Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 1980867, 10 pages
http://dx.doi.org/10.1155/2016/1980867

Research Article

A Dynamic Linear Hashing Method for Redundancy Management in Train Ethernet Consist Network

Xiaobo Nie, Lide Wang, Baohua Wang, Biao Liu, and Ping Shen

School of Electrical Engineering, Beijing Jiaotong University, Beijing 100044, China

Correspondence should be addressed to Xiaobo Nie; xbnie@bjtu.edu.cn

Received 8 January 2016; Revised 11 April 2016; Accepted 5 May 2016

Academic Editor: Antonios Tsourdos

Massive transportation systems like trains are considered critical systems because they use the communication network to control essential subsystems on board. Critical system requires zero recovery time when a failure occurs in a communication network. The newly published IEC62439-3 defines the high-availability seamless redundancy protocol, which fulfills this requirement and ensures no frame loss in the presence of an error. This paper adopts these for train Ethernet consist network. The challenge is management of the circulating frames, capable of dealing with real-time processing requirements, fast switching times, high throughout, and deterministic behavior. The main contribution of this paper is the in-depth analysis it makes of network parameters imposed by the application of the protocols to train control and monitoring system (TCMS) and the redundant circulating frames discarding method based on a dynamic linear hashing, using the fastest method in order to resolve all the issues that are dealt with.

1. Introduction

The train control network is responsible for the transmission of control commands, state detection, and fault diagnosis. The availability of the train control network is directly related to the safety of the train operation [1–3]. With the increasing of intelligence, the existing control network WTB (Wired Train Bus)+MVB (Multifunction Vehicle Bus) cannot meet the reliable and real-time transmission of large amounts of data in the TCMS because of the limitation of bandwidth, so the high bandwidth of industrial real-time Ethernet has gradually become the development direction of the train control network [4]. In 2014, IEC published two new standards IEC61375-3-4 (Consist Ethernet Network, ECN), which is used to connect the device in one consist, and IEC61375-2-5 (Ethernet Train Backbone, ETB) which is used to connect the device in one train, in order to promote the application of industrial Ethernet in TCMS [5].

One challenge of this application is that the TCMS is considered safety-critical real-time system. It must meet at least one hard deadline. In TCMS the control commands and the sampled values must meet the deadline. If this kind of information missed the deadline due to either link brakes or information delays, it will result in severe consequences that the performance of the whole train will be seriously affected. So the TCMS requires high availability and rapid recovery time when a failure occurs. High availability is realized by the redundant management. A redundant network provides route changing when a failure occurred in the regular route. The required communication recovery time means the time duration in which a network recovers failure and the application recovery tolerated delay (or grace time) is the time duration during which the substation tolerates an outage of the automation system.

Nowadays, there are several means or protocols that provide redundancy in different network topologies. Some of these protocols are Rapid Spanning Tree Protocol (RSTP) for commercial network, whose grace time is about 1 s, and Coupled Redundancy Protocol (CRP) and Media Redundancy Protocol (MRP) for cost-sensitive industrial network, whose grace time is about hundreds of milliseconds, and the newest Parallel Redundancy Protocol (PRP) and high-availability seamless redundancy (HSR) protocol for critical real-time system, whose recovery time is zero. These protocols operate superimposed over industrial protocols [6–8].

According to the working principle of the redundancy protocol, the relationship between different application grace time and various redundancy schemes is shown in Table 1.

Table 1: The relationship between different application grace time and various redundancy schemes.

Application	Grace time	Redundancy schemes	Recovery time
Commercial network	1 s	STP	10 s
		RSTP	1 s
Cost-sensitive industrial	Hundreds of milliseconds	CRP	400 ms
		MRP	100 ms
Critical real-time systems	≤10 ms [11]	PRP	0 ms
		HSR	0 ms

According to the analysis above, most of the redundancy schemes cannot meet the demand of recovery time of less than 50 ms which is defined in IEC61375-3-4. The newest published PRP and HSR can fulfill this requirement and ensure no frame loss in the presence of a fault. With respect to PRP, HSR allows roughly halving the network infrastructure. So this paper introduces the HSR to ECN, which allows elements to be added to or removed from the network without interrupting communications and the operation of the train.

In the HSR network, each node supports the IEEE802.1 bridge functionality and forwards frames from one port to the other, except if they already sent the same frame in that same direction. According to the HSR communication theory, every frame when DANH (Doubly Attached Nodes with HSR protocol) node is required to call the discard algorithm to eliminate redundancy when needing to forward the frame.

A duplicate frame is one that arrives after a copy of the original frame arrives correctly from another port.

A circulating frame is one that arrives after a copy of the original frame arrives correctly from the same port. In a ring, they appear when a multicast frame has lost its source or when a unicast frame has lost its source and destination.

The Duplicate Discard method is not specified in the IEC62439-3. It leaves the designer free to choose a method to manage them. Jiang provides a redundancy management method based on open address hash method combined with ring memory, to meet the demand of redundancy discard rate of 100% with fixed memory size when the number of network nodes increases [9]. Xiaozhuo uses the traditional static hash table method; it mainly focused on changing the data storage method resulting in low search efficiency. Araujo provides a staging area synchronous search algorithm to improve search efficiency [10].

Based on the study of hashing algorithms, this paper proposes a dynamic linear hashing method that can effectively improve search efficiency, which can be learned from the experimental results that it can meet the performance of real-time and availability for train Ethernet consist network.

This paper starts with the introduction of the train Ethernet consist network and different redundancy scheme. Section 2 analyzes the requirements for redundancy management in ECN. Section 3 analyzes the redundant frames introduced by HSR, which must be discarded, and the requirements for frame detection. Section 4 analyzes the constraints for the redundant frame discarding algorithm and proposes a dynamic linear hashing method, and Section 5 presents the setup, configuration, result of simulation, and discussion. This paper finishes with the conclusions.

2. Requirements for Redundancy Management in ECN

2.1. Definition of Recovery Time in ECN. Redundancy is one of the least considered aspects in conventional computer networks, but it is one key factor in industrial networks. A redundant network provides route changing when a failure occurred in the regular route. The time required to find and use new routes when redundancy scheme managed at network level is applied in the ECN; recovery time of network function of ECN in case of failure is expected to be shorter than the time during which operations of the consist can be maintained without loss of train application functions.

Recovery time in ECN will include

(i) time for detection of failure,

(ii) redundancy switch over time,

(iii) time for reconfiguration in ECN, if occurred.

2.2. Requirements for Redundancy Managed at Network Level in ECN. Redundancy managed at network level, adopting network topology which has redundancy in it, is a method to recover network function in case of failure at network components.

Requirements for redundancy managed at network level are as follows [12]:

(i) When application needs redundancy in ECN, redundancy scheme will be possible to be applied to ECN.

(ii) When redundancy scheme is applied to ECN, a single network component failure will not prevent the rest of the network from working without separation of the network so that the application can maintain its function.

(iii) The recovery time of network in case of failure depends on the redundancy scheme of the ECN, which will be possible to be estimated for the ECN.

(iv) The recovery time will be less than 50 ms for TCMS (Train Control and Management System) application.

(v) When a network component comes up late or a network component comes up again (reboot), connectivity loss duration time caused by the reconfiguration of

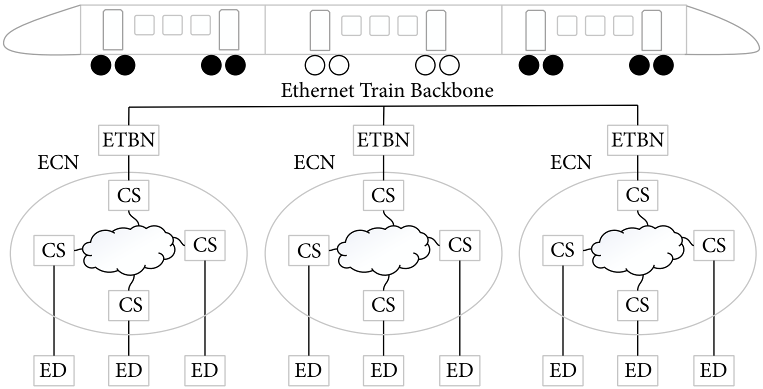


Figure 1: Topology of ECN proposed by IEC61375-3-4.

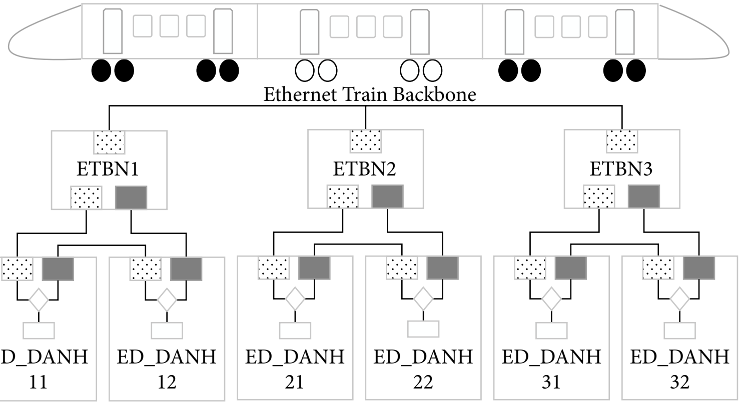


Figure 2: Topology of ECN with redundancy.

the network will be equal to or less than the value of the recovery time requirement.

(vi) Forwarding loop will not be formed in any time to avoid broadcast storm, for instance.

3. Redundancy Management of HSR in ECN

Depending on the flow of network node data frame as separate virtual network structure, it forms a high-availability seamless redundancy (HSR) network. The HSR network requires that all nodes must be "interchangeable type terminal node." There are two links within the network, the single point of failure with no interference resilience, and high reliability. Its outstanding feature is no external switch, which will significantly reduce the network investment.

3.1. Topology of ECN-HSR. The topology of Train Consist Ethernet Network without redundancy management is shown in Figure 1. ECN interconnects End Device (ED) located in one consist. An ECN consists of Consist Switches (CS), connectors, cables, and optionally repeaters and transmits data frames between ED and between ED and ETBN. The ETBN (Ethernet Train Backbone Node) which the ECN attaches will provide a gateway function which provides data transfer between the ECN and the Ethernet Train Backbone.

According to characteristics of ECN and HSR, the proposed ECN topology in this paper is shown in Figure 2.

The increase in availability involves a number of costs. The introduction of HSR into ECN also has a number of drawbacks, that is, the delay caused by each node in the ring, and the halving of the effective bandwidth. The computation requirements for the HSR nodes are higher because of processing HSR frames, removing duplicates and circulating frames, and switching all the traffic.

3.2. The Principle of Redundancy Management of ECN with HSR. HSR are based on sending two copies of every frame via two independent paths, so that if one of them is lost, the other one arrives and there is no break in communication. The redundancy that has been introduced is in the link layer of the open system interconnection reference model. These protocols add a link redundancy entity, which manages protocols, functions, and frames transparently for the other layers, with which the interface is standard Ethernet. This feature allows the use of existing upper stack protocols and applications, which is required for the use of Ethernet for industrial automation networks in general [13–17].

Figure 3 shows a conceptual view of the structure of a DANH implemented in hardware. The two HSR ports A and B and the device port C are connected by the LRE

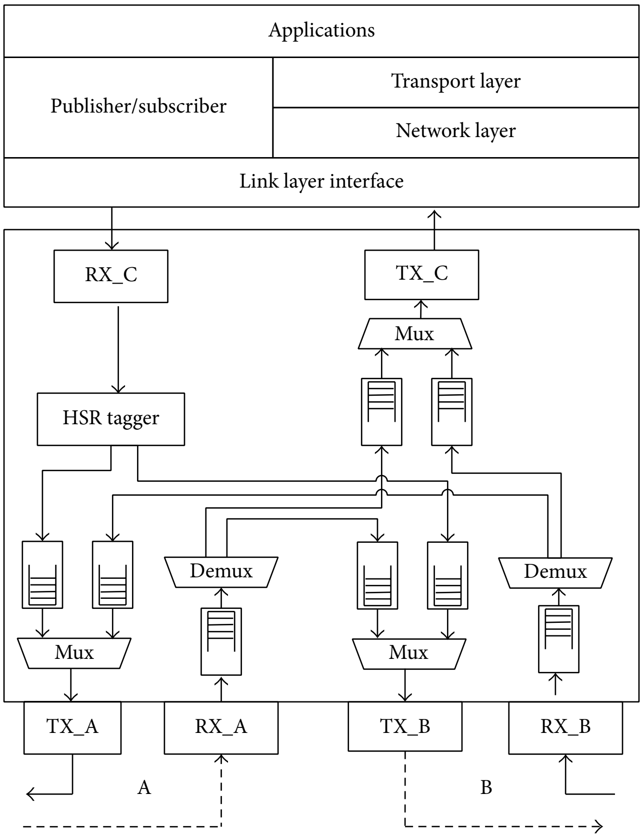


Figure 3: HSR structure of a DANH.

(link redundancy entity), which includes a switching matrix allowing for forwarding frames from one port to the other. The switching matrix allows cut-through bridging, of which the delay caused by the switch just depends on the size of the data. The LRE presents to the higher layers the same interface as a standard Ethernet transceiver would do.

The input circuit checks if this node is the destination of the frame and possibly does VLAN and multicast filtering to offload the processor. The Duplicate Discard is implemented in the output queues.

A simple HSR network consists of doubly attached bridging nodes, each having two ring ports, interconnected by full-duplex links, as shown in the example of Figure 4.

A source DANH sends a frame passed from its upper layers (“C” frame), prefixes it by an HSR tag to identify frame duplicates, and sends the frame over each port (“A” frame and “B” frame).

A destination DANH receives, in the fault-free state, two identical frames from each port within a certain interval, removes the HSR tag of the first frame before passing it to its upper layers (“D” frame), and discards any duplicate.

The nodes support the IEEE 802.1 bridge functionality and forward frames from one port to the other, except if they already sent the same frame in that same direction.

In particular, the node will not forward a frame that it injected into the ring.

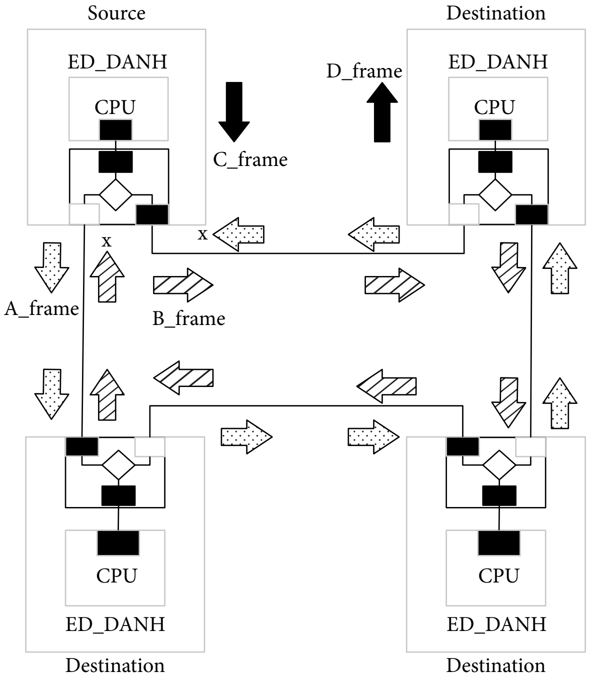


Figure 4: HSR example of ring configuration for multicast traffic.

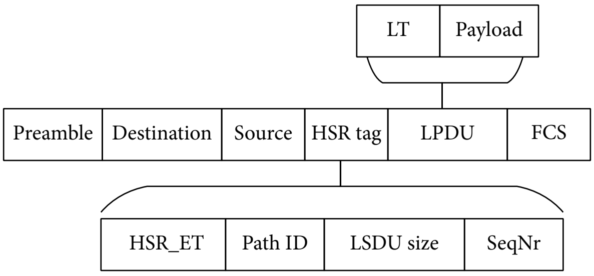


Figure 5: HSR frame format.

A destination node of a unicast frame does not forward a frame for which it is the only destination. Frames circulating in the ring carry the HSR tag inserted by the source, which contains a sequence number. The doublet (source MAC address and sequence number) uniquely identifies copies of the same frame [18–21].

3.3. Duplicates and Circulating Frames. The IEC62439-3 leaves the designer free to choose a method to manage the duplicates and circulating frame and establishes its requirements [10]:

(i) Never reject a legitimate frame, while occasional acceptance of a duplicate can be tolerated.

(ii) Be capable of eliminating more than one duplicate.

In order to control the frames that have arrived and recognize duplicates and circulating frames the HSR protocol adds extra information to the Ethernet frames. It adds an HSR tag [22]. It is shown in Figure 5.

HSR tag helps to recognize HSR frames and identify frames in order to determine whether they arrive again as duplicates or circulating frames. Their fields are as follows:

HSR EtherType (0x892F): they help to identify HSR frames.

LAN/Path ID: this indicates, in general, the sending port. It indicates the direction (clockwise or counter-clockwise).

Link service data unit (LSDU) size: it is the size in octets of LSDU + RCT/HSR tag.

Sequence number (SeqNr): this is used to control duplicates and circulating frames. Each time a node sends a frame, SeqNr is incremented by one.

Therefore, the pair formed by the origin MAC address + SeqNr defines frames and is used to detect and discard duplicates and circulating frames. Hence, the information on each received frame must be stored in a memory to be checked every time a frame arrives in order to detect whether it has arrived before.

It is worth mentioning the need for SeqNr reutilization. The SeqNr field is composed of two octets, so that after $22 * 8 = 65536$ frames, SeqNr is properly repeated. SeqNr can be repeated after t_{wap} (the time between two repetitions of the same sequence number), in the worst case, in which the shortest possible frames arrive continuously one after another.

4. Proposed Redundant Frames Discarding Method

4.1. Constraints for the Duplicate Discarding Method. The time skew between two frames of a pair depends on the relative position of the receiving node and of the sending node. Assuming a worst case in which each node in the ring is transmitting at the same time its own frame with the largest 1536 octets (maximum length supported by the EtherType defined in IEEE 802.2), each node could introduce 125 μs of delay at each 100 Mbit/s. With 50 nodes, the time skew may exceed 6 ms [10].

The Duplicate Discard is implemented in the output queue to offload the processor. The constraints of method are shown in Figure 6.

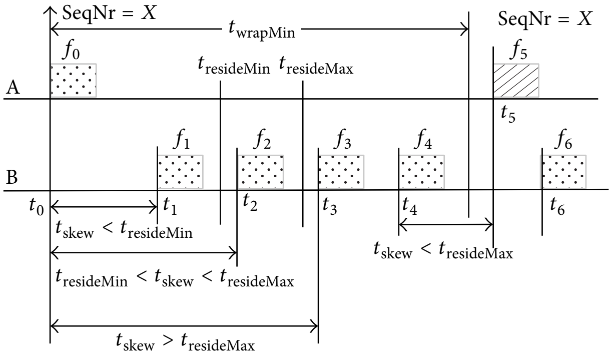

SeqNr = X
t_{wrapMin}
SeqNr = X
f_0
$t_{\text{resideMin}}$ $t_{\text{resideMax}}$
f_5
A
t_5
f_1
f_2
f_3
f_4
f_6
B
t_0
t_1
t_2
t_3
t_4
t_6
$t_{\text{skew}} < t_{\text{resideMin}}$
$t_{\text{skew}} < t_{\text{resideMax}}$
$t_{\text{resideMin}} < t_{\text{skew}} < t_{\text{resideMax}}$
$t_{\text{skew}} > t_{\text{resideMax}}$

FIGURE 6: Cases of the duplicate frame arriving at different time.

The time t_{warpMin} is the minimum possible time between two repetitions of the same sequence number by legitimate frames after 65536 increments (as dictated by the application and network constraints).

In a 100 Mbit/s network, t_{warpMin} is about 400 ms at the theoretical maximum frame repetition rate from the same source.

The time t_{skew} is the time difference between the arrivals of two copies of the same frame.

The time $t_{\text{resideMax}}$ is the maximum time an entry remains in the duplicate table; it is smaller than EntryForgetTime. During this time, the receivers will age out the entry from the duplicate tables. The time $t_{\text{resideMin}}$ is the minimum time an entry remains in the duplicate table.

The frame f_0 arrives at t_0; if the duplicate of frame f_0 arrives at t_1, ED_DANH can find the SeqNr in table, with duplicate always detected.

If the duplicate of frame f_0 arrives at t_2, SeqNr possibly in table, sometimes not detected, is treated as new.

If the duplicate of frame f_0 arrives at t_3, SeqNr is out of table, and the duplicate frame is treated as a new frame.

If the duplicate of frame f_0 arrives at t_4 and the next alias frame f_5 with the same SeqNr and new contents arrives at t_5, SeqNr is still in table, and the frame f_5 can be detected as a duplicate frame; it results that a new frame is considered to be a duplicate frame. This must be avoided because it rejects legitimate frames.

If the duplicate of frame f_0 arrives at t_6, the next alias frame f_5 with the same SeqNr and new contents arrives at t_5; the frame f_6 is considered as a duplicate frame of f_5, and the SeqNr is still in table, so the duplicate frame is detected on wrong alias.

According to the analysis above, the following can be calculated:

Rules for reliable discard: $t_{\text{skew}} < t_{\text{resideMin}}$.

Rules for safe accept: $t_{\text{resideMax}} < t_{\text{aliasRepMin}}$.

$t_{\text{aliasRepMin}}$ is the minimum possible time between the duplicated frame and the alias frame, with the same SeqNr but the new content.

A node that reboots avoids generating aliases by not sending frames during a time longer than EntryForgetTime on start-up.

The worst delay of process data application which demands the highest real-time performance is 10 ms. The node concerned in the ECN has a maximum processing time of 1.5 ms. Thus, the remaining time to cross the network will be given by

$$T_{\text{maxdelay}} = 10 - 2 * 1.5 = 7 \text{ ms}. \quad (1)$$

The delay added by each HSR node T can be broken down into three parts: t_1 for receiving delay which is the time duration from the frame received from the bus to the valid data provided to the microprocessor, t_2 for switching delay which is the frame forward procedure delay caused by the switch, and t_3 for waiting delay which is the frame delay in the queue. Consider

$$T = t_1 + t_2 + t_3. \quad (2)$$

t_1 depends on the operation mode used by HSR node, Store-and-Forward, or cut-through mode.

t_2 is the time to decide whether to forward/receive/discard a frame and it depends fundamentally on the time needed to check whether the frame has arrived before or not.

t_3 is the time a frame has to wait to be sent because another frame is being sent.

t_1 can be reduced by using cut-through mode. t_3 can be reduced by synchronized traffic generation on nodes and/or by controlling frame size. t_2 is directly related to the way of checking whether a frame is new or a duplicate or circulating frame; it can be reduced by the fast redundant frames discarding method, which is mainly discussed in this paper.

The main challenge is that HSR node has to carry out the search when receiving a frame, which adds a delay. It limits the number of nodes allowed in the ring. The maximum tolerated delay time divided by the delay time of each node is the number of nodes allowed in ECN with HSR. The maximum number of nodes N is given by

$$N = 1 + \frac{T_{\text{maxdelay}}}{(t_1 + t_2 + t_3 + t_L)}. \quad (3)$$

t_L is the delay in each link.

4.2. Proposed Dynamic Linear Hashing Method. There are several kinds of methods to look up the redundant frame in the table. In the ECN with HSR, it requires high performance of real-time, such as binary tree, red-black tree, and B tree, all because the query speed is not enough or too complex to implement in embedded real-time system which is not suitable for using. Fixed number slot static hash table which needs growing storage is also not suitable for the embedded real-time system. It is also unable to meet the needs. So this paper introduces a dynamic hash table.

Dynamic hash table is usually double the number of slot growth after the conflict, while hashing function has changed after growth, so elements of the old slot should be reposition; this realization method is the principles of "hash_map" function in the standard library. However, if each slot adjustment should relocate elements, it results in low efficiency, and data insertion time is not uniform. If reposition occurs after each conflict, it cannot tolerate such a large delay for real-time systems. Therefore, this paper proposes a new dynamic scheduling algorithm, of which the method is described as follows.

First the concept bucket is introduced; it means creating conflict table for each hash value in hash tables; few conflict records are stored in the form of table in the bucket. First, calculate the hash value according to the hash function with key value, and then access the corresponding hash bucket based on the hash value, and then traverse the array pairs to get hash bucket.

Due to limited capacity of hash bucket, the hash table may not fill up. Although there are vacancies in hash table, the new hash table entries cannot be installed into the hash table due to excessive conflict. However, the benefits of this implementation are that the maximum cost look-up table can be determined, because the number of most processed conflicts is determined.

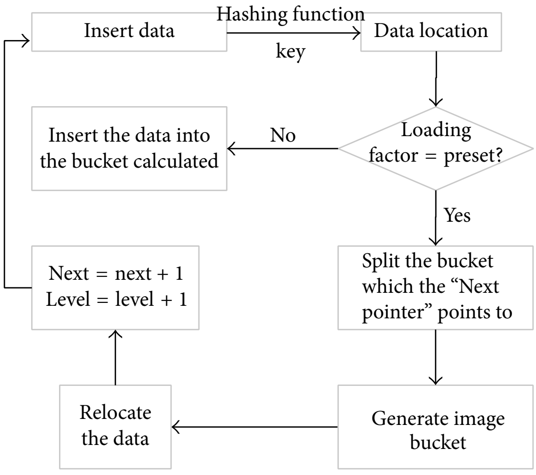


FIGURE 7: Dynamic linear hashing method.

The method implemented by chain address hash bucket capacity is unlimited. Once the hash conflict happens seriously, it will cause the list to be too long and greatly reduce the search efficiency. Therefore, this paper presents a dynamic linear hashing algorithm, based on the rotary split idea. Specifically, linear hash is appropriate to adjust the number of hash buckets with insert and delete data; compared with scalable hash, linear hash bucket does not need to store data pointers in special directory entries and can handle more natural data bucket full condition which allows a more flexible choice of bucket split time, so that it is more complicated to implement.

Rotary split evolutionary hashing algorithm is illustrated in Figure 7:

(1) Each bucket takes turns to split; when the current round split is completed, the next round split, and then split will start from first bucket. The "level" indicates the current number of rounds; its value starts from zero. Initial number of hash table buckets is assumed to be N, the value of $\text{Log}_2 N$ indicates the required minimum number of bits (d_0) for presenting the number N, and $d_0 = \text{Log}_2 N$.

(2) The "level" represents the current round number; the initial number of buckets in each round is $N * 2^{\text{level}}$. Bucket will be in ascending order with the bucket number and split one bucket at a time; it uses the "Next pointer" to point the next bucket split.

(3) The conditions of each bucket split are flexible to choose; for example, it can set a bucket loading factor r ($r < 1$) and split when number of records in the bucket reaches this value. It also can choose to split when the bucket is fulfilled.

(4) When inserting a new entry, calculate its hash value by a hash function first, and then find the corresponding position in the bucket; if the bucket is full, the overflow page increases. Because bucket split is triggered, the bucket split to

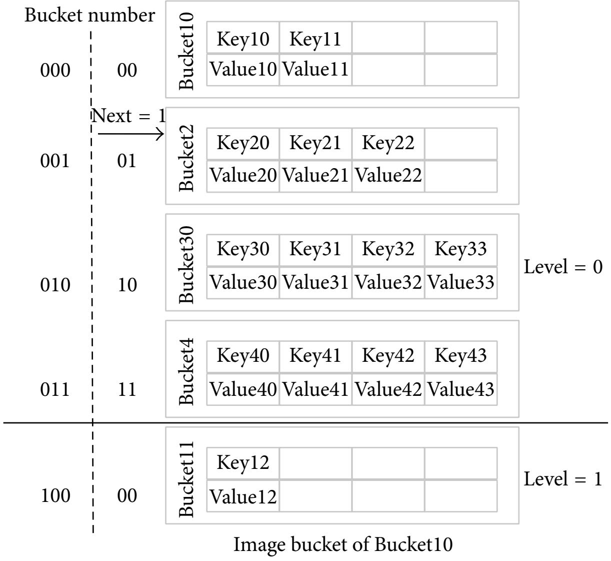


FIGURE 8: An example of the proposed method.

generate image bucket on the next position = 0. Image bucket number = N + next, and all elements of the original bucket are reallocated, and the "Next pointer" moves toward the next bucket.

(5) Bucket split is performed sequentially, and the subsequent image bucket produced is located after the last bucket produced.

An example is given to illustrate the dynamic linear hashing method. It is shown in Figure 8. Each slot is a linked list of pairs (key, value); the top half of each element contains the key, and the bottom contains the associated value. The initial number of buckets is 4, and $d_0 = 2$, and level = 0. When inserting new frame, its slot is in bucket 30 after hashing, but it is full; it triggers the split to generate the image bucket of Bucket10 and relocate the element in Bucket10 to Bucket10 and Bucket11; after relocating then insert the new frame (key12, value12) into the image bucket Bucket11. Then level = 1, and the pointer "Next" moves from Bucket10 toward Bucket2. This is the process of inserting one frame. When inserting more frames, repeat the step before and use the rotary split idea.

In order to verify the validity and performance of the proposed dynamic linear hashing method, other methods such as binary tree and static hashing are used to compare. The time required for inserting frame using different duplicated frame discarding algorithms is shown in Figure 9, and the time required for finding frame using different duplicated frame discarding algorithms is shown in Figure 10.

As it can be seen from the table, the dynamic linear hash algorithm proposed has the highest efficiency.

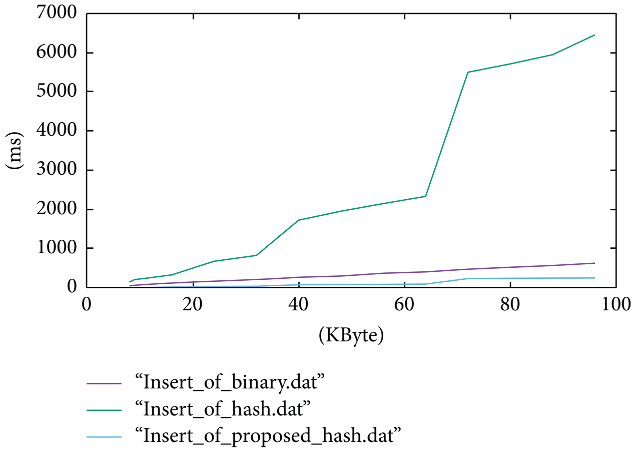


FIGURE 9: Performance of different method of inserting frame.

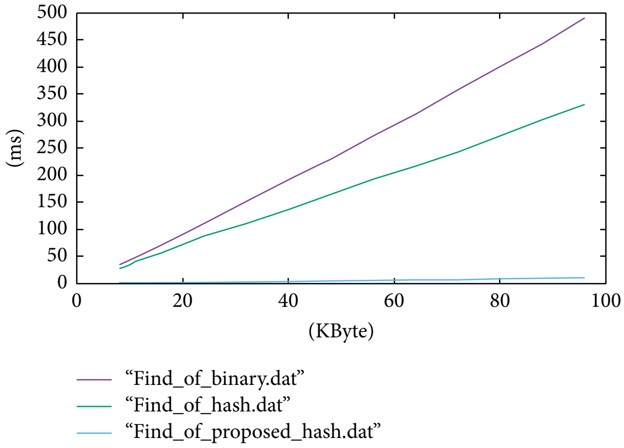


FIGURE 10: Performance of different method of finding frame.

5. The Simulation of Proposed Solutions

5.1. The Setup of the Simulation Environment. In order to verify the proposed method, the paper builds a simulation model of ECN with HSR in OPNET, which is a software package that can accurately analyze the performance and behavior of complex networks; standard or user-specific probes can be plugged into the network model anywhere to collect data and statistics.

The model of ECN with HSR is composed of the train network model, vehicle network model, and End Device node model. The train network model is illustrated in Figure 11. It is a model of train communication network based on Ethernet, TC1, M1, M2, T3, TC2, M3, M4, and T4 for the model of each vehicle; TC1, M1, M2, and T3 compose Consist 1 network model; TC2, M3, M4, and T4 compose Consist 2 network model. The two consists are connected by ETBN1 and ETBN2 switch. "Profile Config" and "Application Config" are used for the configuration of data flows between ED.

The network model of different vehicle is similar. The difference is the type of End Device which is contained in it. The ED contained in different vehicle is shown in Table 2. The network model of TC1 is illustrated in Figure 12.

The node model of each ED is similar with each other. The node model of BCU is illustrated in Figure 13.

In Figure 13, "hub_tx_0_0" and "hub_tx_0_1" are the point-to-point transmitters and "hub_rx_0_0" and "hub_rx_0_1" are

Table 2: ED in different vehicles included.

TC1/TC2	SIV, BCU, HMI, VCU1/VCU8, media, HVAC, DOORR, DOORL, and PIDS
M1	VVVF, BCU, EVR, VCU2, media, HVAC, DOORR, and DOORL
M2, M3, M4, T3, and T4	VVVF, BCU, VCU3/VCU4/VCU5/VCU6/VCU7, media, HVAC, DOORR, and DOORL

SIV: static inverter, BCU: brake control unit, HMI: human machine interface, EVR: electronic video recoder, VCU: vehicle control bus, HVAC: heating, ventilating, and air conditioning, DOORR: right side door, DOORL: left side door, PIDS: passenger information display system, VVVF: variable voltage variable frequency, EVR: Electronic Video Recoder and ERM: event record module.

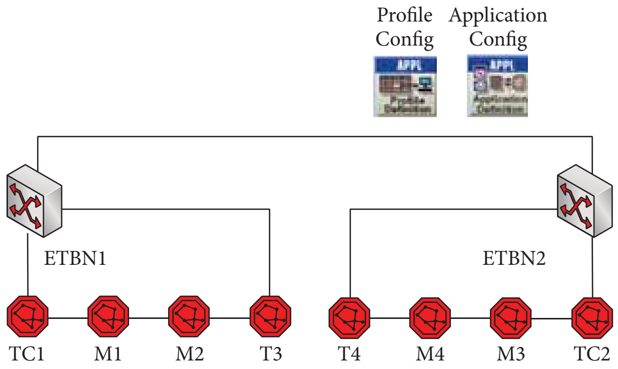


Figure 11: Network model of ETB.

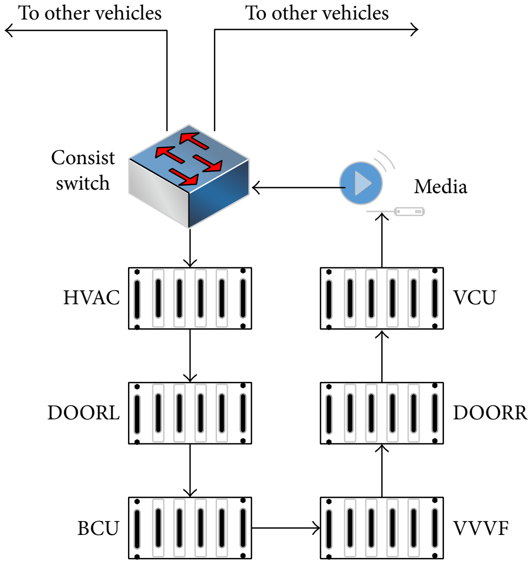


Figure 12: Node model of TC1.

the point-to-point receivers. “LRE” is the node model of the link redundancy entity for HSR, while another module is the simulation of the different protocol of Ethernet.

The simulation establishes a variety of scenarios based on different link models and different traffic configuration:

(a) Establish two scenarios of 10 Mbps and 100 Mbps link model: train traffic is configured in accordance with the actual Ethernet traffic for a type China EMU and improved rationalization provisions (Ethernet minimum frame length is 64 Bytes), considering only the transmission of process data, without considering other data types.

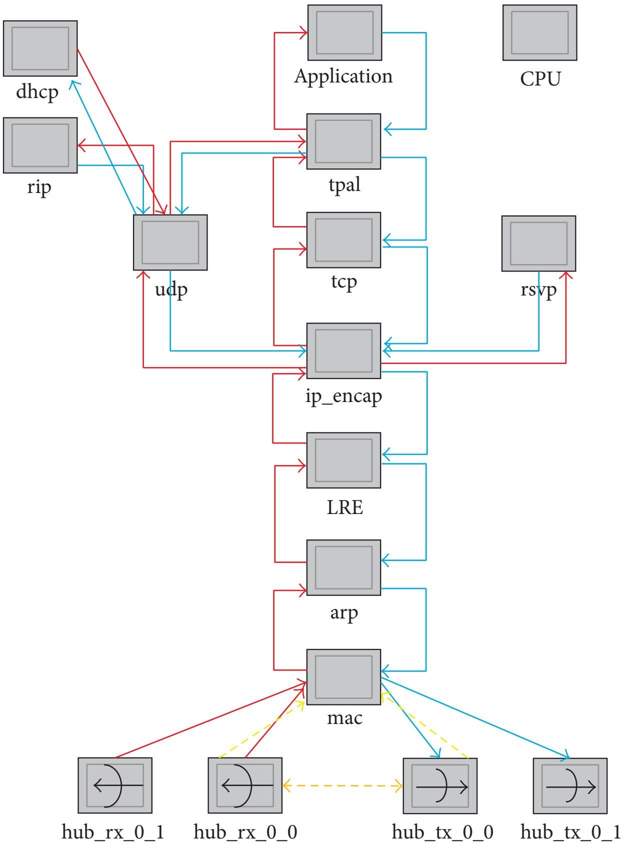


Figure 13: Node model of BCU.

(b) Establish the same scene with step (a) and consider the process data and cycle data, the cycle of BCU data communication is reduced from step (a) to simulate network performance under heavy load conditions, without considering other data types.

(c) Establish the same scene with step (a), transmission of process data, messages, data, and multimedia streaming data, compared with (b), indicating an increase of multimedia data.

5.2. Simulation Configuration. According to the cycle time of data in a type China EMU train communication network and data flow, configuration of data size, cycle time, and traffic flow of Ethernet-based train communications network is as follows.

Two simulation scenarios are covered in this section; simulation data configuration parameters are shown in Table 3, and data traffic flows are shown in Table 4.

Table 3: Simulation data configuration parameter.

Number	Period	Data size in MVB	Data size in ECN
①	128 ms	8 bits	64 Bytes
②	20 ms	32 bits	256 Bytes
③	32 ms	64 bits	512 Bytes
④	512 ms	16 bits	128 Bytes
⑤	256 ms	32 bits	256 Bytes
⑥	256 ms	4 bits	64 Bytes
⑦	256 ms	16 bits	128 Bytes
⑧	256 ms	8 bits	64 Bytes
⑨	32 ms	16 bits	128 Bytes
⑩	44 ms	2 bits	64 Bytes
⑪	10 frames/sec		128∗120 pixels

Table 4: Simulation data flow.

Number	Date flow	Traffic
1	BCU → VCU1	②③④
2	VCU1 → BCU	①
3	VVVF → VCU1	②③
4	VCU1 → VVVF	②
5	SIV → VCU1	⑤
6	VCU1 → SIV	⑤
7	PIDS → VCU1	⑤
8	VCU1 → PIDS	⑤
9	HVAC → VCU1	⑤
10	VCU1 → HVAC	⑧
11	VCU1 → VCU2, VCU3	②⑤⑨
12	HMI → VCU1	④
13	VCU1 → HMI	⑤
14	HMI → ERM	⑩
15	ERM → HMI	⑦
16	Media	⑪

"Period" in Table 3 is the cycle time of data. "Data size in MVB" is the data size of a certain type of traffic in MVB, which is a traditional train control network in TCMS. "Data size in ECN" is the data size of a certain type of traffic in ECN, which is the future train control network in TCMS. "Data size in ECN" is obtained in accordance with the "Data size in MVB."

5.3. Simulation Results and Discussion. According to the current train traffic, in case of 100 Mbps model, the received delays of BCU in TC1, M1, and M2 data separately are 0.080 ms, 0.198 ms, and 0.105 ms, as shown in Figure 14, and they meet the demand of IEC61374-3-4 which is 10 ms.

The received delay of VCU1 in TC1 in the case only concluding the message data is 0.130 ms, is 0.141 ms in the case only concluding the message data and process data, and is 0.6 in the case of adding multimedia data, as shown in Figure 15. It can meet the demand in IEC61375-3-4.

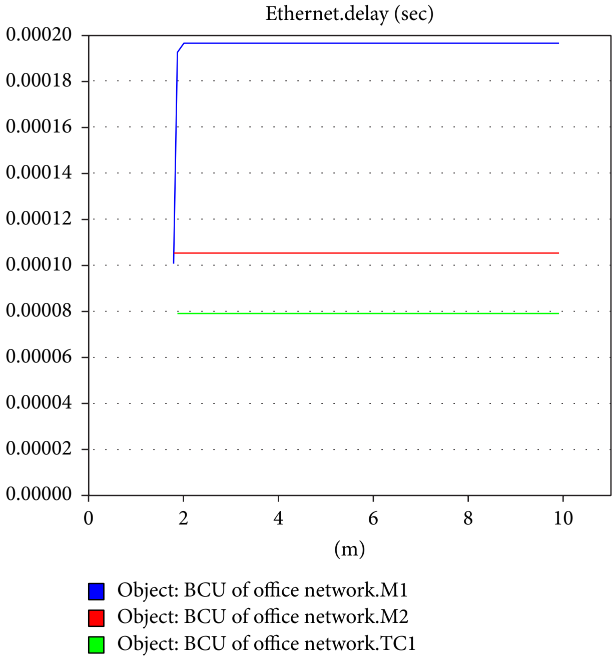


Figure 14: The transfer delay of BCU in TC1, M1, and M2.

6. Conclusion

As a safety-critical application train control network requires strong resilience, HSR protocol allows zero recovery time to recover network functionality when a failure occurs but also brings some network latency. This paper proposed a dynamic linear hashing algorithm to reduce the data exchange time and build a simulation model with OPNET to verify the validity of proposed method. The results show that the proposed method can ensure high availability and real-time performance of train Ethernet consist network.

Competing Interests

The authors declare that they have no competing interests.

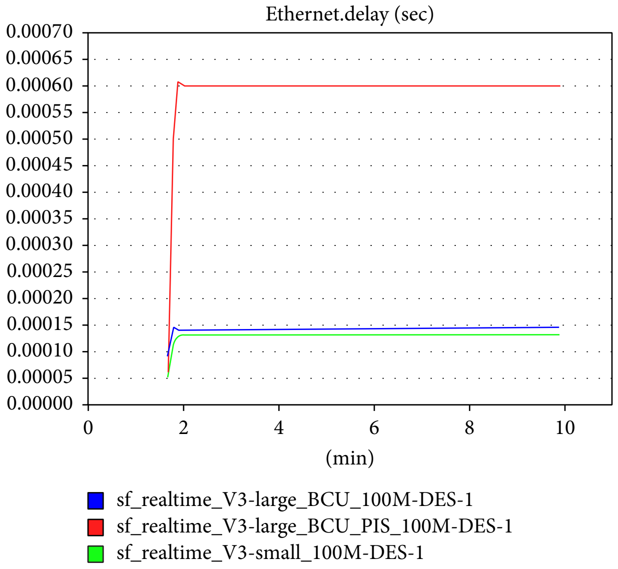


Figure 15: Delay of different traffic.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities (2014JBM113).

References

[1] M. Schmitz, *Ethernet as a Train Bus*, MEM Mikro Elektronik GmbH, Nuremberg, Germany, 2010.

[2] Y. Tian, H.-H. Dong, L.-M. Jia, Y. Qin, and S.-Y. Li, "The bilevel design problem for communication networks on trains: model, algorithm, and verification," *Mathematical Problems in Engineering*, vol. 2014, Article ID 840619, 9 pages, 2014.

[3] C. C. Bantin and J. Siu, "Designing a secure data communications system for automatic train control," *Proceedings of the Institution of Mechanical Engineers Part F: Journal of Rail and Rapid Transit*, vol. 225, no. 4, pp. 395–402, 2011.

[4] K. Kamata and H. Takahashi, "T-Ethernet: the next international standard candidate for train communication network," in *Proceedings of the World Congress on Railway Research*, March 2008, http://www.uic.org/cdrom/2008/11_wcr-r2008/pdf/O.3.3.2.4.pdf.

[5] G. Fadin, "ICT Standardisation on board of trains," in *Proceedings of the CENELEC-ACRI Workshop Information and Communication Technology on Trains*, October 2011.

[6] R. Grübmeyer and S. Rupp, "Ethernet-based fieldbuses for industrial networks: the basics—part 2," *Industrial Ethernet Book*, no. 67–42, 2011.

[7] X. Li, H. Wu, and Y. K. Yang, "Consensus of second-order multiagent systems with fixed topology and time-delay," *Mathematical Problems in Engineering*, vol. 2015, Article ID 138276, 6 pages, 2015.

[8] R. Xie, W. Fan, and Q. Chen, "Distributed H_∞ consensus control of networked control systems with time-triggered protocol," *Mathematical Problems in Engineering*, vol. 2015, Article ID 963282, 13 pages, 2015.

[9] J. Xiaozhuo, *High availability seamless ring protocol implementation in FPGA [M.S. thesis]*, ETH Zürich Computer Engineering and Networks Laboratory, Zürich, Switzerland, 2009.

[10] J. A. Araujo, J. Lazaro, A. Astarloa, A. Zuloaga, and J. I. Garate, "PRP and HSR for high availability networks in power utility automation: a method for redundant frames discarding," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2325–2332, 2015.

[11] M. Rostan, *Industrial Ethernet Technologies: Overview,*, ETG Industrial Ethernet Seminar Series, 2008.

[12] International Electrotechnical Commission, *Electronic Railway Equipment—Train Communication Network (TCN)—Part 3-4(S): Ethernet Consist Network (ECN). Edition 1.0*, IEC61375-3-4, 2014.

[13] H. Kirrmann, K. Weber, O. Kleineberg, and H. Weibel, "Seamless and low-cost redundancy for substation automation systems (high availability seamless redundancy, HSR)," in *Proceedings of the IEEE PES General Meeting: the Electrification of Transportation and the Grid of the Future*, pp. 1–7, San Diego, Calif, USA, July 2011.

[14] J. Kawasaki, M. Sugaya, A. Sobue et al., "Development of 100 Mbps-Ethernet-based train communication network," in *Proceeding of the 9th World Congress on Railway Research*, Lille, France, May 2011.

[15] H. Heine and O. Kleineberg, "The high-availability seamless redundancy protocol (HSR): robust fault-tolerant networking and loop prevention through duplicate discard," in *Proceedings of the 9th IEEE International Workshop on Factory Communication Systems (WFCS '12)*, pp. 213–222, Lemgo, Germany, May 2012.

[16] IEC, "Industrial communication networks: high availability automation networks, part 3: parallel redundancy protocol (PRP) and high-availability seamless redundancy (HSR)," IEC 62439-3, International Electrotechnical Commission, 2012, 2nd edition.

[17] H. Kirrmann, "Parallel Redundancy Protocol an IEC standard for a seamless redundancy method applicable to hardreal time Industrial Ethernet," 2012, http://lamspeople.epfl.ch/kirrmann/publications.htm.

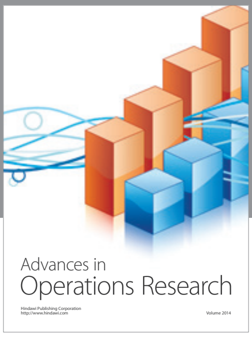
[18] H. Kirrmann, "Highly available automation networks standard redundancy methods; rationales behind the IEC 62439 standard suite," March 2013, http://lamspeople.epfl.ch/kirrmann/publications.htm.

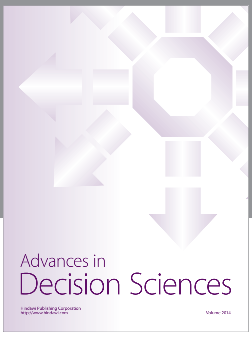
[19] H. Kirrmann, "HSR high availability seamless redundancy," 2012, http://lamspeople.epfl.ch/kirrmann/publications.htm.

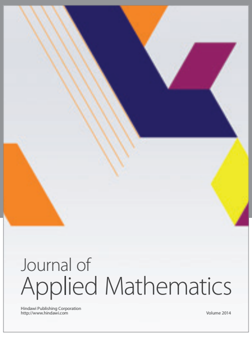
[20] H. Kirrmann, K. Weber, O. Kleineberg, and H. Weibel, "Seamless and low-cost redundancy for substation automation systems (high availability seamless redundancy, HSR)," in *Proceedings of the IEEE Power and Energy Society General Meeting (PES '11)*, pp. 1–7, San Diego, Calif, USA, July 2011.

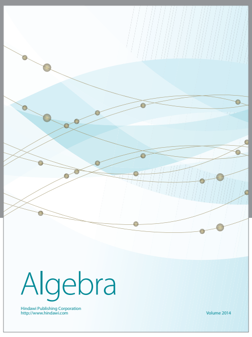
[21] S. A. Nsaif and J. M. Rhee, "Improvement of high-availability seamless redundancy (HSR) traffic performance for smart grid communications," *Journal of Communications and Networks*, vol. 14, no. 6, pp. 653–661, 2012.

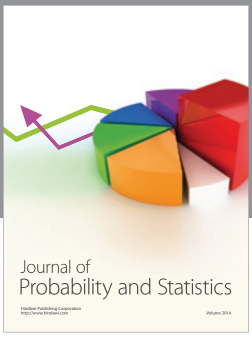
[22] H. Kirrmann, S. Kunsman, and P. Rietmann, "ABB Network Redundancy Using IEC 62439," Protection, Automation & Control World, Autum 2008.


Advances in
Operations Research

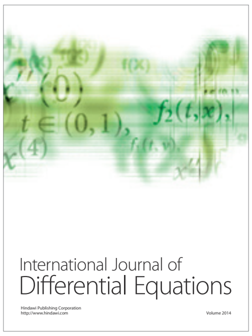

Advances in
Decision Sciences


Journal of
Applied Mathematics

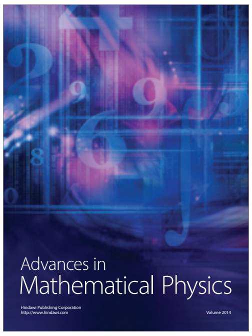

Algebra

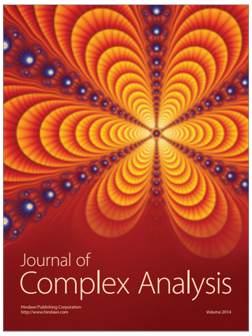

Journal of
Probability and Statistics

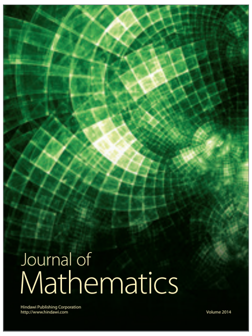
The Scientific
World Journal

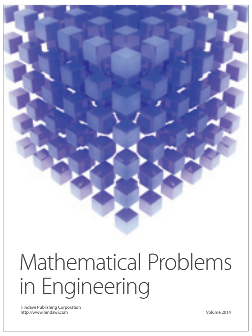

International Journal of
Differential Equations

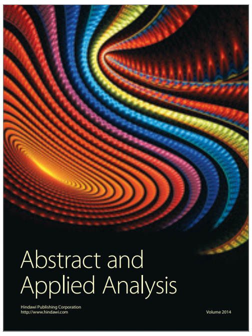
International Journal of
Combinatorics

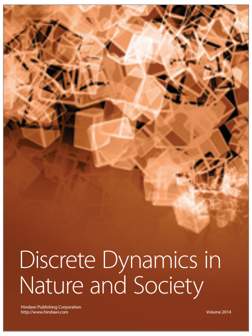

Advances in
Mathematical Physics

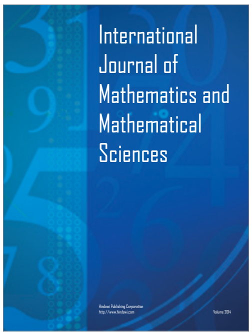

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

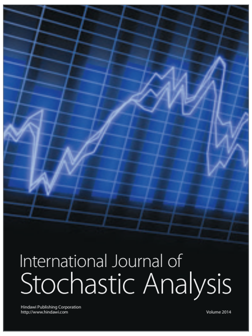

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization