ЕЛЕКТРОНІКА, РАДІОТЕХНІКА ТА ЗАСОБИ ТЕЛЕКОМУНІКАЦІЙ

УДК 627.372

S.G. Bunin, K.O. Plotnyk, Iu.Iu. Voitenko

RETRANSMISSION METHOD OF IMPULSE ULTRAWIDEBAND SIGNALS IN AD-HOC NETWORKS

This paper describes a new method of data relay in wireless Ad-Hoc networks with impulse ultrawideband radio signals (IR-UWB). Proposed method uses concept of “chipset retransmission” rather than typical store-and-forward scheme. This implies a relay by the intermediate nodes of each particular bit of data, encoded as chipsets, and the use of special markers integrated into that bit transferred in the connection. This provides lower delays and, thus, gives unprecedented advantage in fast retransmission for the multihop environment. Some problems still exist for further examination, such as inability to distinguish bits in a stream, error correction on bit level, ensuring uniqueness for each marker and bit code. Therefore before employing this method nodes should exchange some preliminary parameters in order to exploit all the advantages of this method. These parameters are shown in this paper as well as an example of their integration into well known reactive protocols, such as AODV, are studied.

Introduction

The possibility of use Impulse Ultra Wide Band (IR-UWB) signals in Ad-Hoc networks is examined in our days more often [1, 2]. The main advantage of these signals in Ad-Hoc networks is the possibility to significantly simplify multiple access protocols. As each bit of information is presented with a set of very short pulses and position of these pulses in time domain (pulse position modulation) creates orthogonal codes then such systems can use simple multiple access protocol such as ALOHA where each terminal can transmit data without knowing channel state. The usage of IR-UWB signals in Ad-Hoc networks also makes possible to retransmit signal with time delay of one impulse duration as this will be a new orthogonal signal in a case when receiver is synchronized with initial impulse series [3].

In Ad-Hoc, and especially in mobile Ad-Hoc networks (MANET), there is no infrastructure and all nodes are free to move dynamically creating network topology [4]. Routes and their parameters in such networks are varying in time, therefore retransmission delays are critical for such networks. The reason to study retransmission delays is twofold. On the one hand the availability of certain routes can be short period of time so we need to transmit bit, byte or packet as fast as possible over this route with minimal retransmission delays. On the other hand decreasing retransmission delays will automatically increase data rate of the connection.

In classical wire and wireless networks with multihop transmissions, method “store-and-forward” is usually used for retransmission on intermediate nodes [5]. Information is sent to an intermediate node where it is stored and sent at a later time to the final destination or to another intermediate node. The intermediate node verifies the integrity of the message before forwarding it. Such method adds a delay on each intermediate node equals to packet duration T. Hence multihop transmission with N hops requires $N \cdot T$ delay (without signal propagation and processing time). It is obvious that reduced retransmission delay in wireless Ad-Hoc network with limited data rate is achievable when something shorter than a packet is relayed: byte, bit or even impulse.

However, retransmission of a single ultrawideband impulse gives any benefit as single impulse can’t provide information on source or destination node or any other information for retransmission and can be even erroneously received. As one bit of information is coded by set of impulses with pulse position modulation and such code uniquely describes address of node we can implement retransmission of bits, bytes, set of bytes and so on. For abstraction, let’s introduce the concept of chipset – a piece of data that is used for retransmission in multihop network.

We propose new method of retransmission in Ad-Hoc with IR-UWB signals based on markers. We call this method “chipset retransmission”, where chipset means bit, byte or any predefined amount of data. This implies that to each retransmitting chip a special marker (or label) that uniquely describes a route in network is added. Each intermediate node that takes part in retransmission of this chip knows this marker and knows where to retransmit next this chip. Similar approach is used in Multiprotocol Label Switching (MPLS) networks where header with route label is added to each packet and retransmission on each node is made by analyzing route label in packet header with use of label tables [6]. Difference of our approach is that we don't add any header with marker as we retransmit for example, one bit, but we insert marker directly into the chip without changing of main code (address) of source node.

Formulation of the problem

The goal of this paper is to present a new method of retransmission in Ad-Hoc networks with IR-UWB signals called "chipset retransmission" and to consider in details such algorithm in Ad-Hoc network with IR-UWB signals and reactive routing. Advantages and problems of chipset retransmission method compared to method "store-and-forward" are studied.

Route discovery and maintenance

Many reactive routing protocols for Ad-Hoc networks are proposed. Such protocols as AODV and DSR are standardized and described in RFC documents [7]. In this work we employ AODV routing for further considerations. Simplified route discovery in AODV can be described as follows: source node broadcasts route request packet (RREQ); all neighbors upon receiving of this packet check their routing tables for a route to the destination; if one has a route to the destination it unicasts back to the source a route reply packet (RREP); if no route is available it rebroadcasts RREQ until it reaches destination or a node with "fresh enough" route (here we skip all details of AODV routing).

Before going into the 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, which is a set of chips (pulses). 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, which indicates their IP addresses. In order to implement "chipset retransmission" in Ad-Hoc network columns "Code1" and "Code0" must be added to this table. There will be written codes for bits 0 and 1 per neighbor. Table 1 shows an example of such table.

***Table* 1.** List of neighbors

IP	Code1	Code0
192.168.1.1	0110...11001110	0111...1010010
192.168.1.2	1101...01101011	1101...0100111

2. Each route between the sender and the receiver must be marked with a marker. The marker is a unique set of chips (pulses) and it is put in the code sequence that generates the node on the physical level. If, for example, the code consists of 100 time positions (one time position (or slot) is equal to the pulse duration), and the marker, for example, consists of 10 time positions, then the last 10 time intervals of code will be replaced by the marker intervals. In connection with the introduction of markers, each node must have a table of markers, which has format presented on table 2.

***Table* 2.** List of markers

Marker	Bit	Source IP	Destination IP	#Transmitter	#Receiver
3	0	192.168.1.0	192.168.1.3	1	1
3	1	192.168.1.0	192.168.1.3	2	2

Note: Marker – unique marker in decimal form; Bit – defines bit 1 or 0, because transmission of 1 and 0 requires different codes; Source IP – IP address of the source; Destination IP – IP address of the destination; #Transmitter – "transmitter number". Each node has a table of dynamic codes, presented in table 3, where every code generation is based on 2 main codes presented in table 2 with different markers. Thus "transmitter number" is a row number in the table of codes (table 3); #Receiver – "receiver number". Each node has several programmable code matched filters. Each of them can be assigned to detect only specific code. Thus, "receiver number" shows number of a matched filter which detects corresponding *Marker*.

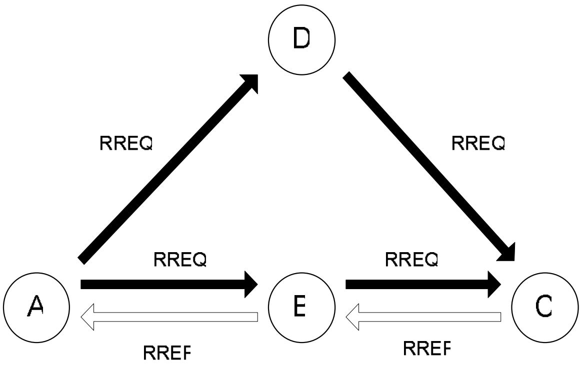
***Table* 3.** Dynamic codes table

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

Now, let us consider in details the process of a route discovery with "chipset retransmission". Suppose, we need to transfer data from node A to node C (Figure).

1. Node A sends a RREQ to its neighbors, where specifies the address of the recipient, the maximum number of hops, sender address, etc. (not described in details).

2. Node C receives several requests RREQ from node A and selects the best (the minimum



Route discovery between nodes A and C

number of hops) route preparing a response RREP to node A, for example over route C–B–A (this operation is typical for AODV protocol). Before sending a message RREP to node B:

a) node C has to create a unique marker for the route. Moreover, if the data transfer will only simplex (A to C), then only one marker required, and if duplex (A–C and C–A), then two unique markers required. We assume that the transfer will be duplex, so create two unique markers, one for each direction in the route. One of the options for obtaining a unique marker through a super node or gateway for the subnet that will store the table occupied/free tokens. Another variant is to store all markers used in the subnet in the table markers at each node, then the node will be able to create its own unique marker, and notify the other nodes;

b) node C sets one of the receiving code matched filters for the route A–C to the code for bit "0" of the node B and route marker A–C, another code matched filter sets to the code of bit "1" of the node B and route marker A–C. For example, a code matched filter No. 1 and No. 2, respectively;

c) node C adds to the marker table codes of bit "1" with respective route marker and sequence number 1, and the code of bit "0" with respective route marker with sequence number 2. In the end marker table looks as shown in table 4;

***Table* 4.** List of markers (for node C)

Marker	Bit	Src IP	Dest IP	#Transmitter	#Receiver
3	0	IP Node A	IP Node C	–	1
3	1	IP Node A	IP Node C	–	2
2	1	IP Node C	IP Node A	1	–
2	0	IP Node C	IP Node A	2	–

d) node C adds to its routing table route C–A and A–C (see table 5). Then node C unicasts RREP packet containing markers of the routes C–A and A–C to the node B.

***Table* 5.** Routing table of the node C

Dest IP	Next node IP	Hop Count
IP Node A	IP Node B	2

3. Node B receives packet RREP from node C and processes markers contained in the following order:

a) sets two code matched filters for the codes of the node A (for bits "1" and "0" respectively) and route marker A-C (for example, code matched filters 1 and 2), other two code matched filters for the codes of the node C (for bits "1" and "0" respectively) and route marker A–C (for example, code matched filters 3 and 4);

b) for both routes A–C and C–A adds to the marker table codes of bit "1" with respective route marker and sequence number 1, and code of bit "0" with respective route marker with sequence number 2. In the end table 2 is updated (see table 6);

c) adds markers to the marker table of the node B as shown in table 6;

***Table* 6.** List of markers (for node B)

Marker	Bit	Src IP	Dest IP	#Transmitter	#Receiver
3	1	IP Node A	IP Node C	1	1
3	0	IP Node A	IP Node C	2	2
2	1	IP Node C	IP Node A	3	3
2	0	IP Node C	IP Node A	4	4

d) node B adds to its routing table route C–A and A–C (see table 7). Then node C unicasts RREP packet containing markers of the routes C–A and A–C to the node B.

***Table* 7.** Routing table of the node C

Dest IP	Next node IP	Hop Count
IP Node A	IP Node A	1
IP Node C	IP Node C	1

Then node B unicasts RREP packet to the node A which performs the same set of procedures with markers and tables 3–5 as described earlier.

This ends the route discovery phase resulting found a route from node A to C and back, the route is listed in the routing table of each intermediate, the route markers are created and stored in marker tables on each node.

Data relay using "chipset retransmission"

Consider the process of transferring data from node A to node C.

1. Node A checks its marker table. For example, bit "1" is transferred, then the node A finds a row with the *Src IP* of node A, *Dest IP* of node C, *Bit* = 1 and defines "transmitter number" (in this case 1), then the table of dynamic code finds the necessary code for "transmitter number" and generates transmission of pulses in this code.

2. Node B detects on one or more receiving code matched filter by the majority scheme, but the "chipset" is still necessary to check for compliance with its marker (number of code matched filter can accurately determine which marker is configured through a table markers – see Table 6). For example, the first receiving code matched filter of the node B a chip is detected. Markers on the table specifies that the first code matched filter corresponds to marker 3, converting to binary 0000000011 (marker length – 10). Next, according to the majority scheme (or can be applied more strict comparison rule, for example, 90 % of the received pulse positions must match) compares the last 10 time slots of the received signal with a template marker (0000000011 – where 0 is no pulse, and 1 – presence of the pulse). If the check is positive, the markers on the table for the code matched filter is determined by the "transmitter number" (in this case 1), and the chip is passed on (in this case, the code bit "1" of the node B and marker of the route A–C). In case of a negative check the chip is discarded.

3. Node C detects on one of its receiving code matched filters a chip by the majority logic. It also verifies the appropriate marker according to the procedure described above. If the check is positive the chipset is stored in a buffer for further packet accumulation.

Thus, the process of transferring chunk of data presented as a "chip" from node A to node C based on "chipset retransmission" in Ad-Hoc networks using UWB pulse signals and routing protocol AODV.

Conclusions

In this paper a method and detailed algorithm of data relay using "chipset retransmission" has been described. This method does not rely on "store and forward" approach and the total retransmission delay can be estimated as:

$$\tau = L \cdot n \cdot t + \sum^{H-1} n \cdot t \qquad (1)$$

where, L – packet length; n – code length in pulse positions; t – duration of a UWB impulse, s; H – total number of hops for the route.

Note, (1) does not include propagation and processing delays. As can be seen from (1), proposed method allows faster packet delivery to the destination than typical "store and forward" scheme if used in multihop connection.

However several problems still exist. For instance, if "chipset" has been detected with a wrong or erroneous marker or its next hop is unknown than it may lead to significant number of drops. Additionally, there is a need in ensuring uniqueness for each marker and bit code which might be a problem in the distributed networks. For example, on node appearance in the network node starts neighbor discovery with a set of preprogrammed broadcast codes but these codes must be knows by other nodes and vice versa.

Another serious issue is bit and "chipset" errors. It is still a subject to define how receiver node should acknowledge reception of a "chipset". It can be done in a variety of ways: by byte, "chipset" or by a packet on the transport level. Important issue is a redundancy coding of the "chipset" in order to mitigate a bit-error probability, especially of the marker part of the code. This is critically important, because if marker is not correctly extracted from the "chipset" it is not possible to retransmit the whole chipset further. This leads to unnecessary losses of data even though to rest of the chipset can be received and decoded correctly.

Also must be defined a way to distinguish one chipset from another. Though it can be done by sequencing each chipset, if the same route is used for several data transfers losses and propagation and/or processing delays might create some difficulties in multiplexing the final packet in a receive buffer.

1. *F. Cuomo et al.*, “Radio Resource Sharing for Ad Hoc Networking With UWB”, IEEE J. Sel. Areas Commun., vol. 20, no. 9, 2002.
2. *R. Negi and A. Rajeswaran,* “Capacity of Ultra Wide Band Wireless Ad Hoc Networks”, IEEE Trans. Inf. Theory, vol. 6, pp. 3816–3824, 2007.
3. *Бунин С.Г., Плотник К.А.* Ретрансляция импульсных сверхширокополосных сигналов в самоорганизующихся радиосетях // Доклады XX Междунар. конф. “СВЧ-техника и телекоммуникационные технологии”. – К.; Севастополь, 2012. – С. 251.
4. *M. Frodigh et al.*, “Wireless Ad Hoc Networking: The Art of Networking without a Network”, Ericsson Rev., no. 4, 2000.
5. *D. Russel*, “The Principles of Computer Networking”, Cambridge Computer Science Texts, 1990.
6. *V. Sharma, F. Hellstrand,* RFC 3469. Framework for Multi-Protocol Label Switching (MPLS)-based Recovery, IETF, 2003.
7. *C. Perkins,* RFC 3561. Ad hoc On-Demand Distance Vector (AODV) Routing, IETF, 2003, 73 p.

Рекомендована Радою
НДІ телекомунікацій
НТУУ “КПІ”

Надійшла до редакції
19 жовтня 2012 року