UDC 621.396

IMPULSE ULTRA-WIDEBAND SIGNAL RELAYING IN AD HOC RADIO NETWORKS

Sergiy G. Bunin, Iurii Iu. Voitenko, Konstantin A. Plotnik

National Technical University of Ukraine “KPI”, Kyiv, Ukraine

A method of impulse ultra-wideband signals relaying in ad hoc radio networks is described. As the relaying signals a group of chipsets is used to represent various minimal information units. A system of markers is introduced to unambiguous determine the relaying routes. The chipset representation of transmitted signals reduces the delays coursed by multistep relaying and increases the data transfer rate.

Introduction

A possibility of using impulse ultra-wideband (IR-UWB) signals in ad hoc radio networks is being studied more often [1, 2]. One advantage of these signals is the use of a simple protocol for the organization of ALOHA-like multiple access protocol, which does not require prior information about the channel occupation status.

As each bit of information is encoded by the sequence of ultra-short impulses and time positioning of these impulses generates orthogonal code (pulse position modulation), it is possible to simultaneously transmit data from multiple sources without channel occupation. Using IR-UWB signals for traffic relaying in ad hoc networks allows a delay of one pulse duration, since relayed signal is orthogonal with respect to the received one if the receiver is synchronized with the received pulse sequence.

Ad hoc networks do not have a fixed topology. In such networks, nodes can free move in any direction, thereby creating a dynamically changing topology of the network [3]. Routing information and its quality is constantly changing over time. As a result, route delay variation is constantly changed that is a serious problem to organize the specified quality of service.

In conventional wired and wireless networks, to relay the signal a “store-and-forward” method is used [4]. Intermediate node which has received the packet, stores it in a buffer, checks the integrity of it by using the checksum recalculation, analyzes source-destination pair, identifies the node that needs to receive the packet and sends it further to the network. This method adds a delay in relaying at each intermediate node, equal to the packet duration of T. Then, the packet relaying in a network with N intermediate nodes will be delayed on NT (excluding the time of signal propagation and its processing). Under limited data rate, to reduce the delays associated with the relaying, less information should be transmitted.

Retransmission of an impulse itself in the network does not provide any advantage as a single impulse bears no information. The bit relaying in the form of chip (Fig. 1) is feasible, since each bit is represented by sequence of impulses forming orthogonal code having a host address.

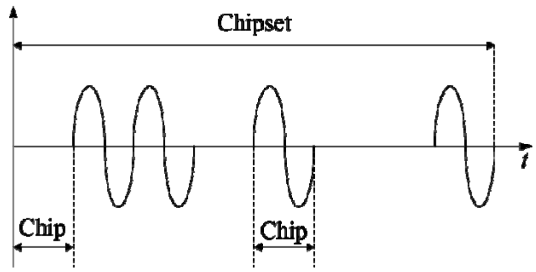

Fig. 1. Chip and chipset examples.

To relay the chipset in the network, one must know not only the sender but also the final recipient of the chipset. This is possible to achieve by introducing the markers. Each chipset adds a token that uniquely characterizes the route in the network. Each intermediate node involved in relaying this chipset has the marker which can identify where to relay the received chipset.

Problem identification

A method of relaying IR-UWB signals, namely chipset relaying, was presented in [5]. Along with the main advantage consisting in capability to minimize relaying delay on each node, this method has several drawbacks:

1). Each node should maintain a complete table of markers in the network that increases the number of control information transmitted. The node assigned a new token for a particular route should know what markers are already being used in the network to guarantee the uniqueness of a new marker. Otherwise, it should be a central node providing such unique markers.

2). Proportional increase of required number of unique markers with grow the network size demands an increase in the number of orthogonal codes for the markers. This complicates the correct detection of the markers and consequently chipsets.

3). If one route is used for multiple transmissions, it is impossible to separate chipsets at the receiver for each communication scenario.

To solve the problems in chipset relaying stated above, we propose another method named as single-code chipset relaying. The basic idea of this method is that the original sender transmits data to the destination host in its code and so forms the all intermediate nodes.

Single-code chipset data relaying

Consider the transmitting IR-UWB signals in a radio network and the proposed method of single-code chipset relaying. Prior to the route discovery and maintenance of the route to each network, the node is assigned by unique mutually (or quasi) orthogonal code which at the same time is its address.

Hence a system introduces markers and marker table at each node, where each table will contain unique tokens within a given node and its neighbors as well as within the same code. This means that each node can choose their own new marker for a new route on the basis of its markers table. Fig. 2 presents the network topology which shows the ability to establish end route between nodes.

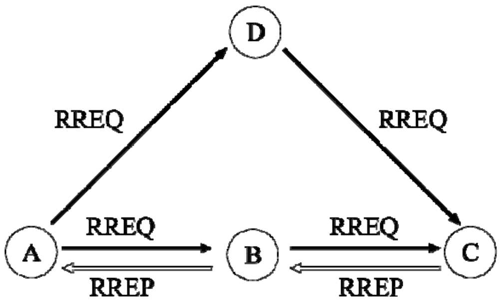

Fig. 2. Route discovery and maintenance between nodes A and C.

According to the topology shown in Fig. 2, the node C may have the following table of markers.

As shown in Table 1, the node C has markers for the node with its neighbors (B, D). Using terms of databases, in this table, field markers Mark, Bit and Code are unique composite keys. The abbreviations Tr and Rec specify the transmitter and the receiver, respectively. Accordingly, the number of unique markers is significantly reduced as the marker must be unique only when matching the sending or receiving code.

Table 1. Marker table for node C.

Mark	Bit	Code	Src IP	Dest IP	#Tr	#Rec
1	0	C	IP D	IP C		1
1	1	C	IP D	IP C		2
2	0	C	IP B	IP C		7
2	1	C	IP B	IP C		8
1	0	B	IP C	IP B	1	
1	1	B	IP C	IP B	2	
2	0	B	IP D	IP C		3
2	1	B	IP D	IP C		4
1	0	D	IP B	IP C		5
1	1	D	IP B	IP C		6

There are many reactive routing protocols for ad hoc networks standardized in RFC documents [6]. For further description of the route discovery and maintenance, we select the routing protocol AODV. Briefly, this protocol can be described as follows. The sender node sends a broadcast request (RREQ) to its neighbors, which in their turn check routing table to find the route with the specified host as the final destination.

If the neighbor has not found a route to a given destination, it rebroadcasts a RREQ packet. This process is repeated as long as the RREQ reaches the destination host or an intermediate node containing in its routing table the route to the final destination.

Route discovery and maintenance

Before going into details of the proposed method, consider some changes required in AODV for implementation of the “chipset retransmission”.

1. Prior to the beginning of the route discovery process, each node in the ad hoc network must allocate 2 codes as a set of chips (impulses). Each code is used to encode one bit (0 or 1). These codes must be orthogonal or quasi-orthogonal to all nodes in the same subnet. According to the AODV protocol, each node has a neighbor table indicated their IP addresses. To implement chipset retransmission in ad hoc network, columns Code1 and Code0 must be added to this table. Codes for bits 0 and 1 must be written per neighbor. Table 2 shows an example of such table.

Table 2. List of neighboring nodes.

IP	Code1	Code0
192.168.1.1	0110010... ...1011001110	0111010... ...1011010010
192.168.1.2	1101000... ...1101101011	1101011... ...1001100111

2. Each route between the sender and the receiver must be labeled by a marker. The marker is a unique set of chips which form the code sequence generated the

node on the physical level. Let the code consists of 100 time positions (one time position is equal to the pulse duration) and the marker consists of 10 time positions. Then, the last 10 time intervals of code will be replaced by the marker intervals. The introduction of markers specifies the utilization of the marker table for each node. Fig. 3 shows an example of a marker in the main code.

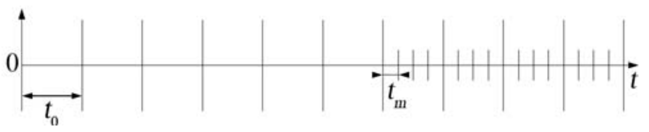

Fig. 3. Example of the marker in the main code, where t_0 denotes the length of the main impulse; t_m is the length of the chipset impulse.

Therefore each node must maintain a marker table which format is presented in Table 3, where Mark is a unique token (for example, the decimal digit 3 in binary description is 0000000011 if the length of the marker is equal 10); Bit defines bit 1 or 0 (for the bit 0 or 1 the node uses different codes); Code presents the code specifying the final destination host; Src IP is the IP address of the sender; Dest IP denotes the IP address of the final destination; #Tr defines “transmitter number”; # Rec denotes “receiver number”.

Table 3. List of markers.

Mark	Bit	Code	Src IP	Dest IP	#Tr	#Rec
3	0	C	192.168.1.0	192.168.1.3	1	1
3	1	B	192.168.1.0	192.168.1.3	2	2

As shown in Table 4, each node has a table of dynamic codes, in which the transmitter can generate bits. Thus, “transmitter number” is a serial number code in the code table node. It implies that the node has some passive filters tuned to its own code. “Receiver number” indicates the number of passive filter configured to receive a code with a marker Mark.

Table 4. Dynamic codes table for a given node.

#	Code
1	0110010... ...1011111110
2	0110010... ...1010101010
3	0110010... ...1101010011

Consider the example of a route discovery and maintenance from node A to node C as shown in Fig. 2. The completion of this process is based on assumption of the route symmetry. This indicates the equivalence of transmit information ways from node A to C and conversely from C to A.

1. Node A broadcasts RREQ to its neighbors, indicating the final destination, the maximum number of hops, the sender address and other supplementary information.

2. Node C receives several RREQ packets from node A, selects the best (the minimum number of hops) and prepares a response RREP to node A forming, for instance, the route C_B_A.

Before sending the RREP packet from node C to node B: node C creates a unique marker related to its own marker table and code of node C; node C sets one of the receiving passive filters for the route A_C to the codes of bit “1” and “0”, respectively; node C adds to its dynamic marker table a code of bit “1” of node A and a marker with, for instance, a sequence number 1. The same operation should be done for a bit “0”. Following this steps, the table of markers looks as shown in Table 5.

Table 5. List of markers for node C.

Mark	Bit	Code	Src IP	Dest IP	#Tr	#Rec
1	0	C	IP B	IP C	-	1
1	1	C	IP B	IP C	-	2
1	0	A	IP C	IP B	1	-
1	1	A	IP C	IP B	2	-

Then, the node C adds to the routing table a route C_A. Here, the hop count equals two, the next IP of node B corresponds to dest IP of node A. Then, the node C unicast transmits a RREP to the node B. This message contains markers associated with the node B.

3. Node B receives the RREP packet and:

a) detects markers in which the final destination is a node B and forms two sets of receiving passive filter to the code of A (bits 1 and 0, respectively).

b) detects markers in which the sender (Source IP) is a node B, adds them to its dynamic code of bit “1” and “0” for node C in accordance with code table.

After accomplishment of steps a) and b), a table of markers looks as shown in Table 6.

Table 6. List of markers for node B.

Mark	Bit	Code	Src IP	Dest IP	#Tr	#Rec
1	0	C	IP B	IP C	1	-
1	1	C	IP B	IP C	2	-
1	0	A	IP C	IP B	-	1
1	1	A	IP C	IP B	-	2

Since after that the RREP packet is transmitted to the node A, the following steps are taken by node B:

c) It creates a unique marker for the code of node C which will receive the data from node A. Since the table of markers already has a marker with the code of node C, it forms a new marker signed by 2. One of the receiving passive filters tuned to the code of bit “1”, the other ones to the code of bit “0”.

d) It performs the same actions as in item a) but for node A.

After fulfillment of actions specified in items a) _ d) the table of markers looks as presented in Table 7.

Table 7. List of marker on the node B.

Mark	Bit	Code	Src IP	Dest IP	#Tr	#Rec
1	0	C	IP B	IP C	1	-
1	1	C	IP B	IP C	2	-
2	0	C	IP A	IP B	-	3
2	1	C	IP A	IP B	-	4
1	0	A	IP C	IP B	-	1
1	1	A	IP C	IP B	-	2
2	0	A	IP B	IP A	3	-
2	1	A	IP B	IP A	4	-

e) Node B adds a route A_C and a route C_A to the routing table. Routing table of the node B is then as follows. At the hop count equaled to unity, the next IP nodes A and C correspond to destination IP nodes A and C respectively.

4. When the node A receives a RREP:

a) It detects markers in which the destination is a node A and assigns two matched filters on bits “1” and “0” (filters with sequence number 1 and sequence number 2, respectively).

b) It detects markers in which the sender is a node A, adds to its table of dynamic codes a code of bit “1” and “0” for a node C.

After fulfillment of steps a) and b), a table of markers looks as presented in Table 8.

Table 8. List of markers for node A.

Mark	Bit	Code	Src IP	Dest IP	#Tr	#Rec
2	0	C	IP A	IP B	1	-
2	1	C	IP A	IP B	2	-
2	0	A	IP B	IP A	-	1
2	1	A	IP B	IP A	-	2

c) It adds a route A_C to the routing table, where the hop count equals two, the next IP of node B corresponds to destination IP of node C.

This ends the route discovery and maintenance phase. As a result, each node obtains unique marker for routes A_C and C_A.

Data transmission by single-code chipset relaying

Consider the process of data transmission from node A to node C over the route discovered above. The transmission will be carried out in the code of C. For instance, in order to transmit a bit “1” from node A to node C, the node A finds in the marker table the row with fields Code C, Bit 1, Source IP A. Then it determines the number of “transmitter” (in this case 1). Using this number from the dynamic codes table, it selects code for bit “1” of node C and respective marker.

Node B at 2nd and 4th receiving passive filters detects bit “1” with code of node C and then checks the marker. Since node A transmits bit with a marker 2, the 4th passive filter will detect a bit “1” using the marker table at receiving passive filter 4. Next identifies fields Bit 1, Code C, Src B and after that the “transmitter” (in this case 2).

This procedure makes it possible to identify next hop code and to relay a bit further. Node C detects the “chipset” in one or more receiving matched filters. This chipset is examined for compliance with the marker.

Conclusion

A method of impulse ultra-wideband radio signals relaying in ad hoc networks is presented. The method reduces signal time delays by introducing intermediate nodes. The presented method of single-code chipset relaying eliminates the shortcomings of a simple chipset relaying by introducing chipset markers. Uniqueness of markers is provided in the line-of-sight of the neighboring nodes. This reduces the required number of unique markers in the network and simplifies the process of creating the token since each node has complete information about the markers within its neighborhood.

Reducing the number of unique markers in the network increases the probability of correct recognition of the chipset in the intermediate nodes.

References

1. Radio resource sharing for ad hoc networking with UWB / F. Cuomo, C. Martello, A. Baiocchi, F. Capriotti // IEEE Journal on Selected Areas in Communications. — 2002. — Vol. 20, N. 9. — P. 1722—1732.

2. Negi R., Rajeswaran A. Capacity of ultra wide band wireless ad hoc networks // IEEE Transactions on Information Theory. — 2007. — Vol. 6, N. 10. — P. 3816—3824.

3. Frodigh M., Johansson P., Larsson P. Wireless ad hoc networking: The art of networking without a network // Ericsson Review. — 2000. — N. 4. — P. 248—263.

4. Russel D. The principles of computer networking // Cambridge University Press, 1989. — 514 p.

5. Sharma V., Hellstrand F. RFC 3469: Framework for multi-protocol label switching (MPLS)-based recovery // IETF. — February 2003. 40 p.

6. Perkins C., Belding-Royer E., Das S. RFC 3561: Ad hoc on-demand distance vector (AODV) routing // IETF. — July 2003. — 38 p.

Received in final form August 21, 2012