

Article

A Joint Reliable Transport Strategy in Internet of Vehicles

Zhaoxu Wang , Huachun Zhou, Bohao Feng * and Yuming Zhang 

School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China

* Correspondence: bhfeng@bjtu.edu.cn; Tel.: +86-010-5168-4319

Received: 29 July 2019; Accepted: 20 August 2019; Published: 23 August 2019



Abstract: Internet of Vehicles (IoV) is promising in bringing various data services from the traditional Internet to vehicle networks. Therefore, a reliable transport service in IoV needs to cross multiple heterogeneous networks with quite different characteristics. However, no single transport protocol is able to cope with such complex scenarios comprehensively with efficient data transmission all the way. To this end, we provide a solution named the joint reliable transport strategy (JR-TS) that selects different transport protocols on demand based on various scenarios, and builds an entire end-to-end route by linking all these transport protocols head to tail. Currently, JR-TS has already included three types of transport protocols to adapt to three typical network scenarios. With the proper implementations and settings, JR-TS can improve end-to-end transport performance and cache capacity efficiency more effectively than any single transport protocol.

Keywords: Internet of Vehicles; reliable transport; dynamic network topology

1. Introduction

With the requirements for mobile service growing, the current Internet of Things (IoT) [1] must adapt to more dynamic environments. As a promising scenario of IoT, Internet of Vehicles (IoV) [2–5] has been identified as a key technology for enhancing road safety and transport efficiency [6,7]. IoV communications among network devices are expected to happen anytime, anywhere for any related services, and, generally, these communications are in a wireless, autonomic, and ad hoc manner; related services become much more mobile, decentralized, and complex [1]. The IoV routers are often implemented on the vehicles, making them highly dynamic in both locations and velocities. Therefore, the reliable data transport services face high packet loss and frequent link interruptions. How to reliably and quickly transport data in IoV, has become a valuable issue on the IoV commercial services.

Currently, data transport researches on IoV focus on tiny, sliced message broadcasting [8,9], link and channel level optimization for a particular type of service [10–13], or congestion control issues [14–16]. However, the growing user requirement for a variety of services makes a huge volume of traffic that should be delivered among vehicles and their immediate surroundings in real-time. Therefore, firstly, a reliable transport protocol (TP) for IoV is needed to satisfy the requirement on large flow transmission. Secondly, IoV involves many manufacturers, spans multiple environments, and it differs widely in application scenarios and user requirements [1]. The reliable TP needs to be built upon the network protocol layer, in order to be backward compatible to multiple physical conditions. Thirdly, the highly dynamic IoV environment requires large on-path cache capacity to maintain the transport performance. Therefore, the IoV infrastructure should be able to afford the cache consumption of the chosen reliable TP.

Although we already have multiple types of TPs, we still lack of a highly adaptive transport solution in the challenging IoV environment. The solution requires achieving high-throughput, low latency, high cache efficiency, strong packet-loss robustness, and strong interruption robustness at the

same time. Most of the time it is a trade-off issue among the main requirements of throughput, latency, cache consumption, and robustness. As a result, we start to seek for a jointly designed transport strategy. We think of dividing a complicated IoV environment into multiple areas. The concept, area, is used to represent one or multiple adjacent links sharing similar physical states. The physical link states include delay, bandwidth, packet loss rate, interrupt frequency, and the queue cache capacity. Hence, the routers in the same area can use the same suitable TP to achieve better local performance. With all the areas achieving better performance connected together, the entire IoV can achieve satisfactory global performance.

Therefore, in this paper, we propose a joint reliable transport strategy (JR-TS) to deal with complicated IoV environment, aiming to achieve balance between transport performance and cache efficiency. To deal with the heterogeneous and dynamic nature of IoV, JR-TS makes joint usage of multiple current TPs. First, according to the specific link environment such as the link delay, packet loss rate, and link stability, JR-TS separately analyzes each area and implements an appropriate TP to fit this area. Then, JR-TS connects multiple areas into a long route by linking their TPs into a single reliable transport flow. Therefore, JR-TS has the ability to provide end-to-end data services across multiple heterogeneous IoV environments. JR-TS can achieve three main purposes: high end-to-end transport performance, controllable cache consumption efficiency, and high adaptiveness to dynamic network topology changing.

In order to check the challenging environment adaptiveness of the current TPs and test the effectiveness of JR-TS, we implement a simulated IoV platform with highly dynamic link states. The test results show that the high dynamics of the network deeply effects the performance of the tested three types of current TPs which cover most of the current communication scenarios. What is more, with appropriate optimization, JR-TS can achieve better performance than any single TP type by dynamically changing the TP solutions following the environment changing.

The rest of the paper is organized as follows. Section 2 provides a brief history on the current three types of reliable transport protocols; the challenging IoV environment is also analyzed. Section 3 shows the fundamental design of JR-TS along with its protocol stack, TP linking, and TP switching. Section 4 shows the evaluated adaptivities to three typical transport environments of the three TPs. Then the efficiency of JR-TS on a simulated IoV platform is tested and proven. In Section 5, we finally give the conclusion of the paper and show future direction.

2. Background and Motivation

In this section, we provide a brief overview on the three types of TPs; then we discuss the transport challenges in the highly dynamic IoV environment.

2.1. Current Three Types of Reliable Transport Protocols

A large amount of reliable transport protocols have been proposed for variety of service environments and service requirements. Some of the transport protocols may have more scenarios to implement and others have less. But at the beginning of the design, each of these protocols were proposed mainly for a particular transport environment. Each particular transport environment shapes a particular basic control method.

When we trace to the sources of these current transport designs, we can see that they evolve from quite a few basic control models. Some of the transport protocols copy the backbone design of the traditional TCP [17]. For example, the sliding-window sending method for reliability control, and the AIMD sending rate control to avoid congestion [18]. For some of the other transport protocols, they come from the basic idea of the licklider transmission protocol (LTP) [19] in the delay-tolerant networks (DTN) [20]. For example, the hop-by-hop retransmission for reliability control and the chunk-level forwarding method for adapting to the dynamic network topology.

According to the original designs on the control model, we classify the current transport protocols into mainly three types: the EF-TP, the HC-TP, and the HF-TP. In each type, all protocols share the similar control model and adapt to similar transport environments.

2.1.1. End-To-End Flow Transport Protocol (EF-TP)

TCP/IP has deeply impact to the current Internet. The other, later proposed, transport protocols naturally borrow the basic congestion and error control model in TCP. Many transport protocols, such as ICP [21], CCTCP [22], and ICTP [23], have a similar control model as the TCP.

As shown in Figure 1a, these protocols have the end-to-end control loop and the data are transported in the form of packet flows. So we identify these protocols as the end-to-end flow transport protocols (EF-TPs).

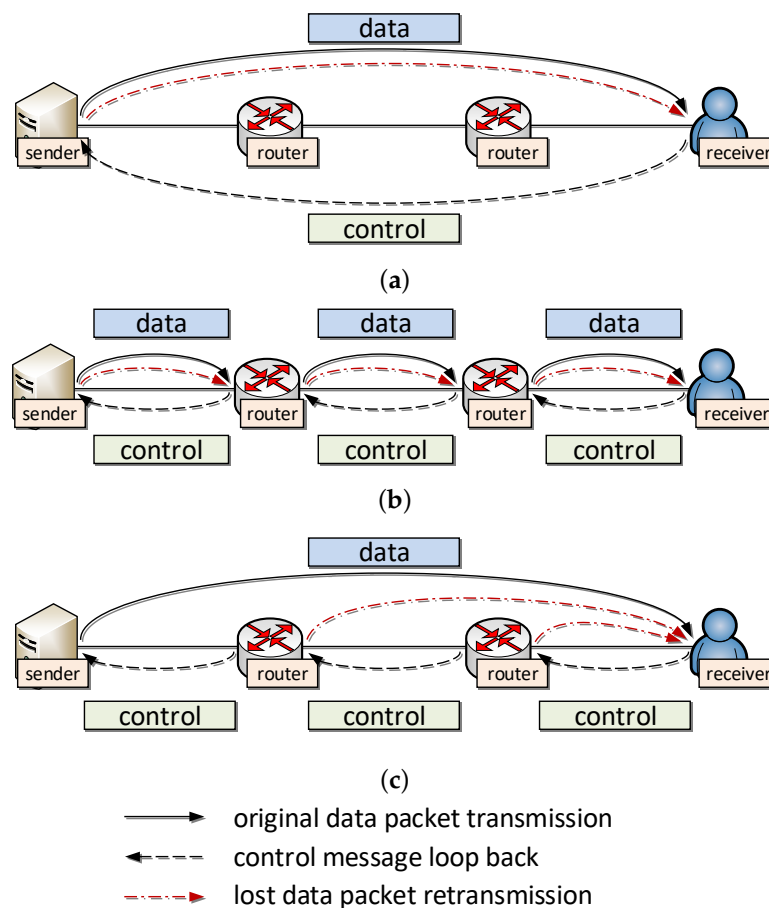


Figure 1. Control loop of three types of transport protocols. (a) End-to-end flow transport protocol (EF-TP), (b) hop-by-hop chunk transport protocol (HC-TP), and (c) hop-by-hop flow transport protocol (HF-TP).

In the EF-TPs, the transport reliability is maintained only by two end terminal nodes, the data sender and the data receiver. The intermediate routers have no responsibility in checking and recovering data error and loss. The EF-TPs generally use the sliding-window method in reliability control, AIMD window size method in congestion control, and retransmission from the data source to recover the on-path packet loss.

The EF-TPs have good real-time and low consumption in stable and low signal interference environment, like the TCP in the early-built wired Internet. However, the EF-TPs also share the similar disadvantage as the TCP. Without the cooperation of the intermediate routers, the terminal control nodes have no effective way to distinguish the congestion packet loss and the bit-error packet

loss. In the wireless IoV environment with high signal interference, the throughput of EF-TPs suffer significant deterioration. What is more, the EF-TPs rely on an end-to-end connection and a relatively stable routing list, and therefore the EF-TPs cannot work in the highly dynamic IoV environment, where the network topology keeps frequently and continuously changing.

2.1.2. Hop-By-Hop Chunk Transport Protocol (HC-TP)

An early proposed reliable transport protocol, the licklider transmission protocol (LTP) [19], is particularly designed to adapt to a highly dynamic topology network, the delay-tolerant network (DTN) [20]. The LTP is a totally novel transport protocol compared to TCP. The LTP makes data packet gathered into a larger unit called the ‘bundle’ or the ‘chunk’. In this paper, we uniformly call it ‘chunk’ for convenience. The chunk is the basic reliability control unit. Each intermediate router caches all the passing chunks and keeps them complete during the current hop transmission. When any packet is lost in the current hop, the upstream router of the current hop keeps retransmitting the lost packet until the downstream router successfully receives this packet. A data chunk is transported to the next router only when it is completely received from the previous router.

These days, the reliable transport issue to the researchers is that the mobile environment grows hotter. MFTP [24] is a typical transport protocol proposed in this background. The MFTP borrows the design of the LTP. The LTP mainly works in some special environment such as the satellite network [25]. While the MFTP directly provides the reliable transport solutions in the commercial mobile environments [26]. Similar designs as the MFTP can be also seen in the Hop [27], the HetNet [28] and the LISN [29].

As shown in Figure 1b, protocols like LTP, MFTP, and Hop have the hop-by-hop control loop and the data are transported in the form of packet chunks. So we identify these protocols as the hop-by-hop chunk transport protocols (HC-TPs).

HC-TP is particularly designed for an unstable network topology. The hop-by-hop transport method of the HC-TP has unique benefit in dealing with multihop and wireless environment. In such environments, the signal interference makes the bit-error packet loss rate high and unpredictable. Since HC-TP does not judge link congestion through packet loss, the end-to-end throughput of HC-TP is not affected by the signal interference. Even in the high packet loss rate link, the HC-TP can still achieve 90–95% usage of the link bandwidth.

However, HC-TP still has disadvantages compared to the EF-TP. The HC-TP force data transported together in chunks. Consequently, a chunk must be suspended waiting in each on-path router until all of its packets has been received from the adjacent upstream router. Such method makes the end-to-end transport delay of the HC-TP significantly higher than the EF-TP. In stable topology IoV and low bit-error packet loss links, the HC-TP has not much higher throughput than the EF-TP, but suffer much higher end-to-end delay. In fact, the HC-TP can not achieve the high throughput and the short end-to-end delay at the same time. The two performance requirements seem to be a pair of trade-offs in the choice between the EF-TP and the HC-TP.

2.1.3. Hop-By-Hop Flow Transport Protocol (HF-TP)

EF-TP has low end-to-end latency and HC-TP has high hop-by-hop throughput. Is there a combined transport protocol that achieves both performance aspects? A reliable transport protocol, R²T [30] was proposed under this issue. On one hand, R²T borrows the hop-by-hop error recovery mechanism from the HC-TP, in order to achieve high throughput in varieties of challenging network environments. Each hop caches the duplicates of all the past packets. The packet cache is prepared for retransmission to recover on-path packet loss. Each hop keeps periodically checking the packet loss. When packet loss occurs, the downstream router call for packet retransmission to the adjacent upstream router. On the other hand, R²T keeps the end-to-end data flow transported in packets, not in chunks, in order to achieve low end-to-end latency. The chunk in R²T only serves for on-path content

caching in order to provide further content service. For the end-to-end transmission, however, it is not necessary to keep chunks complete in R²T.

As shown in Figure 1c, R²T has the hop-by-hop control loop and data are still transported in the form of packet flows. So we identify protocols like R²T as the hop-by-hop flow transport protocols (HF-TPs).

The HF-TP is a comprehensive design, which borrows the advantage design from the EF-TP and the HC-TP. The architecture of HF-TP is more like adding a per hop packet retransmission mechanism onto the EF-TP. Such per hop packet retransmission mechanism ensures the packet loss is recovered locally and keeps high-throughput globally. Especially in the challenging environment where the packet loss rate is high, per hop packet loss recovery is critical to the transport performance. By trading some on-path cache consumption for better hop-by-hop throughput, HF-TP has a throughput as high as HC-TP, and also has as low end-to-end delay as the EF-TP. Currently, HF-TP only has one concrete design, R²T.

However, in the extremely unstable network topology, HF-TP transmits too quickly, and the caching mechanism may leave a large amount of incomplete chunks in the on-path cache routers. The incomplete chunks may waste storage capacity. This is to be tested and proofed in this paper.

2.2. Reliable Transport in Challenging IoV

The challenging environment becomes one of the most difficult problems for the adaptiveness of IoV. IoV mainly serves the environment filled with large numbers of high dynamic nodes, such as the vehicles and the sensors. The IoV nodes have different relative velocities and changing locations. This high dynamic causes the IoV transport environment face frequent network topology changing and the link states fluctuating. From the macro perspective, the network states are unstable and unpredictable, called the ‘challenging’ environment.

The transport protocols which may adapt to the challenging IoV environment, are mainly designed based on the on-path cache [31] capability of the routers. They use the on-path cache to provide hot content storage near the user access terminals or provide content loss retransmission on the unstable network topologies [32]. Different transport protocols deal with the challenging environments in different ways; however, these protocols, more or less, use the on-path cache to achieve better service efficiency.

In the previous research [30], we tested the three types of transport protocols. The results show that EF-TP, HC-TP, and HF-TP have significant adaptiveness differences. However, these results are tested in stable topology networks. Apparently, current test results can not proof that whether these mechanisms still have efficient performance under the highly challenging IoV environment.

3. JR-TS Overview and Design

In this section, we present the architectural design of JR-TS.

3.1. Fundamental Idea

Generally, we used to implement a single reliable transport protocol in a network architecture. For example, the TCP is implemented in the TCP/IP network while the LTP is implemented in the DTN. This is because a network architecture usually works in limited communication scenarios. The TCP/IP was designed when the network was only built by wired links, while the DTN was designed particularly for the wireless and multihop communication scenario.

IoV, however, does not particularly define the characters of the communication scenario. IoV can be implemented covering large varieties of scenarios [6], including many challenging environments [7]. The transport environment in IoV is usually random, fluctuating, and complicated. Sometimes a transport service may need to cross multiple environments which are totally different from each other. An example is shown in Figure 2. In a vehicle communication service scenario, a chunk of data needs to be transported from a static server to a moving vehicle. This data chunk

will travel across the wired network area, the wireless static network area combined with the parked vehicles [33], and the wireless dynamic network area combined with the moving vehicles.

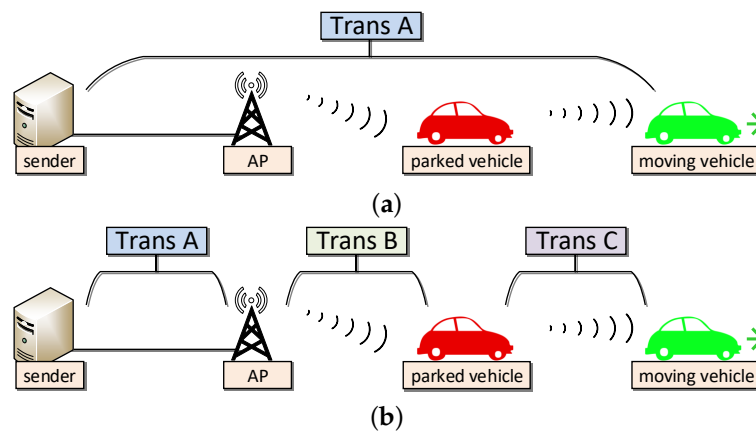


Figure 2. Transport protocol implementation in heterogeneous environments. (a) Single transport protocol. (b) Joint transport protocols.

As shown in Figure 2a, we try to choose a single transport protocol for such service. Unfortunately, there is no alternative transport protocol exists which can adapt to all these transport environments at the same time. However, if we design a joint transport strategy as shown in Figure 2b, there is a solution. We choose different transport protocols separately, and implement them in these environment areas accordingly. When a transport service requires to cross multiple environment areas, the joint strategy can provide an entire end-to-end path by linking all these transport protocols head to tail.

Following the fundamental idea of linking multiple TPs, we design JR-TS. Currently, JR-TS has already included three types of TPs: EF-TP, HC-TP, and HF-TP. JR-TS does not totally copy the entire design of these TPs, but only use the basic transport control model in their link-level implementations. In JR-TS, the three TPs define how data are formatted and controlled during the transmission. As for the end-to-end congestion control and traffic control, we borrowed the corresponding design of R²T.

JR-TS chooses TPs for each area by checking the link states and the router configurations. The link states include the packet loss rate, per hop delay and interrupt frequency. The router configurations include the size of the fast storage and the protocol implementations. As shown in Table 1, different TPs can tolerate different link states. For example, the EF-TP can not tolerate any interruption during the transmission, while the HC-TP can still work in very high link interrupt frequency (to be proven in the rest of the paper). What is more, JR-TS also considers whether the router is able to support a TP. For example, the HC-TP requires all the on-path routers to cache every entire chunk in the buffer during the transmission. So the routers without large enough fast storage cannot support the HC-TP. JR-TS comprehensively consider the limit conditions for each TP, and makes choices for each area. As an example of the practical results from our preliminary test, Table 1 also gives a qualitative summary of the relatively most adapted scenarios for the three TPs.

Table 1. Scenario example on the adaptive decision-making among three implemented TPs.

Transport Protocol	Tolerated Loss	Tolerated Delay	Tolerated Interruption	Required Buffer	Adapted Scenario
EF-TP	low	long	never	small	wired and stable topology
HC-TP	high	short	frequent	large	wireless and high dynamic topology
HF-TP	high	long	several	middle	wireless and low dynamic topology

It should be emphasized that the practical decision-makings are effected by complicated reasons, such as the user requirements, the management requirements and the network conditions. Therefore, sensing the transport environment states and calculating the exact decision-making thresholds are

complicated problems in JR-TS industrial usage. Our current research only focuses on the basic architectural design of JR-TS. Our test platform mainly relies on early experiences and manual settings. The three test scenarios shown in Table 1 are typically set and the performance differences of the three TPs are sharp and significant. This way, we achieve the preliminary effectiveness tests of JR-TS, showing its promising future.

Our proposed joint reliable transport strategy (JR-TS) has three unique designs. The first one, as presented in Section 3.2, is the session layer on top of the transport layer in the traditional network protocol stack, which manages and chooses different transport protocols to adapt to the changing transport environment. The second one, as presented in Section 3.3 and Figure 2b, is the light weight design to link multiple transport protocols head to tail to provide low-latency forwarding across heterogeneous environments. The third one, as presented in Section 3.4 and Figure 3, is the lightweight design to switch the transport protocols rapidly when the area environment changes.



Figure 3. Transport protocol switch according to environment changes. (a) Relatively static peers. (b) Relatively moving peers.

3.2. Protocol Stack

In JR-TS, each single transport protocol only serves in a part of the route, not the entire route any longer. So we need an independent protocol layer to control the entire end-to-end transport, the session layer. The session layer is on top of the transport layer, controlling the entire end-to-end transport and the on-path cache management.

Interestingly, the functionality of the session layer in JR-TS seems similar to the original OSI network protocol architectural design [34]. In the OSI, the session layer was designed to manage multiple transport protocols. In the 20th century, the session layer never came to the ground because the TCP can satisfy most of the communication requirements. Nowadays, we propose the design of the session layer once again in order to manage the three TPs in the novel IoV environments.

As shown in Figure 4, the session layer is implemented in the IoV devices which may become the terminal nodes of a service. This kind of devices includes the data servers, the data users and large amount of the on-path cache routers. The transport layer only uses one TP to control a path between a pair of logically (not physically) adjacent nodes. However, the session layer links these paths into a complete session route between the data sender and the data receiver. Apparently, one route can be combined with multiple TPs.

The session layer and the transport layer can be implemented on the IoV routers. According to the protocol stack implementations, there are three types of router in the JR-TS network: the normal router, JR-TS router, and JR-TS cache router. The normal router only has the network layer, which only supports packet forwarding but not transport controlling. The JR-TS router has the transport layer, which supports transport controlling but not chunk caching. The JR-TS cache router has the session layer, which supports chunk caching. The router with the session layer can respond to the data requests from the users, send the data chunk to the users and play the role similar to a data server.

In order to differentiate the control messages of the transport layer and the session layer, a separated namespace is required to uniquely identify every single packet in the two layers. We borrow the separated namespace of R²T to name a packet in a TP and in a service, separately [30]. We implement the locally unique identifier (LUID) for the transport layer. And we implement the globally unique identifier (GUID) for the session layer.

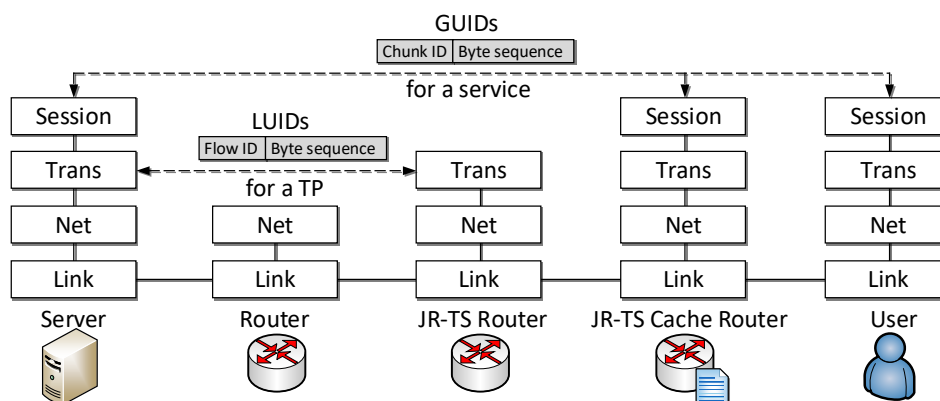


Figure 4. Protocol stack of session-based transport.

The effective ranges of the LUID and the GUID are also shown in Figure 4. The LUIDs serve a specific TP data flow in the range between a pair of logically adjacent JR-TS routers, which are both implemented with the transport layers. Once the packet identified by an LUID has been successfully transported out of the current TP, the corresponding LUID will be immediately eliminated. However, the GUIDs serve around the entire IoV network, providing the globally unique chunk identifiers and the global content services. The GUID uniquely identifies every byte of data in every cached chunk in global scale, so that the chunk can be globally requested, resolved, cached, routed, and transported.

3.3. TP Linking

Linking among TPs means that connecting different TPs head to tail together in the border router located between the different TP areas. The lightweight TP linking requires reducing the border forwarding delay as much as possible. The theoretically best performance is that the data coming from the upstream TP experience no additional latency in the border router when transformed into another heterogeneous downstream TP.

Different TPs have different requirements for rapid storage capacity in the routers:

- EF-TP. Small buffer requirement in every on-path router. Medium buffer requirement in the terminal nodes on both sides. The terminal buffer size is determined by the bandwidth-delay product (BDP) between the two terminal nodes.
- HC-TP. Large buffer requirement in every on-path router and the terminal node. The large buffer size is determined by the adjacent link BDP [30] and the transported chunk size [24].
- HF-TP. Medium buffer requirement in every on-path router and the terminal node. The medium buffer size is determined by the adjacent link BDP and the cached batch size [30].

Generally, the large storage capacity requirement occurs on the terminal nodes' buffer of the control loop. The buffer size is usually determined by the bandwidth-delay product (BDP) of the control loop. According to the buffer requirements of the three TPs above, we make every JR-TS router implemented as large rapid storage capacity as the HC-TP router.

So the linking between EF-TP and HF-TP is simpler in the border router. As shown in Figure 5a, other than the necessary packet order confirmation, the data packets coming from the upstream network interface (NIC) buffer can be linearly forwarded to the downstream NIC buffer without any additional waiting.

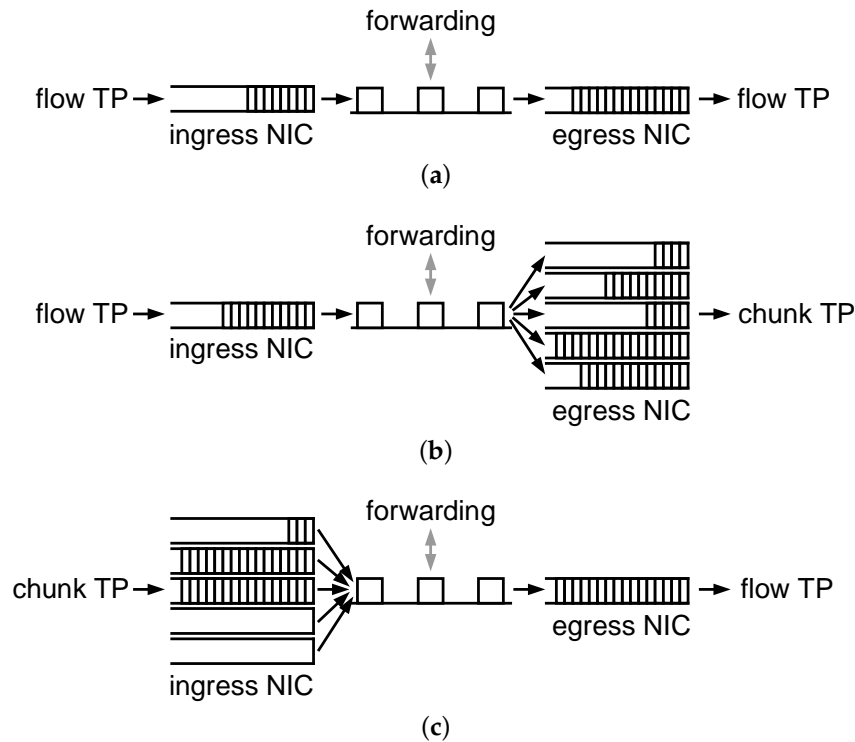


Figure 5. Linking among TPs in the JR-TS router. (a) Linking from flow TP to flow TP, (b) linking from flow TP to chunk TP, and (c) linking from chunk TP to flow TP.

However, the linking between the HC-TP and the EF/HF-TP cannot be as simple as above. The HC-TP transports data in chunks. As shown in Figure 5b, in the upstream border router of the HC-TP area, the data packets coming from the upstream EF/HF-TP area will be cached separately according to the chunks they belong. A chunk is not allowed forwarded into the downstream HC-TP area until confirmed completely received from upstream. This waiting delay is necessary and can not be eliminated. In the opposite direction as HC-TP to EF/HF-TP, as shown in Figure 5c, the router does not require maintaining the coming chunks' completeness, the packets can be forwarded to the egress queue as long as it is received.

In JR-TS, routers do forwarding for every packet. An exception is that in the HC-TP area, routers only read and decide the forward direction for the first packet of each chunk. Then the remaining packets of this chunk simply follow the first packet linearly.

3.4. TP Switching

The switching among TPs means that changing the transport protocol within an area from one TP to another TP. When the transport condition in an area has changed, the current TP is no longer adaptive to this area or there is an alternative TP which can achieve high performance, the TP switching is required. The key issue is how to guarantee the data reliability and avoid network fluctuation during the TP switching.

The three TPs have different control loop model, so the locations of their transport controllers are also different. As shown in Table 2, the EF-TP and the HF-TP have the end-to-end controller. While the HC-TP and the HF-TP have the hop-by-hop controller. The different controllers make the concrete design of the routers different. For example, the three TPs have different forwarding states of their on-path routers; the routers also have different cache states in the ingress/egress queue. Different control loops makes the switching among the three TPs may cause hard interruptions to the ongoing transport services. In general, the data loss and the performance degradation during the

switch should be avoided as much as possible. In JR-TS, the TP switching mechanism is designed according to this purpose.

Table 2. Architectural design comparison among TPs.

TP	End-To-End Controller	Hop-By-Hop Controller	Forwarding State	Ingress Queue	Egress Queue
EF-TP	Yes	No	packet-level	packet cache	packet cache
HC-TP	No	Yes	chunk-level	chunk cache	chunk cache
HF-TP	Yes	Yes	packet-level	packet cache	batch cache

Figure 6 shows the three TPs' adaptiveness to the challenging environment. We give two axes, the packet loss rate and the interrupt frequency, to identify the challenging environment. The two axes express the two main characters in TP decision-making. There are other characters including the link bandwidth, link delay and the QoS requirements. But as a preliminary research, we only focus on the two main characters which affect the TP performance mostly. From Figure 6, we can see that the higher packet loss rate and the interrupt frequency grows, the more challenging the transport environment is.

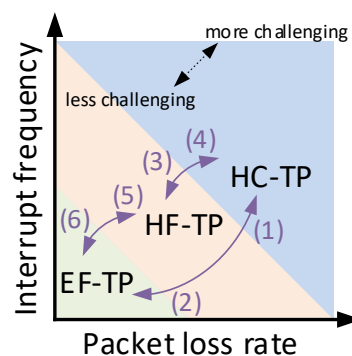


Figure 6. TP switching direction.

3.4.1. From EF-TP to HC-TP

See (1) in Figure 6. Such situations usually imply that the transport environment is becoming relatively more challenging. So the EF-TP treats the TP switching as the connection failure. Once the new HC-TP is built, all the old on-path EF-TP packets will be dropped. If possible, a NACK message is sent back to the old EF-TP terminal sender, in order to announce the transport failure and shut down the EF-TP end-to-end controller as soon as possible.

3.4.2. From HC-TP to EF-TP

See (2) in Figure 6. Such situations usually imply that the transport environment is becoming relatively less challenging. So the switching does not require too much real-time performance. The terminal sender and receiver load the EF-TP end-to-end controllers. The on-path routers switch the forwarding state from chunk-level to packet-level. Each ingress queue of the on-path routers drops all the incomplete chunks, and all the complete chunks cached in the ingress queues and the egress queues are moved up to the session layer in the router. Then all the ingress and egress queues in the router switch the queueing state from chunk-level to packet-level. The on-path routers announce the chunk existence to the old HC-TP terminal receiver. The new EF-TP uses the receiver-driven mechanism that the receiver drives the data transport; the new EF-TP terminal receiver will decide when to command these on-path routers to transport the on-path chunks to it.

3.4.3. From HC-TP to HF-TP

See (3) in Figure 6. The on-path routers load the end-to-end transport controller and switch the forwarding state from chunk-level to packet-level. All incomplete chunks suspended in the ingress queues stop waiting and all the packets in them are forwarded downstream immediately. The egress queues start to cache the duplication of all the sent packets, and also start the batch-level reliability control. For the chunks being transported in the links, the transport can continue. Because the LUID namespace is shared by both the HC-TP and the HF-TP, an HC-TP chunk can be directly treated as an HF-TP batch. Different from the chunks cached in the EF-TP on-path routers, the chunks cached in the HF-TP on-path router do not have to wait the driving command from the terminal receiver. The HF-TP flow is driven by the hop-by-hop controller. So the HF-TP flow can be continuously transported during the TP switching.

3.4.4. From HF-TP to HC-TP

See (4) in Figure 6. The end-to-end controllers are unloaded. The on-path routers switch the forwarding state from packet-level to chunk-level. The packet duplications cached in the egress queues are deleted and the chunk-level data cache are allowed. The ingress queues also start to separately cache the packets from different chunks.

3.4.5. From EF-TP to HF-TP

See (5) in Figure 6. As both are flow TPs, the switching between EF-TP and HF-TP is simple. The switching does not need interrupting the flow or dropping the packets. The LUID namespace has already been shared. So only the hop-by-hop controller has to be loaded, and the sent packet duplication caching has to be started at the egress queues.

3.4.6. From HF-TP to EF-TP

See (6) in Figure 6. Only eliminating the hop-by-hop controller and the packet duplication can achieve the switching.

4. Implementation & Evaluation

In this section, we show the evaluation to the current TPs and JR-TS.

4.1. Implementation

4.1.1. Designs and Developments

A visualized IoV transmission example for JR-TS is shown in Figure 7. The example contains three areas of transport environments: the wired network in area A, the static wireless network among parked vehicles in area B, and the dynamic wireless network among moving vehicles in area C. When a moving car requests for a data service to a server in the backbone network, there are three types of environments for the data flow to cross. According to this example, we use real physical machines to build a simulated IoV test platform. Then we use the platform to realize the IoV simulation topology.

In order to simplify the platform developing and the simulation controlling on the physical machines, we redefine two devices: the software routers and the software hubs. As shown in Figure 8, in the software routers, we run the session layer and the transport layer functionalities of the three TPs, but not any network layer functionalities, like rerouting. In the software hubs, however, we integrate the topology control, the rerouting and the link state simulating functionalities. All the input link environment characters are set by the software hubs, including the link bandwidth, the packet loss rate, the link delay [35,36] and the link interruptions [37,38]. On the other hand, all output performance data are recorded from the software routers, including the transport progress, transport delay, and cache occupancy.

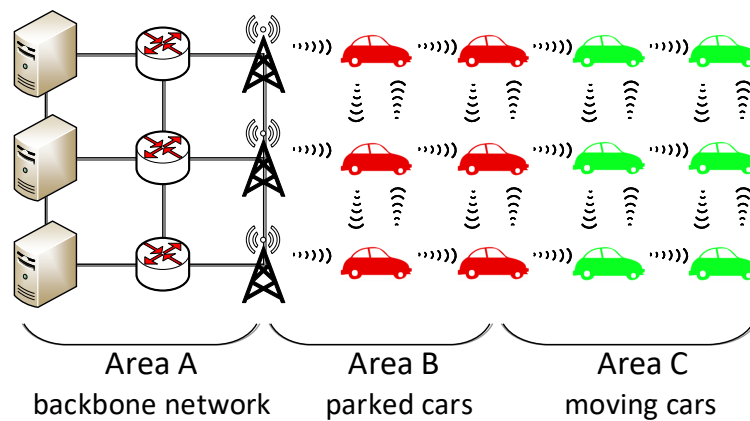


Figure 7. A service scenario example of Internet of Vehicles (IoV) that contains heterogeneous environments.

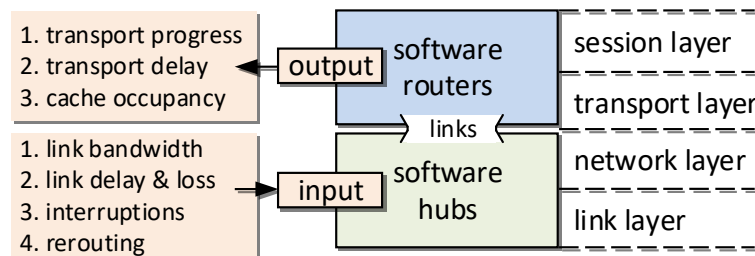


Figure 8. Designs of the IoV test platform devices.

In the evaluations, the two newly defined devices are developed into the user-level process in Linux with C codes. For the reason that our preliminary tests do not contain unpredictable environment fluctuating, the routing table is set manually for each evaluated topology. The packets are forwarded according to the GUIDs in the software hubs. And the transport control loops are controlled according to the LUIDs in the software routers.

4.1.2. Evaluated Scenario

As shown in Figure 9, in the IoV simulation topology, there are three types of environment simulation areas. Each of the area contains five hops of transport distance. In every hop, there is a *rerouting layer* in each side, consisting of three routers. The rerouting layer is built in order to simulate the link interruption which is usually caused by router moving. When a router leaves its original position, its upstream and downstream links are all interrupted. Then another alternative router belonging to the same rerouting layer takes its position. In order to easily simulate the wireless channel changes in a hop, we set the rerouting directly controlled by the software hubs rather than the software routers.

Every software router is implemented with the three TPs: EF-TP, HF-TP, and HC-TP. The link states between every two adjacent rerouting layers, including the basic link characters like the bandwidth, packet loss, delay, and the network characters like the fluctuation, interruption, are all controlled by the software hubs. We implement 45 physical machines as software routers, 15 machines as software hubs, three machines as data senders, and three machines as data receivers. The settings of each area are given below.

Area A: Normal wired network with short link delay (<1 ms), unchanged topology, uninterrupted links, and low packet loss rate. The environment is so stable that even the EF-TP which has no per hop reliability control method can have high throughput that equals 70–90% of bandwidth. In JR-TS, we manually set this area to use the EF-TP by default.

Area B: Typical wireless network with long link delay (1–10 ms), unchanged topology, uninterrupted links and high packet loss rate. In this environment, the EF-TP throughput is no

higher than 10% of bandwidth usage. Our previous tests showed that the HF-TP which has per hop reliability control is fit for such environment [30]. Therefore, in the implementation of JR-TS, we set this area to use the HF-TP by default.

Area C: Highly dynamic and unstable wireless network with long link delay, frequent link interruption and high packet loss rate. In this environment, the EF-TP can hardly work with the throughput no higher than 0.1% of bandwidth usage. The HF-TP can still transport data, but barely leave any complete on-path cached chunks. Thus, in the implementation of JR-TS, we set this area to use the HC-TP by default. However, when the link interrupt frequent is not so high, we also allow area C to use the HF-TP to gain shorter end-to-end transport latency.

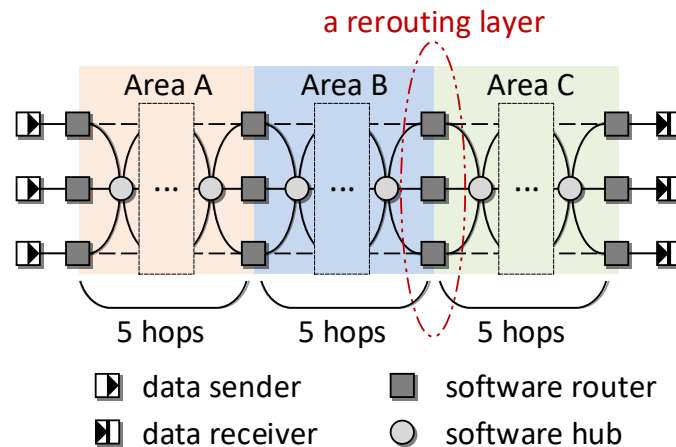


Figure 9. The IoV simulation topology built by the IoV test platform.

4.1.3. Evaluated Cases

For EF-TP, we use the basic control loop of ICP [21] as the EF-TP transport form. We cancel the hop-by-hop packet-level cache and resolution mechanism in ICP, and only discuss the basic scenario that a single data sender provides transport service to a single data receiver. We keep the receiver-driven end-to-end controller as the facilities of data requests.

For HC-TP, we duplicate the hop-by-hop control loop of the MFTP [24] as the HC-TP transport form. Data is transported in chunks in HC-TP. However, the MFTP has other corresponding traffic control and congestion control methods. In order to make the test results clearly show the basic performance of the three types of TP control loops, we choose to not duplicate these control methods. We use software-defined manual control message to centrally broadcast the route messages.

For HF-TP, we directly use the per hop process of R²T [30] as the HF-TP transport form. In each hop, the sent packets are counted and checked in batches. For the packet loss in each batch, the upstream router will retransmitted these lost packet immediately.

For the on-path content cache, if a chunk cached in a router is not complete caused by transport service unpredictably interruption, this chunk is identified as ‘the broken chunk’. The broken chunk state is not tolerable. The broken chunk can not achieve sufficient cache capacity usage, nor provide any further reliable content service. If a broken chunk cannot be complemented in a predictable period of time in the future, it should be deleted and the cache capacity should be free.

The IoV simulation topology has variety of underlying link fluctuation. As an upper layer mechanism, we translate these underlying states into the transport layer characteristics. For example, for the bit error rate caused by inter-symbol interference (ISI), we turn it into the bit-error packet loss rate in the transport layer. Only packet loss can directly effect the performance of the transport layer service.

The rest of this section is organized as follows. Section 4.2 tests the basic transport performance of the three TPs under frequent route interruptions. Section 4.3 tests the basic cache efficiency of the three TPs under frequent route interruptions. Section 4.4 tests JR-TS on the IoV simulation topology to

complete the comprehensive evaluation. Finally, we point out the flexibility and adaptivity of JR-TS in Section 4.5.

4.2. Transport Performance in Route Interruptions

Link interruption makes the routing list fail. Then all the on-path flows are shut down and the data are lost. All the incompletely cached chunks are deleted. So the link interruption causes part of the bandwidth and cache resources wasted. Different TPs have different reflection to the link interruption. So the link interruptions cause different transport latency to the TPs.

We test the transport latency of the three TPs under the simulated interruption environment. In order to clearly present the relationship between the interruptions and the transport performance in a single flow, we choose the software routers in area B and area C of the IoV simulation topology (Figure 9) to build a 10-hop route. We use the software hubs to control the random link interruptions by rerouting among the three routers in each rerouting layer. We set that the interruption happen in a manually set frequency but randomly happen on only one rerouting layer at a time. We record the transport progress and transport delay to represent the transport rate of the three TPs. The results indicate the interruption adaptiveness of the three TPs.

4.2.1. Permanent Interruption

Permanent interruption means that the data receiver is suddenly and completely interrupted from the principal part of the network. There is no other route which can connect to the receiver anymore. However, there is no guarantee that the receiver will access the network again in a predictable period.

Permanent interruption indicates the time left for a transmission process to last. We use the relation between the transport progress and the transport delay to show the transport performance of the three TPs. The transport progress represents how much data has been successfully received by the data user and cached in the on-path routers. At any particular time, we count the current transport progress, P , by the product of the on-path chunk amounts, C , and their moved hop numbers, H , which is given by

$$P = \sum C \cdot H \quad (1)$$

Also, we count the total transport progress P' , which represents the total work should be done for the final success of the entire transmission, by the product of the total amount of the chunks to be transported C' and the total hop number that the chunks need to be transported H' , which is given by

$$P' = \sum C' \cdot H' \quad (2)$$

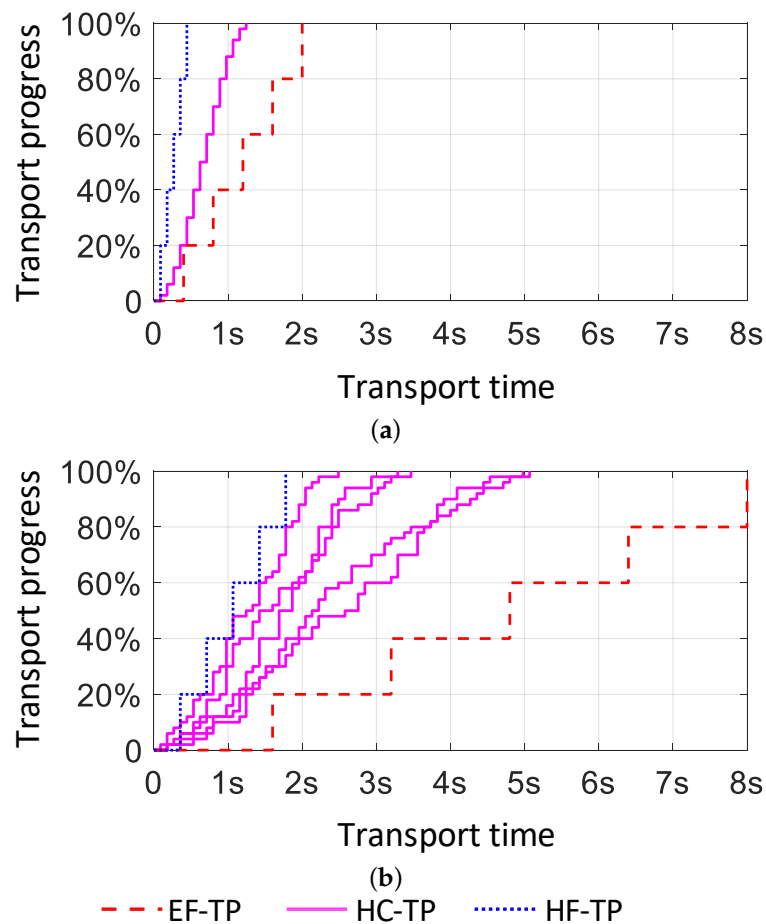
Then the percentaged transport progress p representing the ratio of the current transport progress to the total transport progress, is given by

$$p = P / P' \quad (3)$$

When no data has been sent out from the data sender, p is 0. When all the data have been successfully received by the data receiver, p is 1. When the data have not completely reached the data receiver yet, but some of the data have already been transported for several hops far from the sender, p is presented in percentage.

We set the service session as 1 MB for each chunk, 5 chunks to transport. We also set the simulated environment as the wireless link packet loss 0.1% and the link bandwidth 100 Mbps. Figure 10a shows the results from the single flow test. The whole route has no other background traffic sharing the bandwidth, so there is no concern of the network fluctuation. Results show that the HF-TP has the fastest and smoothest transport performance among the three TPs, while the EF-TP is on the opposite.

The HC-TP shows the slower transport rate both at the beginning and the end of the transport. This is because the HC-TP sends out chunks one-by-one at the beginning, and receives the chunks one-by-one at the end. During these two periods, the HC-TP's data flow does not cover all the hops of the route. Apparently, the transport progress does not reach the highest point. However, during the time in between these points, the HC-TP has almost the equal transport rate as the HF-TP. This is because the HC-TP chunks occupy the entire route and keep the same throughput as the HF-TP; the transport progress shows a good raising rate just like HF-TP.



Attention:

At each step point on the stair-shaped curves above, the transport progress value belongs to the higher point. And the lower point value is empty.

Figure 10. Transport progress in permanent route interruption. (a) Single flow in one route. (b) Multiple flow-sharing bandwidth.

Figure 10b shows the results from the shared route test. When the route bandwidth is shared by three other flows, the bandwidth sharing effect differently to the three TPs. As both are flow transport protocols, EF-TP and HF-TP show good TCP-friendly [39] performance. Their transport progresses are just pulled down to 25% of the single flow test. However, the transport progress of the HC-TP is more unstable and unpredictable. We have tested the HC-TP for multiple times; none of the two test results are completely the same. We pick five of the HC-TP results shown on the Figure 10b. We analyze the reason is that the HC-TP router queues data in chunks rather than packets. Thus, the queuing delay of a chunk fluctuates in much larger time scale than a packet. Consequently, the total end-to-end transport delay of HC-TP has more significant fluctuation than that of the EF-TP and the HF-TP.

In conclusion, we know that firstly the EF-TP is slower than the HC-TP and the HF-TP in lossy environment. Secondly, the EF-TP and the HF-TP have less network fluctuation than the HC-TP in the shared route. Third, the HF-TP has the best transport performance in the lossy but relatively stable network topology.

4.2.2. Temporary Interruption

Temporary interruption means that the data receiver may have short connection failure to the data source. The short failure can be caused by network fluctuation, signal interference burst and network topology changing. The temporary interruption causes the transport connection suspended in a short time, but not terminated.

The temporary interruption directly effects the performance reflection of the three TPs. EF-TP inherits the basic reflection of the TCP, all the unacknowledged packets later than the latest ACK message are treated as lost. HC-TP continues the per hop chunk transmission on the uninterrupted link. The HF-TP transport data in the flows. So the HF-TP is also suspended in interruption like EF-TP.

In the same test environment as above, we set the route interruption happens in the ten links randomly each time. Figure 11a shows that in a single temporary interruption, the total transport delay of the three TPs. The time counting begin since the transport begins and route interruption happens. The EF-TP and the HF-TP are directly suspended when the route is interrupted. However, most of the time the HC-TP transport is not effected immediately, until the data chunk arrives the interrupted link. Figure 11a shows ten results of the HC-TP transport delay, representing ten interrupt link locations. The interrupt location (IL) indicates that the hop number counted from the data sender to the data receiver. For example, number (1) means that the link nearest to the data sender is interrupted. Then the HC-TP transport is suspended immediately when the data transmission starts. On the opposite, when the interruption happens at the number (10) link, it will take 0.8 second to affect the HC-TP transport. If the single interruption exists no longer than 0.8 second at the number (10) link, there will be no effect to the HC-TP transport. From this result, we can see that the HC-TP has high adaptiveness to the link interruption to some extent.

As shown in Figure 11b, we set the route interrupt frequency ranged from 0 time per second to 10 times per second, and each interruption lasts for 50 milliseconds (ms). We can see that the EF-TP is the most sensitive to the frequent route interruption. The HF-TP is less sensitive, but also comes to failure when the interrupt frequency grows higher. However, even when the total route interrupt time grows to 50% of the entire transport time, the HC-TP still keeps stable transport capability in relatively lower transport delay.

The results of Figure 11b indicate that the HC-TP has higher basic transport delay than that of the HF-TP. However, the HC-TP does not require a stable connection between the sender and receiver all of the time. In the HC-TP, the route interruption time does not equal to the transmission congestion time. For example, a single link's failure only shuts down the data chunk transmission on this link; but it does not immediately affect the data chunk transmission on the other links. Hence the route interruption caused by short, frequent and random link failure does not significantly affect the transport performance of the HC-TP. This is the most useful character of HC-TP working in dynamic network topology. However, EF-TP and HF-TP are not that lucky, because they deeply rely on the entire route's integrity.

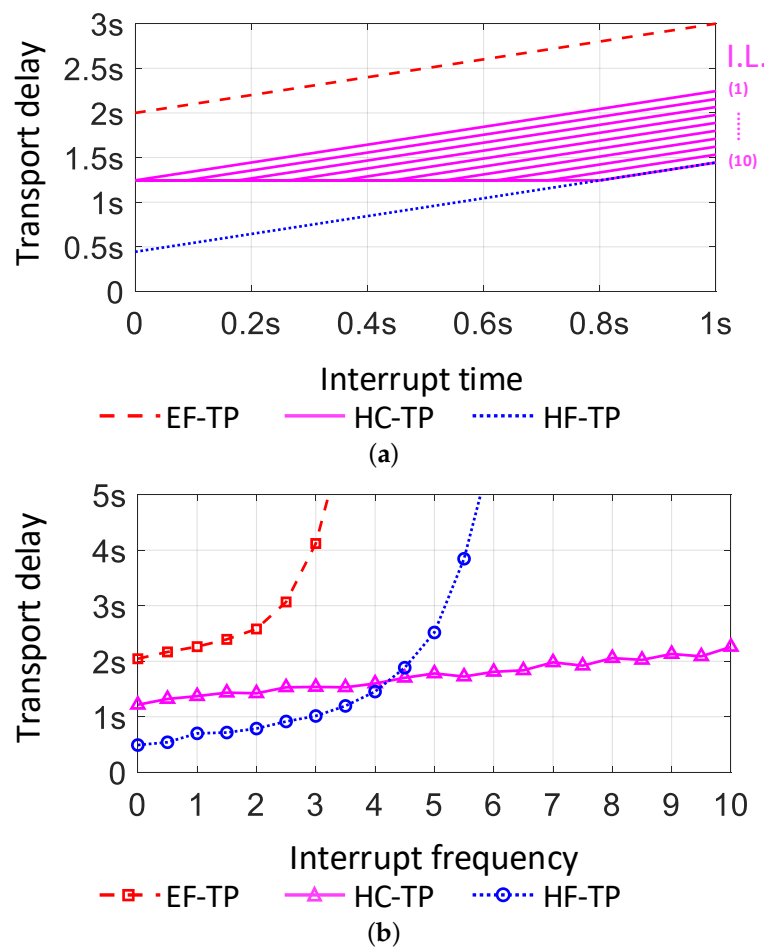


Figure 11. Transport progress in temporary route interruption. (a) Transport delay in single interruption. (b) Transport delay in continuous interruption.

4.3. Cache Efficiency in Route Interruptions

Highly efficient content services in IoV require transported data to be cached in the intermediate routers so that the duplicated data can provide further content service. As the consequence, most of the content services support the on-path data cache. When the network topology keeps changing, or the network environment keeps fluctuating, the transport services are interrupted, causing the on-path cache routers unable to gain the complete duplicates of the chunks. Frequent link interruption causes large amount of broken chunks cached in the intermediate routers.

Generally, we require a reliable content service to provide 100% complete content. Consequently, the broken chunks cached in the intermediate routers can not provide sufficient reliable content service. Once an intermediate router confirms that a locally cached broken chunk has no guarantee to be complemented in a predictable future period, the chunk should be immediately deleted. This process is set in order to quickly spare the limited cache capacity for the other cache requirements.

In this subsection, we test the cache efficiency of the three TPs. First, we test the on-path cache completeness of the three TPs as their transmissions pushing forward. Second, we test the on-path cache efficiencies of the three TPs under the different route interruption frequencies. We continue using the same test topology, test service session and environment settings as in the Section 4.2. We control the link interruption frequency in the software hubs. Further more, we record cache occupancy values from the software routers.

4.3.1. Cache Completeness

In the on-path cache routers, we identify a concept to express the cache consumption, the cache–delay product (*CDP*). We name the occupied cache capacity as *C*, the exist time of such occupancy as *D*, then *CDP* is given by

$$CDP = C \cdot D \quad (4)$$

CDP expresses both how long the cache capacity keeps occupied for and how much of the cache capacity is occupied. Once the transported data start to be cached in the on-path routers, *CDP* starts counting. The more data cached or the longer time the data keeps cached, the larger the *CDP* consumed.

In this test, we show the basic cache process of the three TPs. Using the above test environment, we transport and cache a single chunk separately in the three TPs. As shown in Figure 12, the *CDP* raising rate of the HF-TP is the fastest. The reason for this is the HF-TP has high throughput like HC-TP, and also has short end-to-end delay like EF-TP. This makes the on-path routers cache the passing data in the fastest speed. However, even all the on-path routers start caching data very early, none of them gains a complete chunk duplication until the last packet of this chunk is sent out from the sender and arrives at the on-path routers one-by-one. The cached chunk completions in all the on-path routers happen together nearby the end of the flow. Before that, if there is a route interruption happens, all the emerged *CDP* can be seen as the insufficient consumption. Similar phenomenon happens on the EF-TP, only in lower throughput and longer flow finish time than that in the HF-TP.

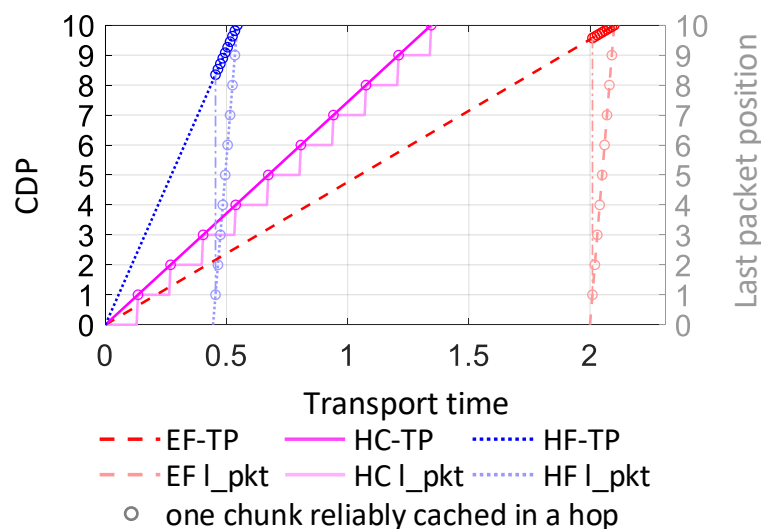


Figure 12. *CDP* occupancy and cache completeness during transport.

The HC-TP has totally different cache progress compared to the HF-TP and the EF-TP. The transported chunk in the HC-TP moves hop-by-hop. When each hop's transmission finished, a complete chunk duplication has already been reliably cached in the current router. Such complete chunk can provide secondary reliable content service immediately. So we can see that when the network interruption happens frequently, the HC-TP has the largest chance among the three TPs to leave reliable cached chunks in the on-path routers.

4.3.2. Cache Efficiency

The firstly cached but finally deleted broken chunks have occupied the cache capacity for a period without providing any service. We treat this kind of cache capacity waists as insufficient *CDP* consumption. On the other hand, if a chunk is completely cached, its *CDP* consumption is sufficient. We identify the *cache efficiency* to express the sufficient *CDP* consumption ratio in the challenging

IoV environment with frequent interruptions and fluctuations. The higher ratio the sufficient CDP consumption is, the better the *cache efficiency* is.

In order to show the cache efficiency of the three TPs in the dynamic IoV environment, we test the CDP consumptions of the three TPs. We count the sufficient CDP consumption (*sCDP*) belonging to the completed cached chunks in the on-path routers, and also the total CDP consumption (*tCDP*) belonging to the entire occupied cache capacity. We calculate the cache efficiency (*e*) for each TP by

$$e = sCDP/tCDP \quad (5)$$

The results are shown in Figure 13. As the route interrupt frequent grows, the cache efficiencies of the EF-TP and the HF-TP drop sharply. When the route interrupt frequency grows higher than a threshold, the network is no longer able to cache any complete chunk in the EF-TP or the HF-TP. However, the HC-TP is always able to keep relatively high cache efficiency, which means that there are large amount of chunks are completely cached in the on-path routers successfully.

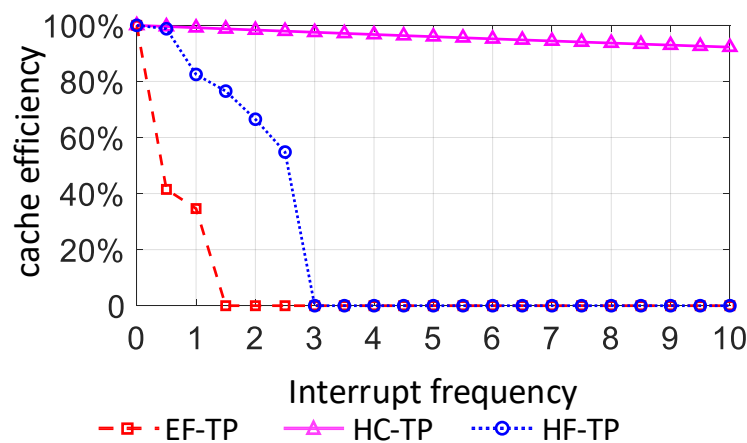


Figure 13. Cache efficiency in continuous interruption.

The results show that the HC-TP is significantly adaptive in achieving sufficient cache in the highly dynamic IoV environment.

4.4. JR-TS Performance in the IoV Simulation Topology

In the previous tests, we already know that the EF-TP is easy to implement and fit for stable, hardly lossy environment. The HF-TP has significantly high throughput in the lossy environment, better performance in dynamic environment than the EF-TP, but suffer low transport performance and low cache efficiency in highly dynamic environment. The HC-TP has longer end-to-end delay than the EF-TP and the HF-TP, but keeps stable transport performance and stable cache efficiency in highly dynamic environments.

In order to test the sufficiency and efficiency of our proposed JR-TS, we use the entire IoV simulation topology as the joint transport environment. Our hope for JR-TS is to take the advantages of variety of TPs in their adaptable areas, to connect them and to provide a larger-scale end-to-end reliable transport service across these areas. We identify the connectivity among different TPs as the sufficiency of JR-TS. Further, we identify the performance better than any single TP as the efficiency of JR-TS. So we test four test cases in this platform. They are the pure EF-TP case which implements and runs EF-TP in all the three areas, the pure HC-TP case, and the pure HF-TP case. Of course the JR-TS case is also tested, which manually implements the EF-TP in area A, the HF-TP in area B, and the HC-TP in area C, and connects them into a single transport route.

4.4.1. Joint Transport Performance

As shown in Figure 14, when areas A, B, and C are implemented with different TPs according to their particular environment characteristics, the transport delay of JR-TS is shorter than the other three pure TPs in most of the time. We can see that when the interrupt frequency in area C is smaller than average 5.5 Hz, pure HF-TP performs better than JR-TS. This is because when the dynamic of area C is not so high, the transport environment of area C is more appropriate to the HF-TP rather than the HC-TP. So the manually set JR-TS which implements the HC-TP in area C can not perform better than pure HF-TP. The dynamically-set JR-TS performance will be tested later.

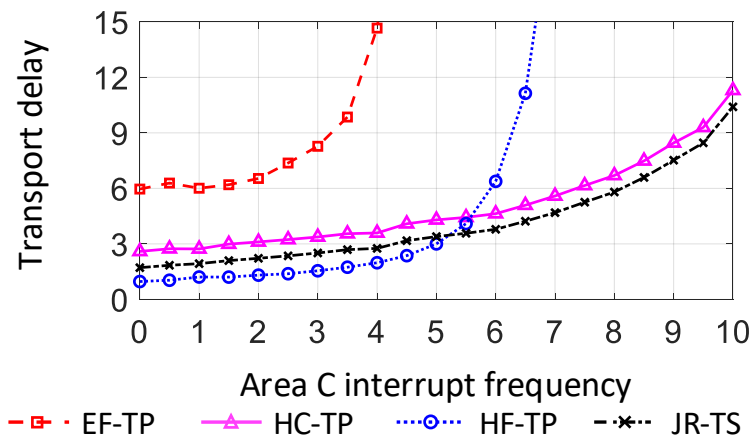


Figure 14. Transport performance in heterogeneous IoV test environment.

4.4.2. Joint Cache Efficiency

As shown in Figure 15, JR-TS has as high cache efficiency as that of pure HC-TP. Our test data shows that pure HC-TP achieves 100% cache efficiency in area A and B, only drop a little in area C; so does JR-TS.

As for the pure EF-TP and the pure HF-TP, however, they are effected badly by the frequent link interruptions. When the route interrupt frequency in area C grows higher than a threshold, the network is no longer able to cache any complete chunk in the pure EF-TP or the pure HF-TP.

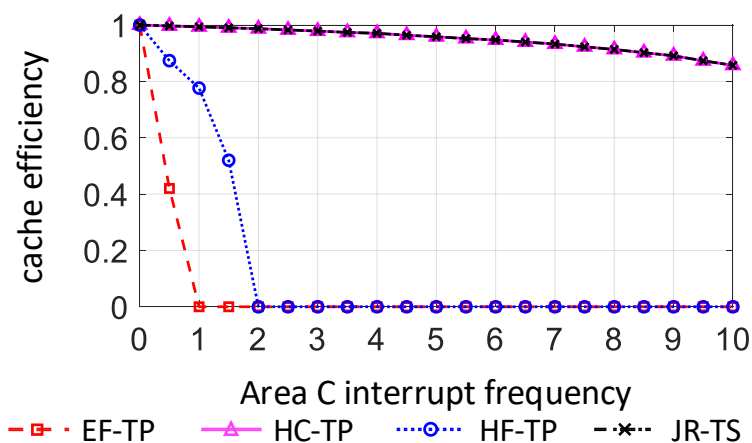


Figure 15. Cache efficiency in heterogeneous IoV test environment.

4.5. JR-TS Optimization Using Normalized Comprehensive Performance Coefficient

In practical TP settings, the network managers usually need to consider both QoS requirements and network resource consumptions. Most of the time we can not directly figure out which TP is most suitable to a network area. We need to consider the current TP's performance and decide whether it

should be switched to another TP. Therefore, a single coefficient is required to comprehensively and intuitively describe the JR-TS performance.

We set a normalized coefficient. Our purpose is to evaluate how much JR-TS can achieve high transport performance and high cache efficiency at the same time. If the results are lower than expected in the highly dynamic environment, we will switch the TP to enhance the flexibility and adaptivity of JR-TS.

We normalize the performance efficiency result data from Figures 14 and 15. We represent transport delay as D . Then, the D value is normalized to α_1 by

$$\alpha_1 = 1/(1 + D) \quad (6)$$

Then the transport delay value is transformed to the range of $[0,1]$. The larger α_1 is, the better transport performance is.

On the other hand, from the Figure 15 we find that the cache efficiency value is already in the range of $[0, 1]$. The larger the value is, the better the cache efficiency is. So we assign the value of the cache efficiency to α_2 directly. Then the comprehensive normalized performance coefficient α is given by

$$\alpha = \alpha_1 \cdot \alpha_2 \quad (7)$$

When $\alpha = 1$, it means that the transported data flow has no end-to-end delay and no cache capacity wasted. When $\alpha = 0$, it means that either the transport delay is too large to work sufficiently, or the on-path cache routers can not cache any complete chunk that the on-path cache functionality totally fails.

The performance α results of all the tested cases are shown in Figure 16. We see that when the interrupt frequent is smaller than 4 times per second, pure HF-TP implementation can achieve better performance α than our joint strategy. This result shows that if only manually set an area use a TP, our new strategy can not ensure better performance than any other single transport protocol. So at the end of this paper, we need to proof that assuming the link can dynamically decide the used TP according to the link environment changing, the new strategy can achieve better performance than any other single TP at all time.

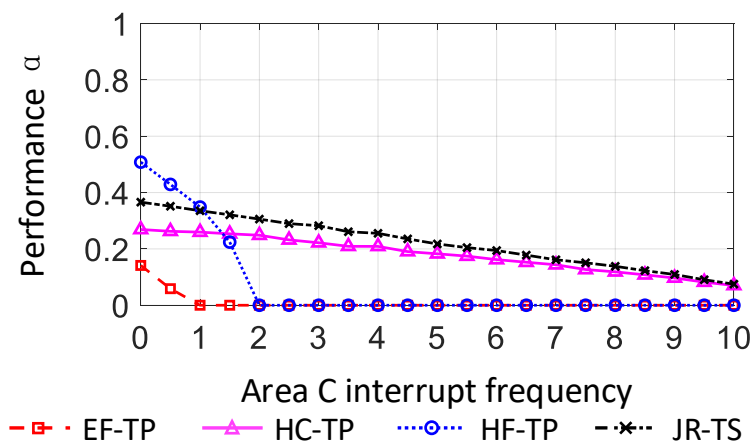


Figure 16. Normalized performance coefficient with manually set TPs.

This paper focuses on mainly two problems. The first problem is how to implement lightweight connection among different transport protocol on the router. The second problem is how to achieve lightweight TP switching on the router when the network environment changes. Thus, we do not solve the problem of how to calculate the judging threshold for the transport switching. This paper assumes that there is a sufficient software-defined judging mechanism to control the transport switching process. So according to the Figure 16, we manually set that when the interrupt frequency is lower than 4 times per second, Area C switch the TP from the HC-TP to the HF-TP. We use β to represent the normalized

comprehensive performance coefficient in the test platform where the dynamic switching mechanism is implemented on our new strategy. We run the test again, and the β results are shown in Figure 17.

The results of Figure 17 show that JR-TS with the simulated dynamic transport switching mechanism can achieve the highest comprehensive performance relatively than all the other pure TPs. So that the sufficiency and efficiency of JR-TS is proofed by now.

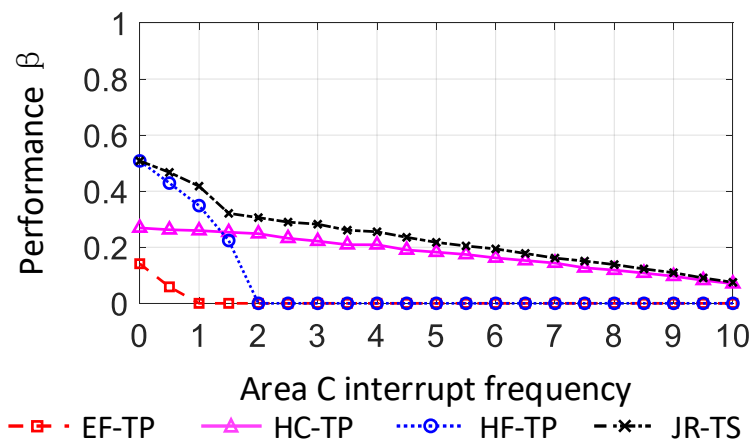


Figure 17. Normalized performance coefficient with dynamically set TPs.

5. Conclusions & Future Work

In this paper, we provide a joint strategy for the reliable transport service, JR-TS, in order to overcome the transport environment complication in IoV. JR-TS links multiple transport protocols from multiple heterogeneous scenarios, and achieves relatively better end-to-end performance.

Our future work is on the auto-transport environment analysis to support dynamic TP switching in more practical IoV services for JR-TS.

Author Contributions: Conceptualization, B.F.; methodology, Z.W.; software, Z.W.; validation, Z.W.; formal analysis, H.Z.; investigation, Y.Z. and Z.W.; resources, Y.Z.; data curation, Y.Z.; writing—original draft preparation, Z.W.; writing—review and editing, Z.W., H.Z., and B.F.; visualization, Z.W.; supervision, H.Z.; project administration, B.F.; funding acquisition, H.Z. and B.F.

Funding: This research was supported by National Natural Science Foundation of China under Grant No. 61802014 and U1530118, and National High Technology of China (“863 program”) under Grant No. 2015AA015702, and the China Postdoctoral Science Foundation under Grant No. 2018M641171.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

- Chen, S.; Xu, H.; Liu, D.; Hu, B.; Wang, H. A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective. *IEEE Internet Things J.* **2014**, *1*, 349–359. [\[CrossRef\]](#)
- Karagiannis, G.; Altintas, O.; Ekici, E.; Heijenk, G.; Jarupan, B.; Lin, K.; Weil, T. Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions. *IEEE Commun. Surv. Tutor.* **2011**, *13*, 584–616. [\[CrossRef\]](#)
- Castillo, J.C.; Zeadally, S.; Antonio, J. Internet of Vehicles: Architecture, Protocols, and Security. *IEEE Internet Things J.* **2018**, *5*, 3701–3709. [\[CrossRef\]](#)
- Cheng, X.; Zhang, R.; Yang, L. Wireless Toward the Era of Intelligent Vehicles. *IEEE Internet Things J.* **2018**, *6*, 188–202. [\[CrossRef\]](#)
- Kaiwartya, O.; Abudullah, A.H.; Cao, Y.; Altameem, A.; Prasad, M.; Lin, C.T.; Liu, X. Internet of Vehicles: Motivation, Layered Architecture, Network Model, Challenges and Future Aspects. *IEEE Access* **2016**, *6*, 5356–5373. [\[CrossRef\]](#)

6. Zhang, T.; Antunes, H.; Aggarwal, S. Defending connected vehicles against malware: Challenges and a solution framework. *IEEE Internet Things J.* **2014**, *1*, 10–21. [[CrossRef](#)]
7. Yang, J.; Fei, Z. Broadcasting with prediction and selective forwarding in vehicular networks. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 1–9. [[CrossRef](#)]
8. Bi, Y.; Shan, H.; Shen, X.S.; Wang, N.; Zhao, H. A multihop Broadcast Protocol for Emergency Message Dissemination in Urban Vehicular Ad Hoc Networks. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 736–750. [[CrossRef](#)]
9. Tian, D.; Liu, C.; Duan, X.; Sheng, Z.; Ni, Q.; Chen, M.; Leung, V.C.M. A Distributed Position-Based Protocol for Emergency Messages Broadcasting in Vehicular Ad Hoc Networks. *IEEE Internet Things J.* **2018**, *5*, 1218–1227. [[CrossRef](#)]
10. Wang, L.; Yang, H.; Qi, X.; Xu, J.; Wu, K. iCast: Fine-Grained Wireless Video Streaming Over Internet of Intelligent Vehicles. *IEEE Internet Things J.* **2019**, *6*, 111–123. [[CrossRef](#)]
11. Sayed, H.E.; Sankar, S.; Daraghmi, Y.A.; Tiwari, P.; Rattagan, E.; Mohanty, M.; Puthal, D.; Prasad, M. Accurate Traffic Flow Prediction in Heterogeneous Vehicular Networks in an Intelligent Transport System Using a Supervised Non-Parametric Classifier. *Sensors* **2018**, *18*, 1696. [[CrossRef](#)] [[PubMed](#)]
12. Zhang, Y.; Feng, B.; Quan, W.; Li, G.; Zhou, H.; Zhang, H. Theoretical Analysis on Edge Computation Offloading Policies for IoT Devices. *IEEE Internet Things J.* **2019**, *6*, 4228–4241. [[CrossRef](#)]
13. Kaiwartya, O.; Abdullah, A.H.; Cao, Y.; Lloret, J.; Kumar, S.; Shah, R.R.; Prasad, M.; Prakash, S. Virtualization in Wireless Sensor Networks: Fault Tolerant Embedding for Internet of Things. *IEEE Internet Things J.* **2017**, *5*, 571–580. [[CrossRef](#)]
14. Taherkhani, N.; Pierre, S. Centralized and Localized Data Congestion Control Strategy for Vehicular Ad Hoc Networks Using a Machine Learning Clustering Algorithm. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3275–3285. [[CrossRef](#)]
15. Lyu, F.; Cheng, N.; Zhou, H.; Xu, W.; Shi, W.; Chen, J.; Li, M. DBCC: Leveraging Link Perception for Distributed Beacon Congestion Control in VANETs. *IEEE Internet Things J.* **2018**, *5*, 4237–4249. [[CrossRef](#)]
16. Hanan, A.H.A.; Idris, M.Y.; Kaiwartya, O.; Prasad, M. Real Traffic data based Evaluation of Vehicular Traffic Environment and State-of-the-art with Future Issues in Location Centric Data Dissemination for VANETs. *Dig. Commun. Netw.* **2017**, *3*, 195–210. [[CrossRef](#)]
17. Postel, J. (Ed.) Transmission Control Protocol. In *IETF Request for Comments (RFC) 793*; Information Sciences Institute, University of Southern California: Marina del Rey, CA, USA, 1981. Available online: www.rfc-editor.org/rfc/rfc793.txt (accessed on 29 July 2019).
18. Jacobson, V. Congestion avoidance and control. *ACM SIGCOMM Comput. Commun. Rev.* **1988**, *18*, 314–329. [[CrossRef](#)]
19. Ramadas, M.; Burleigh, S.; Farrell, S. Licklider Transmission Protocol-Specification. IETF Request for Comments (RFC) 5326. 2008. Available online: www.rfc-editor.org/rfc/rfc5326.txt (accessed on 29 July 2019).
20. Cerf, V.; Burleigh, S.; Hooke, A.; Torgerson, L.; Durst, R.; Scott, K.; Fall, K.; Weiss, H. Delay-Tolerant Networking Architecture. IETF Request for Comments (RFC) 4838. 2007. Available online: www.rfc-editor.org/rfc/rfc4838.txt (accessed on 29 July 2019).
21. Carofiglio, G.; Massimo, G.; Luca, M. ICP: Design and evaluation of an interest control protocol for content-centric networking. In Proceedings of the 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS 2014), Orlando, FL, USA, 25–30 March 2012; pp. 304–309.
22. Saino, L.; Cosmin, C.; George, P. CCTCP: A scalable receiver-driven congestion control protocol for content centric networking. In Proceedings of the 2015 IEEE International Conference on Communications (ICC 2015), Budapest, Hungary, 9–13 June 2013; pp. 3775–3780.
23. Salsano, S.; Detti, A.; Cancellieri, M.; Pomposini, M.; Blefari-Melazzi, N. Transport-layer issues in information centric networks. In Proceedings of the 2nd ACM SIGCOMM Workshop on Information-Centric Networking, Helsinki, Finland, 17 August 2012; pp. 19–24.
24. Su, K.; Bronzino, F.; Ramakrishnan, K.K.; Raychaudhuri, D. MFTP: A Clean-Slate Transport Protocol for the Information Centric MobilityFirst Network. In Proceedings of the 2nd ACM Conference on Information-Centric Networking (ICN), San Francisco, CA, USA, 30 September–2 October 2015; pp. 127–136.
25. Feng, B.; Li, G.; Li, G.; Zhang, Y.; Zhou, H.; Yu, S. Enabling Efficient Service Function Chains at Terrestrial-Satellite Hybrid Cloud Networks. *IEEE Netw.* **2019**, 1–6. [[CrossRef](#)]

26. Ivan, S.; Kiran, N.; Sam, N.; Dipankar, R. Mobilityfirst future internet architecture project. In Proceedings of the 7th ACM Asian Internet Engineering Conference (AINTEC), Bangkok, Thailand, 9–11 November 2011; pp. 1–3.
27. Li, M.; Agrawal, D.; Ganesan, D.; Venkataramani, A. Block-switched Networks: A New Paradigm for Wireless Transport. In Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI), Boston, MA, USA, 22–24 April 2009; Volume 9, pp. 423–436.
28. Feng, B.; Zhou, H.; Zhang, H.; Li, G.; Li, H.; Yu, S.; Chao, H.C. HetNet: A Flexible Architecture for Heterogeneous Satellite-Terrestrial Networks. *IEEE Netw.* **2017**, *31*, 86–92. [[CrossRef](#)]
29. Feng, B.; Zhang, H.; Zhou, H.; Yu, S. Locator/Identifier Split Networking: A Promising Future Internet Architecture. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2927–2948. [[CrossRef](#)]
30. Wang, Z.; Luo, H.; Zhou, H.; Li, J. R²T: A Rapid and Reliable Hop-by-Hop transport mechanism for Information-Centric Networking. *IEEE Access* **2018**, *6*, 15311–15325. [[CrossRef](#)]
31. Chen, M.; Hao, Y.; Hu, L.; Huang, K.; Lau, V.K.N. Green and Mobility-Aware Caching in 5G Networks. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 8347–8361. [[CrossRef](#)]
32. Xylomenos, G.; Ververidis, C.N.; Siris, V.A.; Fotiou, N.; Tsilopoulos, C.; Vasilakos, X.; Katsaros, K.V.; Polyzos, G.C. A survey of information-centric networking research. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1024–1049. [[CrossRef](#)]
33. Malandrino, F.; Casetti, C.; Chiasserini, C.F.; Sommer, C.; Dressler, F. The Role of Parked Cars in Content Downloading for Vehicular Networks. *IEEE Trans. Veh. Technol.* **2014**, *63*, 4606–4617. [[CrossRef](#)]
34. ITU. *Information Technology—Open Systems Interconnection—Systems Management Overview*; ITU: Geneva, Switzerland, 1997.
35. Abboud, K.; Zhuang, W. Modeling and Analysis for Emergency Messaging Delay in Vehicular Ad Hoc Networks. In Proceedings of the 2009 IEEE Conference and Exhibition Global Telecommunications (GLOBECOM), Honolulu, HI, USA, 30 November–4 December 2009; pp. 1–6.
36. Wang, Y.; Zheng, J.; Mitton, N. Delivery Delay Analysis for Roadside Unit Deployment in Vehicular Ad Hoc Networks With Intermittent Connectivity. *IEEE Trans. Veh. Technol.* **2015**, *65*, 8591–8602. [[CrossRef](#)]
37. Wisitpongphan, N.; Bai, F.; Mudalige, P.; Sadekar, V.; Tonguz, O. Routing in Sparse Vehicular Ad Hoc Wireless Networks. *IEEE J. Sel. Areas Commun.* **2007**, *25*, 1538–1556. [[CrossRef](#)]
38. Kaiwartya, O.; Tiwari, P.K.; Kumar, S.; Prasad, M. Dynamic Vehicle Routing Solution in the Framework of Nature-Inspired Algorithms. In *Designing and Implementing Global Supply Chain Management*; IGI Global: Hershey, PA, USA, 2016; pp. 36–50.
39. Floyd, S.; Fall, K. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Trans. Netw.* **1999**, *7*, 458–472. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).