Routing Algorithms for Ad-hoc High Speed Hybrid Wireless Networks
Andrey Gaynulin, Vladimir Krylov

gaynulin@tecomgroup.ru
krylov@unc.sci-nnov.ru

Abstract: This article describes the new approach to dynamic recourse allocation and routing protocols in the high speed wireless UWB Ad hoc networks based on hybrid switching technology. The specified network is PAN oriented (Personal Area Network) and can be used for cable replacement purposes. Unlike the majority of the existing Ad Hoc wireless networks, which have a tendency to use the trace packets based routing approach for finding the shortest single path from source to destination, the suggested approach for packet-based traffic routing is based on the node’s internal routing tables and data paths maps. Also, the new routing and resource allocation scheme is suggested for stream-based traffic.

Key words: Ad-hoc networks, bandwidth estimation algorithm, resource management, routing, wireless networks.

INTRODUCTION

An ad hoc wireless network [7] is a dynamic network without any fixed infrastructure and centralized control system. Indeed all components (nodes) of this network are potentially mobile without any wired connection between them, and any node can act as a router. This freedom of mobility, however, results in frequent link disconnections between the components since they have a limited communicating distance determined by their radio range.

Such networks are self-organized, which means that there is no master/slave division and each node is intelligent enough to provide all the functionality for supporting self-organizing algorithm.

Each node in such a network can perform a connection to an existing network (if there is at least one node of that network in his radio range), or start acting as a first member of a new network (if no existing network has been found in his radio range).

Each node is able to initiate the data transmission process as a source, take part in the data transmission process as a router or act as a destination node. It also should be able to handle the network’s map changes properly and correct the data routes, if any links have changed or been broken.

1. Ad-hoc high speed wireless network

This chapter describes the specified ad hoc wireless network structure: physical and logical channels, node functionality and stored information (routing tables, networks data paths’ map) and brief description of self-organizing mechanism, as an important part of networks changes monitoring process.

1.1. Available Resources

The specified network uses the UWB [6] signals for data transmission. The whole band is divided into sub-bands or frequency slots. Each Frequency slot is divided into time slots.

Each physical data channel is represented by a unique time-frequency code (see fig. 1). Time-frequency code is a periodical sequence, which specifies the frequency-hopping rule for current channel.

Figure 1. TFC sequence sample

Each physical channel can be used in two modes – shared channel and dedicated channel.

Shared channel is bidirectional and can be accessed by an unlimited number of nodes within the
network. It’s obvious, that the shared channel should be used for packet-switched traffic transmission. The nodes in the network can use any MACA-based protocol (as it is commonly used in the packet-switched wireless networks) for channel access.

Dedicated channel can be accessed only by two nodes – the source and the destination ones. It is unidirectional, if it is used for high speed stream-based traffic transmission, or bidirectional, if it is used for high speed packet-based traffic transmission purposes.

Each network uses at least one shared channel, which is used for system and signaling packets’ transmission. It is known as “system channel” and it’s time frequency code is the same for each device – node of such class of networks. This channel can also be used for low speed packet-based traffic transmission.

1.2. Self-organizing algorithm

As the specified network is an ad hoc one, so each node should be able to support all the types of dynamic network changes: node’s appearing and disappearing and different nodes’ movements, which causes links’ map to be changed.

The network self-organizing algorithm is based on sending the periodical requests to explore the medium within the radio range and find out if there are new devices, which have not been added to the network yet. Such a request is called the “discovery request”. Discovery request is periodically sent by each node of the network (the unit, which sends such a request is named “caller node”) via the system channel.

Each newly appeared device “listens” the transmitting medium for a specified time interval. If it finds the discovery request among the received signals it tries to enter the found network and become its node. The following algorithm is used: after receiving a discovery request signal the new device sends the answer to the “caller node” about itself.

If the “caller node” has correctly processed the answer then it sends to the new device a request to establish connection and provide each other with information, like routing tables and device parameters. After this operation the “caller node” informs all the rest network nodes about the new node of the network using the signaling packets via system channel.

As all the devices within the network periodically send the discovery request packets, so those packets can be used by other nodes of this network to refresh the network’s map information and notify each other about broken and newly established links, so that the routing information is up to date.

Such an approach may be potentially very resource hungry, but as the network is PAN-oriented, the maximum number of the nodes within this network is usually not more than 20 devices. And discovery requests and routing table updates packets number will not be too much.

2. Dynamic resource allocation protocols

This chapter describes the approaches used for dynamic resource allocation and Quality of Service estimation protocols in the specified UWB wireless ad hoc network. These protocols use the same proactive (or table-driven) routing algorithm for data transmission multiple paths’ searching but apply for different aims, while estimating those paths’ parameters and select the most efficient one for packet-based and stream-based traffic transmission.

2.1. Routing protocol

The major part of existing ad hoc network routing protocols can be broadly categorized into proactive, reactive and hybrid protocols [5].

A proactive protocol, also called a table-driven protocol, is able to offer routing information on the spot. For example, the Destination-Sequence Distance Vector (DSDV) [5] and Wireless Routing Protocol (WRP) [3] belong to the class of proactive protocols. On the other hand, a reactive protocol, also called a source-initiated on-demand protocol, offers routing information with some latency since it usually needs to launch the route discovery process if it has no pre-discovered route. For example, the Ad hoc On Demand Distance Vector (AODV) routing [4] belongs to the class of reactive protocols. Compared to the reactive protocols, the proactive protocols provide smaller delay in sending out a packet. However, they use up more bandwidth since they need to periodically broadcast the routes for all nodes in the network. The bandwidth usage for control messages is proportional to the size of the network and as it has already been mentioned, as this network is PAN-oriented, then this parameter is pretty small. The hybrid protocols (HSLS, ZRP) use the combined proactive/reactive schema, so that, for example, ZRP protocol uses IARP as proactive for the neighbor nodes and IERP as a reactive component.

As the described network already uses to collect the entire network’s map data for self-organizing issues, so it is convenient to use table – driven approach for routing data transmission paths.

The described network belongs to the ad hoc class of the networks and all the nodes within it are mobile, so searching of single path is not enough for reliable connection, because when any changes in network’s map occur, it will be necessary to re-route the transmitting data and lose the established connection.

To increase the reliability of the data transmission it is necessary to find some additional alternative routes (if they exists) and use them if any link break occurs. As the network is usually is not very large and the whole routing process is done on the source node’s side, then it is useful to find all the available acyclic independent routes from the source to destination.

Two routes are called “independent” when a set of nodes forming one of the routes is not a subset of nodes forming the other route (see fig. 2):
The independence criteria is very important for the routing due to the following reason: if routes are dependent, data transmitting via the longer route is not efficient comparing with the shorter one, because additional packet retransmission will increase the summary traffic in the network and waste the whole network’s throughput.

Implementation of the routing algorithm is based on the modified wave Lee algorithm. The source Lee algorithm does the following: on the first step it assigns the 0 index to the source node. On the second step, the source node propagates the wave to its neighbors in the network’s graph and assigns to them the 1 index. After it, the node propagating wave (which is the source node on this step) checks, if there is a destination node among the marked nodes. If it is, then the route is found and the propagation is stopped. Otherwise, each node with index 1 sequentially propagates its own wave and marks all unmarked neighbors with index 2 (the assigned index is counted by adding 1 to the index of the propagating node) and checks if there is a destination node among the marked ones. Then this process is repeated until the destination node is found.

This algorithm is pretty fast and guarantees finding of the single shortest acyclic path from node to destination, but in spite of all its advantages, there are some disadvantages which makes it unsuitable for our purposes in the unmodified state: it finds only single path when we need to find all the available acyclic independent paths from source to destination.

To accomplish this task it is necessary to make some modifications to this algorithm. First of all, to find all the available paths this algorithm should be able to mark one node more than once. So On each step, when a node is propagating the wave and mark all the neighbors (including the marked ones) with the certain index, it not only the assigns the index to the node, but also specifies the route, which is currently ended with this index. Such an approach allows finding multiple paths to the destination, but does not guarantee those paths to be independent and acyclic. To solve this problem on each step it is necessary to make the validation that the path newly assigned for the node is independent from those paths, which have already been assigned to this node. If it is, then the paths are independent and this index (and the corresponding path) can be assigned to this node. Such an approach allows adding independency verification. For acyclic path’s routing we suggest to add a modification, which will make the node to verify, if the investigating node has already been included in the pre-routed path, and if it is, then the specified node is not marked.

The algorithm is stopped, when there is no more routes, except for the destination one, which can propagate the wave, but to reduce the number of computations, which are necessary to find all the necessary paths, we have added an additional modification, which allows to stop algorithm on the specified wave index (on each step it is just verifies, if the current index is less than the terminating one).

2.2. Routing sample

Let’s illustrate the algorithms workflow on the following example: the network consists of six nodes (see fig. 3), which are numbered from 1 to 6, it is necessary to find all the available paths from the source node (which number is 1) to the destination node (which number is 6). The specified network is represented with the following graph, where octagons represent nodes and the lines between nodes represent existing links.

![Sample network](image)

Figure 3. Sample network

On the first step the algorithm (see fig. 4) assigns the 0 index to the source node 1. On the second step (see fig. 4) it propagates the wave and marks unmarked neighbors 2, 3 and 5 with index 2 (the assigned index is counted by adding 1 to the index of the propagating node) and checks if there is a destination node among the marked ones. Then this process is repeated until the destination node is found.

![Routing steps 1 (left) and 2 (right)](image)

Figure 4. Routing steps 1 (left) and 2 (right)

On the third step (see fig. 5) nodes 2, 3 and 5 propagates the wave with index 2. Node 2 marks nodes 4 (it cannot mark node 1 and node 3 cause those routes will be either cyclic or dependent). Node 3 marks nodes 4 and 6 (it cannot mark node 1 and node 2 cause those routes will be either cyclic or dependent). Node 5 marks node 6 (it cannot mark node 1 because this route will be cyclic).
On the fourth step (see fig. 6) the only node, which can propagate the wave is node 4 (node 6 is the destination one and does not propagate the wave). It tries to propagate the wave for the index 3 for both available paths (1-2, 1-3) and for one of them (1-2) it is able to mark node 6, for the other (1-3) it is not, because the resulting path (1-3-4-6) path is dependent with another existing path (1-3-6) already available on the node 6. Node 2 and 3 cannot be marked, because those routes will be either cyclic or dependent.

On the next step the algorithm is stopped, as the only node, which can propagate the wave is the destination one.

At this point there are three available routes from node 1 to node 6. They are 1-3-6, 1-5-6 and 1-2-4-6. All these routes are acyclic and independent.

To illustrate the computing limitation of the modified algorithm let’s limit the maximum index with 3. In this case, the algorithm will be stopped when the second wave has been propagated and the result of such routing will be 2 paths: 1-3-6, 1-5-6.

3. Stream-based traffic routing

The stream-based high-speed traffic is suggested to be transmitted in the described network using the dedicated channels. If the throughput of one dedicated channel is not enough to transmit the requested traffic stream, then it is possible to allocate necessary number of channels for data transmission and use them simultaneously.

As there is limited number of dedicated channels, available within the specified network, then the main aim of the dynamic resource allocation protocol for such issues is to provide most efficient way of network’s available bandwidth usage.

The first stage of this process is finding all available acyclic independent routes from the source node to the destination one, which provides the necessary throughput. In case of stream-based traffic transmission the throughput of the route may be measured in the number of dedicated channels, available on its nodes.

This routine is performed using the routing protocol, described before. The only thing that should be additionally checked while looking for the available routes is the check for the number of available channels on each node within the route. It is obvious, that if a node uses dedicated channel number 1 for transmitting data to receiving node, then the receiving node cannot use dedicated channel 1 neither for transmitting nor for receiving any other data. Those nodes, which are within the radio range of the transmitting node and not within the radio range of the receiving node, can use this channel for transmitting, but cannot use it for receiving signals. Those nodes, which are within the radio range of the transmitting node and not within the radio range of the receiving node, can use this channel for receiving, but cannot use it for transmitting signals. Those nodes, which are within the radio range of both the receiving and transmitting nodes, cannot use this channel neither for receiving, nor for transmitting signals.

That is why after completing routing process each found channel should be checked, if it is possible to transmit the requested number of dedicated channels’ occupying traffic through it.

This check is performed like the following: each node has a set of channels available for transmitting and a set of channels available for receiving. Those channels, which have already been occupied, are marked with “occupy” flag. So on each pair of the nodes from the start of the route the set of available channels for transmitting on the transmitting node is compared with the set of available channels on the receiving node and if the intersection of these sets results in the number of channels equal or more then the required one then the necessary number of channels are marked with “occupy” flag and this process continues until the destination node is reached. Obviously it is better to occupy first of all those channels, which are not available for receiving on the transmitting node and, in the same time, which are not available for transmitting on the receiving node.

If the route appears to be correct according to the previous condition, then it may be used for data transmission.

It may appear that there is more than one route, which satisfies the requirements and it is necessary to select one of them, which is most efficient in whole network’s throughput usage. Such a selection may be done based on the following reason.

Each node, which is occupying a channel for transmitting data, automatically permits using of this channel for receiving and/or transmitting data for all his neighbors and the neighbors of the receiving nodes, as it has been described in this section below.
So, the more neighbors the transmitting node has, the more channels of the whole networks throughput become unavailable. Let us call the number of neighbors of the node the “node’s weight”. So the more weight of the node is, the less efficiently it uses network’s throughput. Let us call the weight of the route the sum of weights of all the nodes within the route. Obviously, the more route’s weight is, the less efficient route uses the whole network’s throughput. So, the most efficient route will be the one, which weighs less.

Let’s illustrate the resource allocation process on the same network, which was used in the previous section. We guess that the transmitted traffic stream can be transmitted via one dedicated channel and the dedicated channels’ pool consists of three channels. All the channels within the network are initially free (none of them are marked with the “occupy” flag).

So, as it was shown in the previous section, on the first stage, the specified routing protocol finds 3 available routes from node 1 to node 6. They are 1-3-6, 1-5-6 and 1-2-4-6. All of them are acyclic and independent.

Now let’s measure the weight of each of these routes. For the first route (1-3-6) it is $3+4+3 = 10$. For the second route (1-5-6) it is $3+2+3 = 8$. For the third route (1-2-4-6) it is $3+3+3+3 = 12$.

As the second route “weighs” less than others, so it should be more efficient in whole network’s throughput usage.

Let’s check it by representing each route’s resource allocation map on each node.

For the first route (1-3-6) the following resource allocation map can be used:

```
  rcv   trm
node 1   node 2   node 3
  rcv
node 4   node 5   node 6
```

**Figure 7. Resource allocation map (route 1: 1-3-6)**

This map is representing the following – for each node there are two sets of channels: channels, available for receiving (“rcv” row) and channels, available for transmitting (“trm” row). White color means that channel is free, gray color or shading means that the channel is occupied for data transmitting. On this resource allocation map gray color marks the channel, used for transmitting from on link 1-3, cross hatching – channel for transmitting on link 3-6.

For the route 1-5-6 resource allocation map is:

```
  rcv   trm
node 1   node 2   node 3
  rcv
node 4   node 5   node 6
```

**Figure 8. Resource allocation map (route 2: 1-5-6)**

On this resource allocation map gray color marks the channel, used for transmitting from on link 1-5, cross hatching – channel for transmitting on link 5-6.

For the route 1-2-4-6 resource allocation map is:

```
  rcv   trm
node 1   node 2   node 3
  rcv
node 4   node 5   node 6
```

**Figure 9. Resource allocation map (route 3: 1-2-4-6)**

On this resource allocation map gray color marks the channel, used for transmitting from on link 1-2, cross hatching – channel for transmitting on link 2-4, horizontal hatching - channel for transmitting on link 4-6.

It’s obvious that the most light-weight route 1-5-6 really appears to be the most efficient while using the whole network’s throughput.

After selecting the route for data transmission the source node broadcasts system packets with information about the route used and resources occupied while using it. On getting confirmations from all other nodes from the network it starts establishing the route and transmitting data.

4. Packet-based traffic routing

The packet-based low-speed traffic is suggested to be transmitted in the described network using the shared system channel. In this case all the system data packets have higher priority and user data packets have low priority, so that even if the network is highly loaded with user data transmission it does not affect the system information exchange.

For packet switched data the routing protocol should be able to provide the requested QoS, for example the average delay time, basing on the following parameters: the requested speed, available system channel’s bandwidth and current system channel’s payload. If the network is fully connected, it is possible to represent it with an equation known from the queuing systems theory like the M/G/1 model with packets’ priorities [1]. In this case each network node can easily measure the current network’s payload (because it is able to receive all the
packets, transmitted in the network) and provide a prediction for the average delay mean [2], based on the measured payload and requested transmitting speed.

For example if the packets’ length distribution looks like the following (it was measured on the Nizhny Novgorod state university router for 2 weeks):

**Figure 10. Packets’ length distribution**

and the intensity of system packets is about 20-50 packets per second per node, then for the system channel with the 500 Mbps throughput the average packet’s queuing time is following for data and system packets:

**Figure 10. Average packets’ delay (data packets – upper, system packets - down)**

X scale – the utilization value (in percents), Y scale – the average packet delay time in seconds.

For more common case, when the network is not fully connected, then such a model cannot be used for getting precise average delay time value. And it’s necessary to use an adaptive delay measurement algorithm, which counts the average delay time relying on the M/G/1 analytical model for the source node’s cluster, but uses the value’s correction based on the RTT measurement for the system packets, going from the source node to the destination one. The RTT system packets are marked with the received time on each node of the route from the source node to the destination node and back. So when this packet returns to the source node such markers allow estimating the user’s packet’s loading in the specified cluster by matching of the system packet delay time with the payload in the cluster using the M/G/1 model computations. Such calculations are not very precise in the low-loaded networks (less then 5% of the available throughput is used), because the time shift on the neighbor nodes may be comparable with the average delay time for system packet transmitting. But as the payload grows, so the accuracy of the traffic payload estimation becomes better, even it is still provides the upper estimate for the cluster’s payload.

So, after estimating the current cluster’s payload, it is possible to predict the average delay time by using the model, described below, by increasing the utilization value on the requested transmission speed divided by the channels throughput. But it is not enough – for the source (non – retransmitting) node this utilization value shift should be doubled (as the receiving node retransmits packets and the real number of packets is twice as more in this cluster, then the input number) and for the retransmitting nodes (except for the one before the destination node) this value should be tripled, because in its cluster there are packets sent by the previous node, there are packets, sent by the current retransmitting node and there are packets, sent by the next retransmitting node. If all those rules are included in the estimation, then the summary average delay will be equal to the sum of the delays on each node of the route, except for the destination one. And this summary average delay will be the exact value to compare with the user requested traffic delay and predicts the upper estimation for the QoS, provided by the network.

It’s obvious that such estimation depends on the accuracy of the information about the system packets delays, so that for each node it is necessary to periodically send “ping” system packets not only to the neighbors in radio range (like it is done with the discovery requests packets in self – organizing algorithm), but also to the distant randomly selected neighbors. Obviously, not only the source node, but also each node on the route of the ping packet may extract some average delay information from this packet, which allows minimizing the number of such packets and still keeping the delay information up-to-date. So, we suggest using of the following algorithm – each node periodically sends such a packet to the randomly selected node if information about the delays to this node is not up-to-date. On receiving a response for this packet (or on extracting necessary information from the “ping” packet, sent by another node) the node marks this information as an up-to-date and specifies the period of validity starting from that certain moment.

5. Conclusion

In this paper we have focused on the dynamic resource allocation problems in the hybrid high-speed UWB network with a pool of available dedicated channels and a shared packet-switched system channel.

We have presented a new routing scheme based on modified Lee wave algorithm, which allows coping with these problems. The proposed protocol finds all the available independent routes from source to
destination, select one of them, which is the most efficient by one of the applied criteria (either average packet queuing delay time for packet-based traffic or whole network’s throughput usage efficiency for stream-based traffic). It also stores alternative path for each selected route, so that if the link break occurs the data flow can be easily redirected using another data path.

In general, proactive routing protocols uses some portion of the limited wireless bandwidth due to routing control information broadcasting in acquiring the entire network topology and in keeping it accurate. But the described routing protocol converts this drawback into an advantage by using the information collected by the broadcast message not only to quickly perform all the path resolution issues, but also to provide some failover protection and quick data flow restore in link failure situations.

ACKNOWLEDGMENT

The authors wish to thank collective and the head of the chair of theory of circuits and telecommunications (NSTU) for their assistance. This work was supported in part by a grant from Foundation for Assistance of Small Innovative Enterprises (FASIE), Russia.

REFERENCES