and one IVI master via APs, and IVI users were randomly assigned to three different APs.

5.1. Total Transmission Delay (TTD)

In this section, we discuss the testbed experimentations for performance comparisons with the existing IVI schemes. TTD means the time duration in which the packet is delivered successfully. It was noted that TTD will depend on how effectively messages are delivered and processed in the IVI system. Figures 24 and 25 show the TTDs of three candidate schemes for the different numbers of users and devices, respectively. In the experiments, the number of IVI devices was fixed to 30, or the number of IVI users was fixed to 20.
In each figure, 10 or more users or devices were employed for performance analysis, since almost the same performance was given for 10 or less users and devices. This implies that the proposed agent-based IVI scheme does not give performance gains for small scales of users and devices. However, it is noted that the proposed scheme can be used effectively for large number of users and devices, as seen in public transportation, such as trains, airplanes, etc. From the figures, we can see that the proposed system gives lower TTDs than the existing peer-to-peer and C-AVI schemes. This performance gain comes from the use of distributed processing based on IVI agents.

In Figure 24, the peer-to-peer scheme gave larger TTDs, as the number of users increased. This is because the user has to perform the connection setup process, each time a new device is used. On the other hand, in the C-IVI, the user can control other devices through the master, and thus it showed better performance than the peer-to-peer scheme. However, the C-IVI scheme provided larger TTDs
than the proposed A-IVI scheme. This is because in the C-IVI scheme the processing loads of IVI master will get larger, as the number of users increases. On the other hand, the proposed A-IVI gave the best performance among the three candidate schemes. This is because in the proposed scheme the IVI agents are employed and direct communication between IVI users and IVI agents is used for performance improvement.

Figure 25 shows the TTDs for the three candidate schemes for different numbers of IVI devices. From the figure, we see that the proposed A-IVI scheme provided the best performance. As the number of IVI devices increases, much more messages for registration and communication will be generated. However, in the proposed scheme, the number of these messages can be reduced with the help of IVI agents that support a cluster of IVI devices.

5.2. Throughput

We now compare the throughputs for the three candidate schemes to see how many requests were processed per unit time. In this experiment, to measure throughput, the volume of messages delivered between IVI users and IVI devices were measured. Figures 26 and 27 show throughput comparisons for different numbers of IVI users and devices, respectively. In the experiments, the number of IVI devices was fixed to 30, or the number of IVI users was fixed to 20. For all schemes, throughputs tended to increase, as the number of users or devices increased.

In Figure 26, we see that the proposed A-IVI scheme provided the best throughput. The performance gaps between the proposed A-IVI scheme and the existing schemes tended to get larger, as the number of users increased in the system. In the peer-to-peer system, each user is required to discover all devices and establish a connection with each device. This processing delay leads to low throughput. In the C-IVI system, the user can use the service with only a single authentication process. However, all messages must be transmitted via the IVI master. This tends to induce larger processing delay and IVI master’s loads, as the number of users increases. However, in the proposed A-IVI scheme, the users will send large amounts of data directly to the IVI agent without using the IVI master in the direct communications. This allows the proposed A-IVI scheme to reduce the loads of IVI master, and thus the proposed scheme provided the best throughput performance.

In Figure 27, we see that the proposed A-IVI scheme provided better throughput than the existing two schemes. As the number of devices increases, the number of messages generated will increase.
However, in the proposed scheme, the IVI agent can be used to reduce the messages generated by many IVI devices. This is helpful to improve the throughput performance.

![Figure 27. Throughput for different number of devices.](image)

5.3. Master Load

Master load means the volume of data sent to or from the IVI master. High master load means low performance and large processing time. In the experiments, the number of IVI devices was fixed to 30, or the number of IVI users was fixed to 20. As shown in Figures 28 and 29, as the number of users or devices increased, the master load increased for all candidate schemes. However, the proposed scheme can perform the registration, monitoring and data transmission operations with relatively less data volume by using the IVI agents and the multiple URI scheme, compared to the existing C-IVI scheme. It is noted that the performance gaps between the proposed A-IVI scheme and the existing C-IVI scheme became larger, as the numbers of users and devices increased, as shown in the figures.

![Figure 28. Master load for different number of users.](image)
6. Conclusions and Future Works

In this paper, we proposed agent-based IVI systems to provide more effective communication between IVI users, the IVI master, and IVI devices. In the proposed A-IVI system, we introduced a new entity, an IVI agent, which is responsible for a group (cluster) of IVI devices in the IVI system. The IVI master will communicate with the IVI devices via the IVI agent. With the help of IVI agents, performance can be enhanced, and the loads of IVI master can also be reduced.

By implementation of the proposed A-IVI scheme and the testbed experimentations, the proposed A-IVI scheme was compared with the existing peer-to-peer and C-IVI schemes. From the experimental results, we can see that the proposed A-IVI scheme provides better transmission delays and throughput performance than the existing peer-to-peer and C-IVI schemes, in particular, in case there are ten or more IVI devices in the vehicle.

The proposed agent-based A-IVI scheme can be used effectively for IVI systems with large numbers of users and devices, as seen in public transportation, such as trains or airplanes. However, in IVI systems with small numbers of users and devices, such as cars, the proposed scheme tended to provide almost the same performance with the existing C-IVI scheme. For further study, a more enhanced scheme needs to be developed for IVI systems with small numbers of users and devices by exploiting the features of the proposed A-IVI scheme and the characteristics of IVI devices. For example, the concept of fog computing may be applied in this future work.

Author Contributions: D.-K.C. wrote the initial manuscript; J.-H.J. and H.-B.N. revised the manuscript; S.-J.K. proofread the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Technology Innovation Program on National Standard of MOTIE (20002214).

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Coppoal, R.; Morisio, M. Connected Car: Technologies, Issues, Future Trends. *ACM Comput. Surv.* 2016, 49, 1–36. [CrossRef]
2. Swan, M. Connected Car: Quantified Self becomes Quantified Car. *J. Sens. Actuator Netw.* 2015, 4, 2–29. [CrossRef]
3. Ricci, C.P. Network Selector in a Vehicle Infotainment System. U.S. Patent Application No 20130219039A1, 22 August 2013.

4. Cox, S.A.; Marko, P.; Patsiokas, S. Remote Vehicle Infotainment Apparatus and Interface. U.S. Patent Application No. 20090075642A1, 19 March 2009.

5. Macario, G.; Torchiano, M.; Violante, M. An In-Vehicle Infotainment Software Architecture Based on Google Android. In Proceedings of the IEEE International Symposium on Industrial Embedded Systems, Lausanne, Switzerland, 8–10 July 2009; pp. 257–260.

6. Rao, Q.; Grunler, C.; Hammori, M.; Chakraborty, S. Design methods for augmented reality in-vehicle infotainment system. In Proceedings of the 51st ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 1–5 June 2014.

7. Hassija, V.; Chamola, V.; Saxena, V.; Jain, D.; Goyal, P.; Sikdar, B. A survey on IoT security: Application areas, security threats, and solution architectures. IEEE Access 2019, 7, 82721–82743. [CrossRef] [PubMed]

8. Rodriguez-Rodriguez, I.; Rodriguez, J.V.; Zamora-Izquierdo, M.A. Variables to be monitored via biomedical sensors for complete type 1 diabetes mellitus management: An extension of the “on-board” concept. J. Diabetes Res. 2018. [CrossRef] [PubMed]

9. Choi, D.K.; Jung, J.H.; Kim, J.I.; Gohar, M.; Koh, S.J. IoT-based Resource Control for In-Vehicle Infotainment Services: Design and Experimentation. Sensors 2019, 19, 620. [CrossRef] [PubMed]

10. Choi, D.K.; Jung, J.H.; Koh, S.J. Enhanced Cluster-Based CoAP in Internet-of-Things Networks. In Proceedings of the ICOIN, Chiang Mai, Thailand, 10–12 January 2018; pp. 652–656.

11. MBUX. Available online: https://www.mercedes-benz.com/en/innovation/connected/mbux-mercedes-benz-user-experience-revolution-in-the-cockpit/ (accessed on 5 June 2020).

12. iDrive. Available online: https://www.bmwusa.com/explore/connecteddrive.html (accessed on 5 June 2020).

13. Audi Connect. Available online: https://www.audi.com/en/experience-audi/models-and-technology/digital-services/audi-connect.html (accessed on 5 June 2020).

14. Hyundai BlueLink. Available online: https://www.hyundaiusa.com/us/en/blue-link (accessed on 5 June 2020).

15. Google Android Auto. Available online: https://www.android.com/auto/ (accessed on 5 June 2020).

16. Apple Car Play. Available online: https://www.apple.com/ios/carplay/ (accessed on 5 June 2020).

17. Microsoft Windows in the Car. Available online: https://www.theverge.com/2014/4/5/5585148/microsoft-windows-in-the-car-concept (accessed on 15 June 2020).

18. Samsung Digital Cockpit. Available online: https://news.samsung.com/global/samsung-pioneers-5g-based-mobility-with-launch-of-digital-cockpit-2020 (accessed on 15 June 2020).

19. Naver AWAY. Available online: https://away.naverlabs.com/ (accessed on 15 June 2020).

20. Rao, S.; Chendanda, D.; Deshpande, C.; Lakkundi, V. Implementing LWM2M in Constrained IoT Devices. In Proceedings of the IEEE Conference on Wireless Sensors (ICWiSe), Melaka, Malaysia, 24–26 August 2015; pp. 52–57.

21. Shelby, Z.; Hartke, K.; Bormann, C. The Constrained Application Protocol (CoAP). 2014. Available online: https://iottestware.readthedocs.io/en/master/coap_rfc.html (accessed on 15 June 2020).

22. Raspberry Pi. Available online: https://www.raspberrypi.org/ (accessed on 12 May 2020).

23. Iris. Available online: https://github.com/kataras/iris/ (accessed on 1 June 2020).

24. Go-Coap. Available online: https://github.com/go-ocf/go-coap/ (accessed on 1 June 2020).