Computer Communications 35 (2012) 1574–1589

Contents lists available at SciVerse ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Onto scalable Ad-hoc networks: Deferred Routing

David Palma *, Marilia Curado

Centre for Informatics and Systems, University of Coimbra (CISUC), Coimbra, Portugal

ARTICLE INFO

Article history:
Received 9 August 2010
Received in revised form 24 April 2012
Accepted 28 April 2012
Available online 8 May 2012

Keywords:
Scalability
Deferred Routing
Mobility
Ad-hoc networks

ABSTRACT

A new approach for scalable routing in infrastructure-less wireless mobile networks is presented, requiring minor changes in existing link-state routing protocols and aggregating routing information with different levels of granularity into a hierarchy. The obtained results show that this routing scheme has better performance and is more efficient, exchanging up to ten times less routing traffic than other routing solutions. The proposed solution is particularly useful in large-scale scenarios, being robust against mobility phenomena, allowing limited wireless devices such as sensors and mobile phones to be part of these networks.

1. Introduction

The generalized spread of wireless capable devices has recently allowed users to be more and more connected to different services, requiring only the presence of an infrastructure capable of supporting them. However, in the near future, users are expected to own several gadgets requiring wireless connections [1], demanding a considerable amount of physical resources from the available infrastructures.

The required infrastructure for the massive every day use of wireless devices is still not a reality in many areas, and represent a major problem on the creation of wireless networks for these devices. In order to cope with the lack of existing infrastructures, the concept of Ad-hoc networks has been proposed, creating a multi-hop network where each wireless node behaves as router. However, these networks which typically need to handle mobility, do not scale with the existing routing protocols.

Considering mobile Ad-hoc networks (MANETs) for future wireless communication, a number of routing schemes has already been proposed. Focusing on the topology-based routing protocols, which do not require any additional mechanism for node's position awareness such as Global Positioning Systems (GPS) or other positioning schemes, proposals have been developed for both proactive and reactive routing protocols.

Proactive routing protocols for MANETs were inspired by the typical protocols used in wired networks, based on the periodic exchange of update messages in order to maintain the routing tables. Some well known proactive routing protocols are the "highly dynamic Destination Sequenced Distance-Vector routing protocol" (DSDV) [2], the "Clusterhead Gateway Switch Routing protocol" (CGSR) [3], the "Dynamic Address RouTing for scalable Ad-hoc and mesh networks" (DART) [4] and the "Optimized Link-State Routing protocol" (OLSR) [5].

Proposed as an alternative to the expensive periodic update of proactive routing schemes, reactive protocols were introduced, performing route discoveries on-demand and avoiding the waste of resources experienced with proactive solutions. This approach seems more suitable for mobile Ad-hoc networks where topology changes occur constantly. However, on-demand solutions suffer from an initial delay on retrieving a routing path which may not be acceptable and the flooding for route retrieval may be too expensive. In this category, the "Dynamic Source Routing Protocol" (DSR) [6], the "Ad-hoc On-Demand Distance Vector Routing Protocol" (AODV) [7] or, one of the most recent, the "Dynamic MANET On-demand Routing Protocol" (DYMO) [8], represent some of the existing reactive protocols.

Despite the existing literature, efficiently keeping a MANET scalable is still an open issue. The usage of clusters or alternative hierarchies, such as [9] or [10], aim specifically at this issue, but typically require complex mechanisms or additional hardware to achieve their goals. Other approaches such as the "Fisheye" and "Hazy Sighted Link State" Routing Protocols [11,12], aim at maintaining scalable routing mechanisms by having imprecise or slightly out-of-date routing information regarding distant nodes. While these mechanisms reduce the amount of routing information and new versions have been proposed [13], they do not support clusters nor do they take into account a well defined network hierarchy as this work does.

A new concept named as Deferred Routing is defined, using both clusters and a well defined hierarchy. Moreover, the required changes for a link-state routing scheme are presented, without the

* Corresponding author. Tel.: +351 965637055.
E-mail addresses: palma@dei.uc.pt (D. Palma), marilia@dei.uc.pt (M. Curado).

0140-3664/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comcom.2012.04.026

need of any particular changes in the existing wireless technologies. By assigning different granularity levels of routing information to each existing cluster, scalable routing in MANETs is achieved. Each node will solely keep detailed information about its own cluster, and will maintain aggregated information about the network according to a pre-defined cluster hierarchy, allowing smaller and more stable routing tables. Routing decisions are cluster-based, being postponed to further clusters in the hierarchy if necessary, without previously knowing the entire path taken. This routing approach can be used with any link-state routing protocol, such as the OLSR protocol, requiring only minor changes to its routing messages.

In Section 2 the Deferred Routing approach is described, presenting the overall idea behind the concept and the necessary changes required to adapt a link-state routing protocol. A methodology for the evaluation of the Deferred Routing scheme is then defined in Section 3, presenting two distinct scenarios with different characteristics representing a University Department and a possible campus with a total area of 3 km^2. The obtained results are presented in Section 4, comparing them with the OLSR, C-OLSR and AODV routing protocols. An overview of existing related work is included in Section 5 and finally, in Section 6, the final thoughts on this work are presented.

2. Description of the Deferred Routing concept

This concept presents a new perspective on how scalable routing can be achieved in large wireless networks, handling mobility phenomenons with minor disruptions. An important aspect of the presented work is the usage of the Deferred Routing approach, applied in conjunction with a typical clustering mechanism and an existing link-state routing protocol, requiring only minor changes to fully implement this routing scheme. For instance, the generalized Max–Min [14], or the NSLOC [15] clustering algorithms could be used for grouping the nodes according to a predefined number hops or by using other parameters such as nodes' geographic position or context, while the OLSR protocol could be used for the link-state routing.

2.1. Deferred Routing

Deferred Routing can shortly be explained as a routing procedure where nodes which lack the necessary information to reach a destination, postpone this task to other nodes, by choosing appropriate gateway (Gw) nodes, ensuring that this decision is taken by nodes closer to the desired destination.

Behaviours close to Deferred Routing can be found in everyday tasks such as driving. Typically, when one drives within one's living/working area, most of the possible routes are already well known and predefined, not requiring much thinking when choosing the best path to be taken. However, when a journey to a less well known location has to be taken, some careful planning is necessary. Despite being easy to have access to maps with all the existing roads, choosing the final route to reach the desired destination may require considering aspects, such as distance, traffic and road quality.

The most common solution for routing when driving is to choose highways, heading towards known locations, to specific exit points such as borders. In fact, when travelling long distances, it might not only be harder to get accurate maps, but it is also a demanding task to consider all the possible alternative routes to reach a destination. Again, a typical and easy solution to solve this problem is to simply drive towards well known and marked areas, such as capitals, important cities, regions or even countries. These landmarks act as gateways for the driver, and throughout the journey, more and more detailed information will be available on the road signs when the driver gets closer to a certain destination.

Taking into account this everyday routing approach, adapting it to computer networks is straightforward and will allow a significant improvement in routing performance when compared with typical routing approaches for wireless Ad-hoc networks. Moreover, this scheme limits the impact of node mobility as it relies on condensed views of the network, such that a node moving from one cluster to another (a cluster can be seen as a city or region in a map), will not impact someone travelling from a more distant cluster (which can correspond to a country).

2.1.1. A cluster hierarchy for Deferred Routing

Similarly to any road map, a well defined organization of clusters is essential to allow an efficient understanding of the network. For this purpose a binary tree hierarchy is defined for Deferred Routing, assigning clusters to contexts and defining different granularity levels. The clustering process is out of the scope of this work as any clustering scheme can be used; thus all cluster maintenance aspects such as the update of nodes' Cluster ID depend on the used clustering scheme. Moreover, the cluster creation mechanism may rely on existing context information such as user provided information or even positioning awareness [16].

Different hierarchical routing approaches have already been defined by several authors to handle routing in Ad-hoc networks, however these works do not effectively handle the dynamic behaviour of such networks, relying on well defined structures which are hard to maintain or on specific hardware for devices connecting different levels of the hierarchy, as further described in Section 5.

The proposed hierarchy for Deferred Routing is mostly virtual, except for the leaf clusters. This hierarchy mimics a typical road map, where a cluster can be within different virtual clusters with several granularity levels. This allows information to be aggregated with different levels of detail, similarly to a map which has countries, regions, cities, blocks, and so on. This allows the optimization of a routing table, keeping more concise information about distant clusters, and being more resilient to node mobility, since nodes changing clusters will not render any changes to most of the nodes' routing tables as they only keep aggregated views of the network. Only neighbour clusters are affected by the addition or deletion of a node in their brother cluster, reducing the normal overhead of such an operation.

Algorithm 1: View determination algorithm

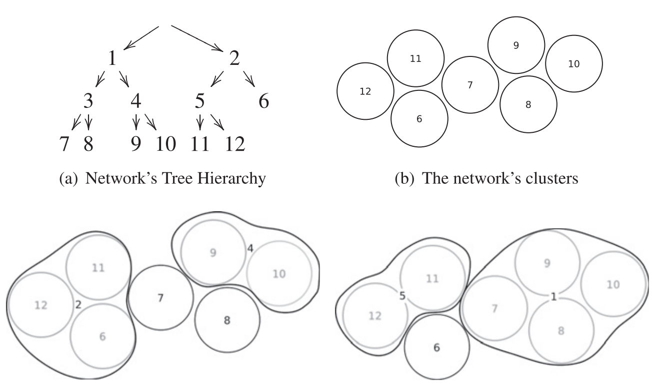
procedure Determine_View($CID_{own}, CID_{foreign}$)
  $level_{own} \leftarrow$ GET_LEVEL(CID_{own})
  $level_{foreign} \leftarrow$ GET_LEVEL($CID_{foreign}$)
  if $level_{foreign} > level_{own}$ **then** // Needs to be Raised
    $CID_{foreign} \leftarrow$ JOIN_VIEW($CID_{foreign}, level_{foreign} - level_{own}$)
  else
    if $level_{foreign} < level_{own}$ **then**
      $CID_{own} \leftarrow$ JOIN_VIEW($CID_{own}, level_{own} - level_{foreign}$)
    end if
  end if
  if CID_{own} **mod** $2 = 0$ **then** //To check if the CIDs are "brothers"
    $even \leftarrow -1$
  else
    $even \leftarrow 1$
  end if
  while $CID_{own} + even \neq CID_{foreign}$ **and** $CID_{own} \neq CID_{foreign}$ **do**
    // Perform a join until both CIDs are at the same level
    $CID_{foreign} \leftarrow$ JOIN_VIEW($CID_{foreign}, 1$)
    $CID_{own} \leftarrow$ JOIN_VIEW($CID_{foreign}, 1$)

(*continued on next page*)

if CID_{own} **mod** $2 = 0$ **then**
$even \leftarrow -1$
else
$even \leftarrow 1$
end if
end while
return CID_{own}
end procedure
procedure JOIN_VIEW(CID, n_{level})
$CID_{new} \leftarrow \lceil [CID - (2^{n_{level}+1} - 2)]/2^{n_{level}} \rceil$
return CID_{new}
end procedure
procedure GET_LEVEL(CID)
$Level \leftarrow \lfloor log_2(CID + 1) \rfloor$
return *Level*
end procedure

Fig. 1 represents a network hierarchy suitable for Deferred Routing and the corresponding clusters. The previously mentioned binary tree is shown in Fig. 1(a), where the identifiers (IDs) from 1 to 5 are aggregated views of the network, containing several clusters. The actual network is depicted in Fig. 1(b), as it would be recognized by any cluster in a typical routing approach supporting clusters. However, this way of representing the network is prone to mobility disruptions, and requires a large amount of routing information. With Deferred Routing each cluster has its own unique view of the network, aggregating the remaining clusters into new ones according to their position within the hierarchy. For example, with the given tree, a node inserted into the clusters 7 or 8 will only perceive the network as shown in Fig. 1(c), in such a way that sibling clusters see each other, and clusters in higher hierarchy levels are joined into new broader clusters. An additional network view is presented in Fig. 1(d), showing that a cluster in a higher level will not have more detail about a neighbour cluster, reinforcing the concept of defining independent regions.

In order to implement the desired network organization, a node within a cluster must be able to determine its own position in the hierarchy tree, and how it should keep information about other clusters. The necessary operations to ensure this process are presented in Algorithm 1, allowing the determination of the correct view of neighbour clusters, by aggregating them according to their level. For instance, referring to Fig. 1 and considering nodes in cluster 7, any routing message received from cluster 9 will be seen as a message received from cluster 4. This view is determined by considering the hierarchical position of both clusters, using the View Determination Algorithm. Since clusters 7 and 9 are not siblings from the same parent cluster, each one of them will be brought up to its corresponding parent's level until both are brother siblings.



(a) Network's Tree Hierarchy (b) The network's clusters (c) Clusters' 7 and 8 network view (d) Cluster's 6 network view

Fig. 1. Possible network perspectives for the tree hierarchy.

Thus, cluster view 7 will be raised in the hierarchy (*Join_View (7)*) becoming view 3 and cluster view 9, following the same procedure will become view 4. Thus, any node in cluster 7 will only perceive routing messages received from cluster 4 even though they might have been sent from cluster 9 or 10. By using such abstraction and hierarchy, less cluster definitions are required and node mobility, for instance from cluster 9 to 10 is transparent for other clusters.

Another important task is to check whether or not the observed information corresponds to an aggregated view containing a nodes' own cluster. This will represent an unnecessary set of information, since it has less detail when compared to what is already known by a node. Algorithm 2 performs the task of determining if a cluster (*contained*), is within another cluster (*container*). This procedure is important for nodes to assess whether or not to discard received aggregated information which they may already possess with higher detail. As an example, routing messages received in cluster 7 from cluster 9, which are "seen" as messages received from cluster 4, will contain information about cluster 3. This results from nodes in cluster 9, which receive routing messages from this cluster due to the View Determination Algorithm (both clusters 7 and 8 are aggregated into cluster view 3). However, when nodes in cluster 7 receive a routing message containing information about cluster view 3, they must disregard it as they belong to this cluster thus having more detailed information about it.

2.2. Deferred Routing in a link-state protocol

As previously mentioned, implementing the Deferred Routing paradigm in a typical link-state protocol is simple, requiring only some changes in existing routing messages and minor adjustments in the protocols' procedures such as including a cluster identifier in each message and trigger inter cluster routing when required. Next, all the necessary changes for the OLSR protocol to support Deferred Routing are presented. Since a new network hierarchy is used, the required modifications will be focused on including cluster information in routing messages and mechanisms to handle the different network views. Moreover, routing between clusters is introduced, ensuring data delivery in clustered networks.

2.2.1. Routing messages

With the purpose of avoiding the creation of additional routing messages, the existing routing packets sent by the OLSR link-state routing protocol should be modified in order to support Deferred Routing. The necessary changes require the inclusion of a Cluster Identifier (CID) in every message, representing each cluster, as names represent geographical locations in a map. Moreover, information about Cluster Connectivity should also be included, in both *HELLO* and Topology Control (*TC*) messages, if the OLSR protocol is considered.

Algorithm 2: Cluster Containment Algorithm

procedure CONTAINS_CLUSTER (*container*, *contained*)
if $contained < container$ **then**
return *FALSE*
end if
$container_{level} \leftarrow$ GET_LEVEL($container$)
$contained_{level} \leftarrow$ GET_LEVEL($contained$)
$gap \leftarrow contained_{level} - container_{level}$
$contained \leftarrow$ JOIN_VIEW $contained, gap$
if $contained = container$ **then**
return *TRUE*
else
return *FALSE*
end if
end procedure

Algorithm 3: Send Hello Algorithm

```
procedure SEND_HELLO_MESSAGE
    ...
    //Share IP Mappings
    message.ip_mappings_shared ← LIST_CREATE()
  for each mapping_entry in temp_ip_cluster_mapping_table
    // Allocate a new Mapping Structure
    new_tuple ← Pmo_Alloc(Ip_Cluster_Mapping_pmh)
    new_tuple.src_addr ← mapping_entry.ip_addr
    new_tuple.cluster_id ← mapping_entry.cluster_id
    // Insert new Mapping Entry into a list
    LIST_INSERT (message.ip_mappings_shared, new_tuple, TAIL)
  endfor
    // Process Temp Connectivity
  for each cluster_entry in temp_cluster_connectivity_table do
    for each gw_entry in cluster_entry.gw_table do
      if gw_entry.exp_time > current_time then
        Cluster_Connectivity_Create(cluster_connectivity_table,
        gw_entry.gw_addr, cluster_entry.cluster_id, gw_entry.hop_count,
        gw_entry.originator, gw_entry.seq_num, gw_entry.age,
         gw_entry.exp_time)
    end if
    // Deallocate Temp Mapping Entry
    Pmo_DeAlloc(gw_entry)
  endfor
    // Process Temp Cluster Connectivity
    Pmo_DeAlloc (cluster_entry)
  endfor
  // Share Cluster Connectivity
  message.cluster_connectivity ← LIST_CREATE()
  for each cluster_entry in cluster_connectivity_table
    // Create and Allocate a new Cluster
    conn_tuple ← Pmo_Alloc(Connectivity_Tuple_pmh)
    // Create a new Cluster List
    conn_tuple.gateways ← LIST_CREATE()
    for each gw_entry in cluster_entry.gw_table do
        if gw_entry.exp_time < 0 then
          continue
        end if
      if gw_entry.seq_num = -1
      gw_entry.seq_num ← message.seq_num
        end if
    // Allocate a new Cluster Connectivity Structure
    new_tuple ← Pmo_AllocCluster_Connectivity_Tuple_pmh
    // Set the available parameters
    new_tuple.hop_count ← gw_entry.hop_count
    new_tuple.gw_addr ← gw_entry.gateway
    new_tuple.age ← gw_entry.age
    new_tuple.originator ← gw_entry.originator
    new_tuple.seq_num ← gw_entry.seq_num
    new_tuple.exp_time gw_entry.exp_time - current_time
    // Insert Cluster Connectivity structure into list
    LIST_INSERT(conn_tuple.gateways, new_tuple, TAIL)
  end for
    // Insert the newly created Cluster into list
    LIST_INSERT(message.cluster_connectivity, conn_tuple, TAIL)
  end for
  ...
end procedure
```

Algorithm 4: Message Received Algorithm

```
procedure Process_Routing_Message(message)
  ...
  src_addr ← message.src_ip
  cluster_id message.cluster_id
  Ip_Cluster_Mapping_Create(ip_cluster_mapping_table, src_addr,
   cluster_id, true)
  for each mapping_entry in message.ip_mappings_shared do
    // If this information comes from within the same cluster,
  it will be always processed
    // otherwise only if it does not contain an aggregated view
  of this cluster
    if cluster_id ≠ own_cluster_id and
        Contains_Cluster(mapping_entry.cluster_id, own_cluster_id)
    then
      continue
    end if
    Ip_Cluster_Mapping_Create(ip_cluster_mapping_table,
     mapping_entry.ip_addr,
             mapping_entry.cluster_id, false)
  end for
  if cluster_id ≠ own_cluster_id // This Node is a Gateway
      Gw_Connectivity_Create(gw_connectivity_table, src_addr,
      cluster_id, 1, own_addr,
            -1, GW_TIMEC)
      for each cluster_entry in message.cluster_connectivity do
        // Ignore information regarding the same cluster, as
  well as
        // information about the received cluster's message
        if Determine_View(cluster_id) =
         Determine_View(cluster_entry.cluster_id)
            and Contains_Cluster(cluster_entry.cluster_id,
             own_cluster_id)
          continue
        end if
        for each gw_entry in cluster_entry.gateways do
            Gw_Connectivity_Create(gw_connectivity_table,
              src_addr, cluster_entry.cluster_id, gw_entry.hop_count + 1,
                  gw_entry.originator, gw_entry.seq_num,
                  gw_entry.exp_time)
        end for
      end for
      // Since this message comes from a different cluster, it
  must not be processed
      // by the OLSR protocol which only handles Intra-
  Cluster Messages
      return // Consequently, end the procedure
  else // Message received from the same cluster
    for each cluster_entry in message.cluster_connectivity do
      // Ignore information regarding the own cluster
      (should only occur with out-of-date moving nodes)
      if Contains_Cluster(cluster_entry.cluster_id, own_cluster_id)
then
        continue
      end if
      for each gw_entry in cluster_entry.gateways do
        // A node does not need information about its own
  connectivity
        if gw_entry.gw_addr = own_addr then
            continue
        end if
```

(continued on next page)

* (continued)

Algorithm 4: Message Received Algorithm

```
            Cluster_Connectivity_Create(cluster_connectivity_table,
              gw_entry.gw_addr, cluster_entry.cluster_id, gw_entry.hop_count,
              gw_entry.originator, gw_entry.seq_num, gw_entry.age,
              gw_entry.exp_time)
          end for
        end for
      end if
      ...//Link-state Routing Procedures
  end procedure
```

The Cluster Connectivity information is the most relevant and the only mandatory information to be added to existing routing messages in conjunction with the Cluster Identifier. This cluster information consists of a list of the existing gateway nodes available within the considered cluster to another cluster and its own characteristics. Each Gw entry contains information about the required number of hops to reach the announced cluster (these hops concern cluster hops, not node hops), and other parameters such as its *age* and *originator*, which will be explained with more detail later in this work. When new Cluster Connectivity information is created by a Gw node, it is temporarily stored until a new periodic routing message is sent, including this information. Such behaviour allows more accurate routing information between all the nodes, avoiding incoherent routing tables and overheads.

By using, a clustering algorithm information, the mapping between Node Identifiers and their respective cluster could be known. However, in order to be independent from any other mechanisms, the inclusion of this additional information is also considered and should occur when a *HELLO* message is being prepared to be sent. These Node ID mapping entries ($mapping_{entry}$) correspond to the pair of a node (ip_{addr}) and its containing cluster ($cluster_{id}$). The creation of these messages is shown in Algorithm 3.

In order to reduce unnecessary overheads, no IP mapping information is added to *TC* messages since this information is already present in *HELLO* messages. The remaining instructions in Algorithm 3 are the same for both routing messages. Moreover, the used IP mappings could be further aggregated, similarly to [17], even though Deferred Routing does not rely specifically on any IP distribution scheme, being able to work with any unique identifier.

IP mapping information is only included in routing messages for a short period of time whenever any new information is created. For instance, if a node changes its cluster or a new node enters the network, this new IP address will be sent through the affected clusters which are the new node's cluster, its sibling and the cluster from where the node departed in the case of a moving node.

As a result of nodes being separated into different routing clusters, each message must be processed according to its own cluster. Therefore, a routing message is only received by the adopted link-state routing protocol if it belongs to the same cluster of the node that received it, ensuring intra-cluster routing. In order to implement this behaviour, the routing procedures were modified to firstly obtain all the known IP Mappings – when processing *HELLO* messages – and secondly to gather the existing cluster information of the network. The information obtained from a received routing message may be interpreted into two different ways, for the establishment of a new Gw (if the received message comes from a different cluster), or for the knowledge propagation of existing Gws within the cluster. Algorithm 4 represents these procedures.

Upon receiving a routing message, the information regarding the existing clusters is conveniently processed either by creating a new Gw entry, if a node received a message from a different cluster, or by adding new information about a node's own cluster. This pre-processing is required for getting all the necessary details about the network topology, aggregating the information according to the Deferred Routing hierarchy, being able to choose the appropriate Gws to reach a destination. Afterwards, in the case of a routing message coming from a different cluster, it must be discarded so that the link-state protocol is unaware of such message, avoiding unnecessary processing, otherwise it is still considered by the routing protocol for intra-cluster routing.

Additional procedures are used to maintain all the received information for a limited period of time (exp_{time}), guaranteeing that no out-of-date information is used for routing data packets. These procedures consist on removing old entries if they are not updated within a predefined amount of time, while keeping an ordered list of the existing Gws for each cluster. Moreover each generated Gw entry has a sequence number (seq_{num}) and the ID of the originator of such entry (*originator*), which is responsible for incrementing the *age* of the Gw for each update it sends, in order to keep track of duplicate or out-of-order information as well as knowing which Gws are more reliable.

Algorithm 5: Inter-Cluster Routing

```
procedure Packet_Arrival_Handle(void)
  ...
  Get_Packet_Information
    // Look up the destination's Cluster ID
    if CLUSTER_ID_LOOKUP(dest_addr) = −1 then // Cluster not found!
      return
    else
      if (next_hop ← NEXT_HOP_FINDER(dest_addr)) = −1 then
        // Next hop not found!
      return
    end if
  end if    SEND_PACKET(next_hop)
end procedure

procedure NEXT_HOP_FINDER(dest_addr)
      // Get the destination's Cluster ID
      if(cluster_id ← Cluster_Id_Lookup(dest_addr)) = −1 then
    // Cluster not found!
  return
else
    if cluster_id = own_cluster then
        // Link-State Routing failure detected!
    return −1
  end if
end if
// Find the most suitable next hop
if gw_entry ← Hash_Table_Get(gw_table, cluster_id) then
      // Get the next hop information from an ordered list
      // (the first occurrence will be the best entry)
      next_hop_tuple ← List_Access(gw_entry.gateway_table, 0)
      number_hops ← next_hop_tuple.hop_count
      next_hop ← next_hop_tuple.gateway
    end if
if number_hops ≠ 1
    // This node is not a GW, or it may not be the best within
  its cluster
    // Check if any other node has better connectivity
    if gw_entry ← Hash_Table_Get(connectivity_table, cluster_id) then
```

* (*continued*)

Algorithm 5: Inter-Cluster Routing

// Get the next hop information from an ordered list
// (the first occurrence will be the best entry)
next_hop_tuple ← *List_Access*(gw_{entry}.*gateway_table*, 0)
if *next_hop_tuple.hop_count* < $number_{hops}$ **then**
$number_{hops}$ ← *next_hop_tuple.hop_count*
$next_{hop}$ ← *next_hop_tuple.gateway*
end if
end if
end if
return $next_{hop}$
end procedure

procedure CLUSTER_ID_LOOKUP($dest_{addr}$)
if
$mapping_{entry}$ ← *Hash_Table_Get*($ip\_cluster\_mapping_{table}$, $dest_{addr}$)
then
return $mapping_{entry}$.$cluster_{id}$
else
return −1
end if
end procedure

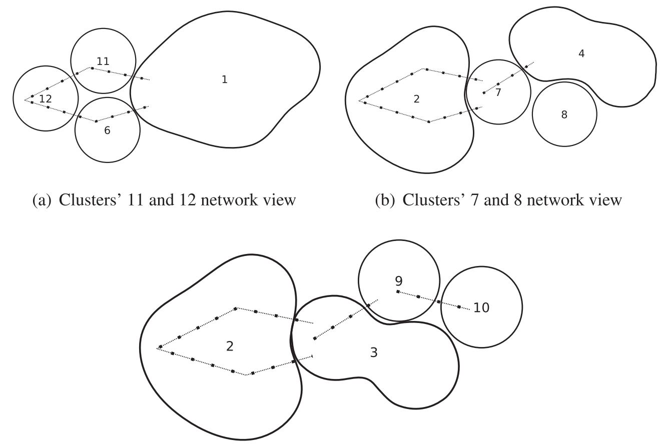
2.2.2. Routing data traffic

Having the necessary knowledge about the existing clusters and how they can be reached through the available Gws, it is still necessary to correctly route each traffic packet. Whenever the destination node is within the same cluster as the source node, the link-state routing protocol should be able to correctly route any packet. However, if source and destination are in different clusters, the gathered cluster information will have to be used by extra Deferred Routing procedures, responsible for inter-cluster routing, ensuring that the packets are able to reach their destination. Since the proposed routing paradigm is different from typical link-state or distance-vector protocols, an end-to-end path is not established. The packets are rather forwarded to the most suitable Gw capable of reaching the desired cluster, using the Deferred Routing Hierarchy to get closer, cluster after cluster.

It is assumed that first, the intra-cluster routing protocol tries to handle traffic packets, however, when it fails to do so, the inter-cluster routing is used, processing the received packets as shown in Algorithm 5. This routing procedure consists first of checking if the node is a Gw to the desired cluster and then, if so, check if there is a more suitable one within its own cluster. If the node being considered is already the best Gw, it simply sends the packet to the node in its neighbour cluster. Otherwise it queries the link-state routing protocol to know how to reach the chosen Gw and forwards the packet to the next hop in the routing table which, in its turn, does the same until the Gw is reached.

2.2.3. Deferred Routing example

Taking into account the concept of Deferred Routing, the presented algorithms and Fig. 1, a node in cluster 12 shall have in its routing table connectivity to clusters 11, 6 and 1. Considering how routing messages are processed, this network perspective resulted from routing messages received from clusters 11 and 6 as shown in Fig. 2(a). Both these clusters have connectivity with cluster 7 which, due to the View Determination Algorithm is perceived as cluster 1. Even though neither cluster 11 nor cluster 6 explicitly sent any routing messages to 12 and despite cluster information included in routing messages concerns only the cluster in which they were created, by overhearing these messages, cluster 12 is able to create its own network perspective and announce connectivity with cluster 1, adding an extra cluster hop.



(a) Clusters' 11 and 12 network view (b) Clusters' 7 and 8 network view

(c) Clusters' 9 and 10 network view

Fig. 2. Deferred routing example.

Having the required routing information, cluster 12 is now able to send data traffic to any cluster. For instance, if a data flow is to be created with a node in cluster 10, the first worthy aspect is that this node's IP address will be only known as being in cluster 1. Thus, after identifying the destination cluster, the sender node will choose a gateway in its cluster with reachability to the desired cluster. Since both cluster 11 and 6 have the same number of cluster hops any of them might be chosen depending on a Gateway Metric later defined in this work. When one of these clusters is reached, the same procedure is repeated. A Gw node capable of reaching cluster 1 is chosen and the message is forwarded to it, using the OLSR routing table within the cluster. After reaching the cluster 7, which in this scenario is the only possible cluster to be reached, the destination of the data flow is no longer cluster 1 but cluster 4 instead, as depicted in Fig. 2(b). At this point, cluster 8 is directly connected to cluster 10 and could be an option. However, neither cluster 7 or 8 perceive cluster 10 and thus, the data messages are automatically forwarded to cluster 9. Such decision is straightforward since at cluster 7, direct connectivity exists to cluster 4 and sending to cluster 8 would have an extra cluster hop. Once the data flow has been received by cluster 9, as shown in Fig. 2(c), being cluster 9 a brother sibling to cluster 10, all data messages are forwarded to a Gw node with connectivity with cluster 10.

2.3. A gateway metric for Deferred Routing

In the presented protocol, gateway nodes "overhear" their neighbours' routing information and consider themselves indirect gateways to clusters which they are not neighbours with, increasing their Cluster Hop Count connectivity by one. This approach allows nodes to choose the appropriate nodes to forward data to other clusters, but requires a robust scheme to guarantee that this choice is not ambiguous between nodes within the same cluster. Such aspect is important as the propagation of Gw information in large scale networks is subject to delays and lost routing packets, leading to routing inconsistencies and poor reliability.

A previous work, presented in [18], defines a new routing metric which adds to the number of cluster hops extra relevant information. On Deferred Routing, the choice of an appropriate Gw throughout different clusters should consider using the most stable and reliable gateway. These two aspects can easily be obtained from the stored gateway *age* and expiry time information. The *age* of a gateway is a property that reflects how stable a node is as gateway, being more stable for higher ages. Moreover, it is important to be aware of how valid the existing information is, since when a node receives information about a gateway, it may be about to expire or it might just have been sent.

Taking into account the number of "cluster hops", a gateway's *age* and the validity of the existing information, which may be more or less up-to-date, a suitable metric for reliable routing may be derived, allowing robust routing in large scale networks. However, in addition to the three defined parameters, it is also important to understand what they represent and how they can be used simultaneously. Representing adequately the difference between possible values for the number of cluster hops is important and requires a mapping to an appropriate function. As cluster hops are being considered, the difference between 1 and 2 hops is significant, however with higher hop values, the difference of 1 cluster hop may not be relevant as both are already undesirable. This characteristic can be represented by a sigmoid function with a predefined threshold hop_{th} number of hops, according to the network's number of hops. The hop parcel $h(x)$ of the metric is defined in Eq. (1).

$$h(x) = \frac{1}{1 + e^{hop_{th} - x}} \quad (1)$$

A gateway stability may be represented by its *age*, however, when two gateways are compared, the difference between their ages must be correctly understood, similarly to the number of cluster hops. Depending on the number of refreshes/updates that a Gw receives from a different cluster, its *age* is higher or lower. As a result, a "younger" gateway is less stable than an "older" one. However the difference between their ages is not so significant when both have already been stable for a long period of time, such that two gateways with higher ages have similar stability factors. The metric's *age* parcel $g(y)$, presented in Eq. (2), mimics this behaviour. A routing protocol which also considers stability is presented in [19], where route stability is more important than the path hop count. Despite using different metrics for link stability, this protocol still suffers from typical on-demand protocol disadvantages such as flooding and path retrieval delays, not being suitable for large-scale networks.

$$g(y) = \frac{1}{\sqrt{y}} \quad (2)$$

Routing information is typically maintained only for a limited amount of time, relying on updates to this information, so that it does not expire. Usually, the most recent information should reflect the correct network perspective. In spite of being recent, due to network delays, newly created information may not have been delivered throughout the entire network, creating incoherent views of the network. In order to avoid this, information "validity" should take into account its expiration time. To achieve this behaviour, the expiration time should be modelled into a function $v(z)$, such that the threshold $validity_{th}$ represents the most valid information, considering a maximum expiration time of MAX_{expiry}, as shown in Eq. (3).

$$v(z) = \frac{((MAX_{expiry} - validity_{th}) - z)^2}{(MAX_{expiry} - validity_{th})^2} \wedge validity_{th} < MAX_{expiry} \quad (3)$$

By joining these three parameters into a weighted function, each one mapped to its own function, a metric capable of providing consistent views of routing tables in large scale networks is achieved – even when using Deferred Routing, which only maintains limited information. This metric is presented in Eq. (4). The maximum value for the metric will be 1, representing the worst possible value for a gateway.

$$m(x,y,z) = w_{hop} \times h(x) + w_{age} \times g(y) + w_{val} \times v(z) \quad (4)$$

Furthermore, the metric can be adjusted to specific networks by changing $w_{hop}, w_{age}, w_{val}$ weights, as well as by tweaking the existing thresholds, tuning the results according to the existing scenarios, such that:

$$\sum_{i \in hop,age,val}^{i} w_i = 1 \wedge \forall x,y,z \in \mathbb{R} : m(x,y,z) \leqslant 1 \quad (5)$$

In the following section a methodology for the performance evaluation of Deferred Routing is presented, defining two different scenarios and appropriate evaluation metrics.

3. Performance analysis methodology

Wireless Ad-hoc networks have become interesting for future networks due to their unique characteristics, such as being infrastructure-less, mobile and self-X. As a result, a myriad of possible scenarios has been proposed for these networks, from rescue operations to social events.

3.1. Objectives

Having defined the Deferred Routing concept as a new routing approach for scalable Ad-hoc networks, a thorough evaluation of its performance is required. When evaluating a new routing approach it is important to consider different aspects:

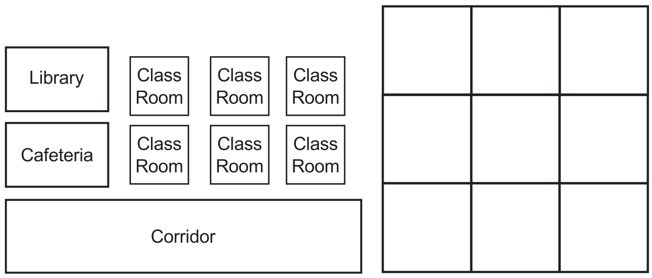
- identify the context in which the concept is inserted;
- determine if the concept solves the issues to which it is proposed;
- ensure that the concept does not raise new issues itself.

Taking these 3 different aspects into consideration, the Deferred Routing performance assessment must involve the evaluation of a large scale network with dynamic characteristics, determining the concept's ability to handle mobility phenomenons and increased routing information. Moreover, it is important to measure the overhead introduced by this concept and its performance, in different conditions.

3.2. Simulation conditions

In order to evaluate the performance of the presented Deferred Routing paradigm (DefeR in the presented figures), two scenarios incorporating different characteristics have been used. One of these scenarios, from now on known as *Scenario 1*, has 3 distinct mobility behaviours following the Random Waypoint Mobility Model and another node setting without mobility. This mobility model has been extensively used in literature and while some works have shown some disadvantages in using it [20], it is still widely used in recent works [21,22], as it provides a generic form of mobility without being tied to particular applications. Moreover, the random waypoint implemented in OPNET Modeler Wireless Simulator [23] guarantees a uniform distribution of the x and y coordinates within the boundaries of the scenario, as well as different initial states (Pause or Moving) for each node, ensuring a "steady-state" distribution of the Random Waypoint Model [24]. Different node densities per cluster, as well as different traffic flows throughout the simulation time, were defined, as presented in the scenario's description. The other used scenario, *Scenario 2*, is a simple square area, with equal cluster densities. Static and mobile versions of the scenario were simulated, using only one type of traffic flows. These scenarios are both depicted in Fig. 3.

The simulations of both scenarios were performed using several protocols in order to provide a thorough evaluation. In addition to the Deferred Routing Scheme, the C-OLSR protocol [25], the AODV



(a) Scenario 1 (b) Scenario 2

Fig. 3. Defined scenarios representation.

reactive approach and the OLSR protocol were also evaluated. Since the C-OLSR protocol presents 3 different approaches for routing, the distributed version of this protocol was used as it avoids bottlenecks from using Clusterheads in the clusterhead-based and hybrid approaches.

The described scenarios were simulated using the OPNET simulator, with a total of 30 runs per scenario, always using different seed values and the Linear-Congruential Random Number Generator Algorithm, for a total simulated time of 15 min (900 s). The considered wireless nodes follow the IEEE 802.11 g standard [26], and have a maximum range of 25 meters (Transmit Power of $2.27e^{-5}$ W) which should correspond to a realistic range of common wireless cards [27,28]. However, due to the accurate radio model implemented by default in the OPNET Simulator, asymmetric links or even unidirectional links may occur, as well as channel errors and multi-path interferences, among others.

Regarding the Deferred Routing specific parameters, the hop, age and validity weights, used for the presented metric, were 0.6, 0.2 and 0.2, respectively, according to the results obtained in a previous work [18]. All other simulation parameters not mentioned here use their values set by default in the OPNET Modeler Wireless Suite Simulator, version 16.0.A PL1.

3.2.1. Scenario 1 description

The first chosen scenario is intended to be dynamic and it has been inspired in a University Department, representing one floor where different clusters exist due to different rooms such as class rooms, a library, a cafeteria and a big corridor connecting all the rooms. All the clusters have 49 nodes which is the best number of nodes handled by OLSR [29], and their specifications are described in Table 1, where the speed and pause time intervals follow a uniform distribution. There are 6 Class Rooms with the same characteristics, summing up, with the other rooms, to a total of 9 clusters. Traffic flows are the same for all the rooms, with 6 source nodes which send data to nodes in their own cluster, and an additional source node which randomly chooses a destination node in the network, exploring the sense of community [16]. Every source node creates 3 different flows during the simulation, representing simple file transfers, interactive gaming and information exchange [51], as described in Table 2. All the speed and pause values presented within an interval are randomly chosen following a uniform distribution.

Table 1
Clusters' description.

Room type	Area	Mobility characteristic	
		Speed (m s^{-1})	Pause time (s)
Class room	20 × 20	Static	Static
Cafeteria	22 × 35	0.2–1.0	60–120
Library	22 × 35	0.2–1.0	180–600
Corridor	25 × 120	0.2–1.0	10–60

Table 2
Flows' characteristics.

Flow	Simulation time	Packet size	Inter-arrival time
1 (6×)	Start:160;stop:280 s	4 kb	2–6 s
2 (6×)	Start:280;stop:520 s	4 kb	2–6 s
3 (6×)	Start:520;stop:760 s	Exponential (1 kb)	Exponential (1)

3.2.2. Scenario 2 description

In order to provide a thorough analysis of the proposed routing approach, a different scenario was also simulated, following a typical organization for wireless nodes which can be found in several works (e.g. [30]), where nodes are deployed in a square area. This scenario has a square area of 3 by 3 km, with 9 clusters of 49 nodes each with an area of 1 km by 1 km, using the same flow characteristics as the previously presented scenario except that all the flow destinations are randomly chosen. Static and a mobile versions of this scenario were defined, where all the nodes are either static or mobile following the Random Waypoint Mobility Model, with a pause time defined by a uniform distribution between 60 and 120 s. The mobile versions were defined with speeds within a range between 0.2 and 1.0 m/s, as well as within a range between 8 and 9 m/s in a different simulation set. These speeds were defined by a uniform distribution, being randomly chosen, corresponding to the plausible velocity of a person walking (≈ 0.8 m/s) and to the speed of vehicles in an urban scenario (≈ 8 m/s) [31], which varies depending on the density of vehicles according to existing traces in literature [32].

3.3. Evaluation metrics

Having defined possible scenarios for the evaluation of the proposed routing approach, it is important to choose the appropriate evaluation metrics to be used. One of these metrics is the path length (hop count), from source to destination, which typically is minimized by routing protocols, contrary to the Deferred Routing. The average percentage of losses and end-to-end delay also reflect a protocol's ability to choose suitable paths and should be taken into account. In addition to these metrics, it is also important to measure the required resources and, therefore, routing traffic overheads.

The topology awareness of a routing protocol is a metric representative of a routing protocol's stability and knowledge about the network's structure, registering topology changes during the simulation. A topology change occurs whenever a new *TC* or a *TC* with a higher sequence number is received and also when a *TC* entry is deleted after expiry. Each topology change triggers a routing table recalculation, however in order to reduce computational overhead, the routing table is only recalculated by default at most every 1 s, processing all the received topology changes between each recalculation. Such technique is compliant with the OLSR specification and used in existing implementations [33,34]. Moreover, all the analysed protocols use this improvement in order ensure a fair comparison between them.

The amount of processed topology changes in routing table calculation reflects a protocol's stability and will also be analysed, referred as Average Topology Changes per Routing Table calculation (AToCRT) and defined by Eq. (6).

$$AToCRT = \frac{\text{Number of Topology Changes}}{\text{Number of Routing Table Calculations}} \quad (6)$$

The number of routing table calculations possible in a 900 s simulation is defined in Eq. (7), with i being the simulation instant where n Topology Changes occur. Since the number of topology changes is influenced by the mobility of nodes, the different speeds used in an evaluation will be reflected in the AToCRT metric and also on the total number of routing table calculations. In particular, with higher speeds, an increased number of Topology Changes throughout the time will trigger a higher number of routing table calculations, with a maximum of 1 per second, as defined by $f(n)$.

$$\text{Routing Table Calculations} = \sum_{i=1}^{900} f(TopologyChanges_i),$$
$$f(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n > 0 \end{cases} \quad (7)$$

Additionally, since the OLSR protocol is used in the presented implementation of the Deferred Routing and C-OLSR protocols, the number of Multipoint Relay Nodes and of *TC* messages will also be discussed in the evaluation. Since the AODV protocol is a reactive protocol, the aspects related with topology awareness can not be considered, but the overall routing performance will still be evaluated.

4. Simulation results

For the sake of comparison, every scenario was simulated using the standard OLSR protocol, and the Deferred Routing scheme with OLSR as link-state protocol, similarly to the work entitled "Deferred Aggregated routing for Scalable Ad-hoc networks (DASH)" presented in [35]. Moreover, all the mobile nodes are kept within their clusters, even in the scenarios with the AODV and OLSR versions, allowing a fair comparison between all of them.

The following results have a 95% confidence interval obtained from the central limit theorem which states that, regardless of a random variable's actual distribution, as the number of samples (i.e. runs) grows large, the random variable has a distribution that approaches that of a Normal random variable of mean m, corresponding to the same mean as the random variable itself.

4.1. Scenario 1 results

In this scenario, in addition to the C-OLSR and AODV protocol, a comparison between the Deferred Routing approach with two versions of the OLSR protocol is also presented. These two versions were used in order to better understand the OLSR protocol's high memory and simulation time requirements. Comparing with Deferred Routing, the OLSR protocol consumes much more memory (8 times more) for the same scenario, and for that reason a Limited Memory version of this protocol, using the same amount of memory as the Deferred Routing approach, was used to present the performance comparisons. These results are more significant, in particular if one considers wireless mobile networks mainly composed of processor and memory limited wireless devices such as cellphones and personal digital assistants (PDAs) [36].

The performed memory limitation, Limited OLSR, consists in ignoring memory allocations performed by the OLSR protocol whenever the maximum limit is reached. This procedure was necessary not only to allow a better comparison but also in order to provide a useful comprehension of how the OLSR protocol performs in memory constrained devices.

4.1.1. Path length

The total hop count is many times used as a routing metric by several protocols in order evaluate their performance. Even though the Deferred Routing approach does not really concern the number of node hops, but essentially the number of cluster hops, Fig. 4, which presents the average number of node hops, reveals that it has a better performance than the standard AODV protocol which aims at minimizing the number of node hops. Moreover, when comparing the obtained performance with the OLSR and C-OLSR protocols, similar results are achieved.

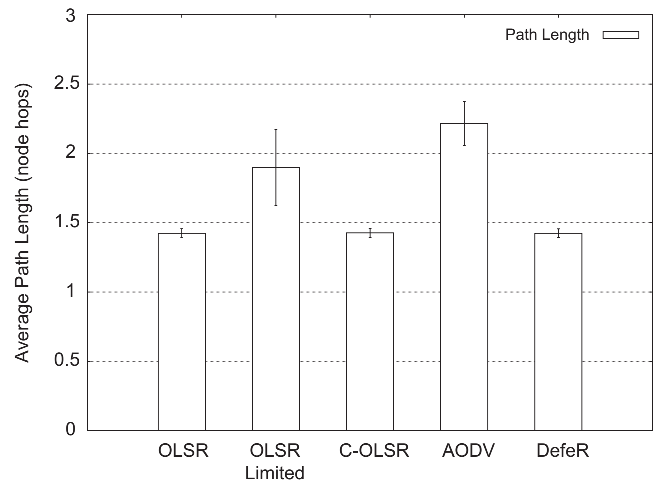


Fig. 4. Average path length (scenario 1).

The standard implementation of the OLSR protocol has obtained an average path length of 1.42 hops, similarly to the other protocols. However, the Limited OLSR version had a slightly worse performance, with an average of 1.90 hops which is better than the path obtained by AODV but worse than the remaining protocols. These higher path lengths may be due to less stable routing tables and due to the on-demand procedure used by the AODV protocol which return the fastest path available and not necessarily the shortest.

4.1.2. Average losses and delay

In routing performance evaluation the percentage of registered losses (lost bits) and average end-to-end delay are important aspects to take into account. In Fig. 5 the number percentage of losses is represented in the left axis and the delay, in a base-10 logarithmic scale, is presented in the right vertical axis, showing once more that the Deferred Routing approach outperforms the OLSR, C-OLSR and AODV protocols. Both Deferred Routing and C-OLSR have a

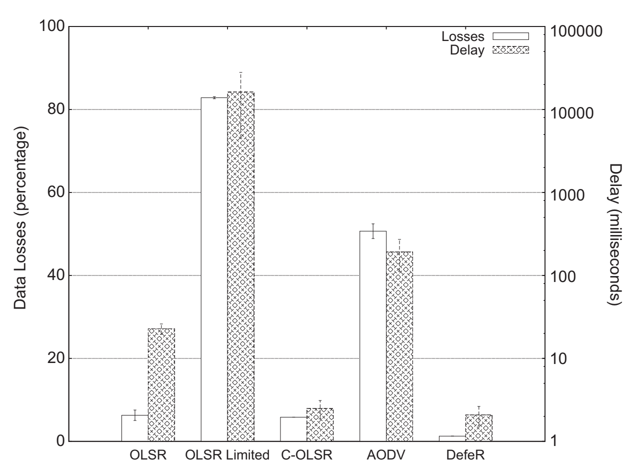


Fig. 5. Average losses and delay (scenario 1).

small delay, of 2.2 and 2.5 ms respectively. However, the C-OLSR has nearly 5 times more losses while the AODV protocol, has 40 times more losses in addition to a higher delay of 191 ms.

Since the AODV protocol has an initial Route Discovery Time, it is even less competitive as it takes, in average, 212 ms to complete this route discovery process, with total initial delay of 403 ms.

Analysing again the performance of the "resource demanding OLSR", it has a behaviour similar to the C-OLSR protocol regarding both losses and delay. However, these results stand out from the memory limited version, which has a significant amount of losses (83%) and an average end-to-end delay of 16 s, performing much worse than the AODV protocol. These losses and delay result from the memory limitation imposed which has incomplete routing paths as the OLSR requires a large amount of memory and processing time to determine the shortest path available considering all the 441 nodes.

4.1.3. Routing messages overhead

The most important aspects considering large scale routing are typically related with the amount of routing traffic or routing overhead, exchanged between the network nodes. This aspect depends mainly on the routing protocol and it becomes clear that the Deferred Routing approach has a smaller overhead. Depicted in Fig. 6, the presented scheme's sent and received routing traffic is only 15% of the corresponding total routing traffic by the C-OLSR protocol.

On the routing traffic aspect, reactive routing protocols are typically more efficient as they do not periodically send routing messages. However, despite the good performance of the AODV protocol, the Deferred Routing still generates less routing traffic.

Comparing with the typical OLSR protocol, only the Deferred Routing and AODV protocols perform better as they either use clusters or are reactive. However, the C-OLSR protocol also uses clusters but has a higher amount of received routing traffic. This is due to inter-cluster routing information needed by the routing protocol.

The memory limited version of the OLSR protocol has less received routing traffic than the normal OLSR and C-OLSR protocols, even though it sends more routing traffic as it has less stable routing tables and Topology Control messages are not properly forwarded.

4.1.4. Topology awareness

Fig. 7, presented in a base-10 logarithmic scale, shows the average number of network changes, neighbour additions and deletions, and also two hop additions and deletions. All these parameters reflect the stability and topology awareness of a routing protocol, and therefore, its reliability. As these parameters concern OLSR specific operations, the AODV protocol is not considered.

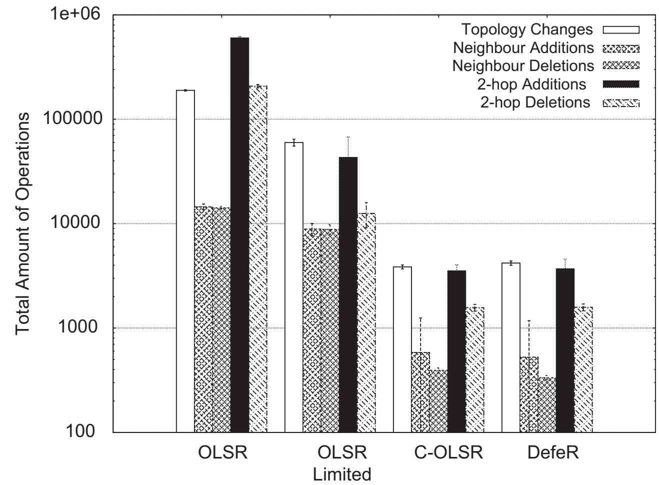


Fig. 7. Average number of network changes (scenario 1).

Regarding topology awareness, the Deferred Routing and the C-OLSR protocols have a similar behaviour as both consider the same clusters. Moreover taking into account that both protocols use OLSR within their clusters, the obtained results validate the Deferred Routing approach implementation.

The maximum number of topology changes processed in each routing table calculation in a base-10 is presented in Fig. 8, revealing that the routing problems found with the limited version of OLSR are due to an insufficient number of observed topology changes. On the other hand the OLSR protocol reveals its instability by processing a large number of topology changes, increasing the complexity of each routing table calculation. The clustered version of the OLSR protocol has almost half of the accumulated topology changes registered by Deferred Routing, however, its inferior routing performance suggests that it might be missing important route updates.

The Deferred Routing approach reveals that it is able to achieve much better results than the standard version of the OLSR protocol, having for example, 45 times less topology changes. This significant difference is clearly an improvement towards scalable routing

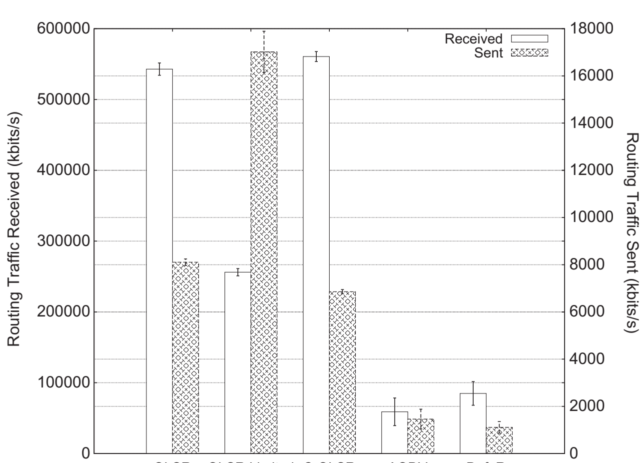


Fig. 6. Routing messages overhead (scenario 1).

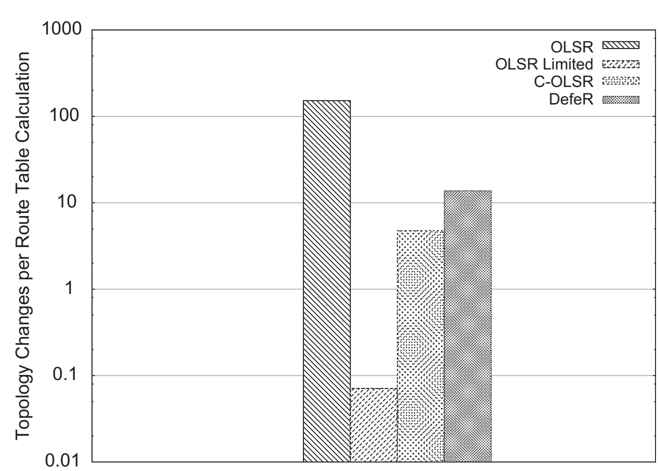


Fig. 8. Topology changes per routing table calculation (scenario 1).

using OLSR as it requires less routing table changes, while being more lightweight.

4.1.5. Average MPR count and sent TCs

Within the family of link-state routing protocols, OLSR stands out for the use of Multipoint Relay (*MPR*) nodes in order to efficiently propagate routing information between the network nodes, without creating unnecessary overheads. The AODV protocol is not considered as it does not have such messages or special nodes.

Fig. 9 shows how Deferred Routing is able of sending almost 5 time less Topology Control messages than the C-OLSR protocol, even though both have a similar number of *MPR* nodes. This is mainly due to the fact that the C-OLSR requires additional cluster *TC* messages in order to propagate its routing information.

Returning to the OLSR protocol, Deferred Routing is capable of enhancing a protocol's own characteristics, requiring only 14% *MPR* nodes of those required by the OLSR protocol, and sending 7 times less Topology Control (*TC*) messages.

4.2. Scenario 2 results

This scenario avoids biased results due to the usage of a specific setting with particular characteristics. Typical wireless multi-hop evaluations consider simple scenarios where all the nodes have similar characteristics as used, for instance in [37,38] or [39]. According to this, further simulations were performed, using a fully static and fully mobile scenarios with different speeds, comparing the Deferred Routing approach with the OLSR, C-OLSR and AODV protocols. The memory limited OLSR version, in this scenario, has the same performance as the standard performance of the OLSR protocol since it is a simpler scenario and no high memory consumptions were registered.

4.2.1. Path length

The obtained path length (i.e. average number of node hops) for this scenario, presented in Fig. 10, reflects the spirit behind the Deferred Routing Scheme where the main target, as in similar works [12,40], is scalable routing without necessarily finding the path with a smaller number of hops. Particularly, for the static scenario, the proposed approach has a higher path length when compared with the remaining protocols.

In the mobile scenario at a lower speed, Deferred Routing presents a smaller path length than the AODV protocol but higher than OLSR and C-OLSR. Despite this fact, the Deferred Routing scheme has less losses overall in both scenarios, as depicted by

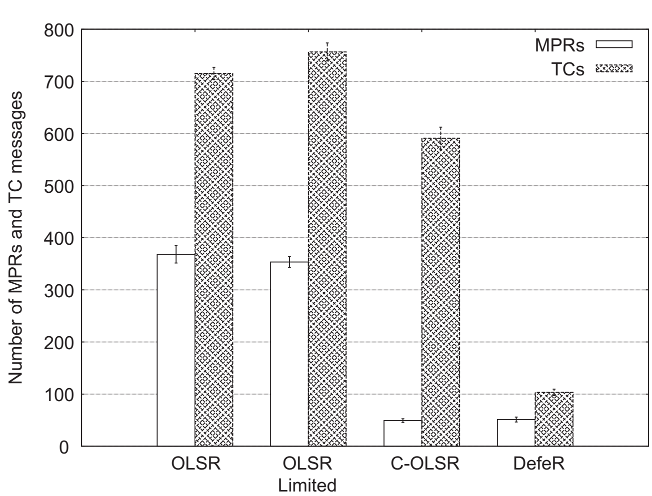


Fig. 9. Average number of *MPR* nodes and *TCs* (scenario 1).

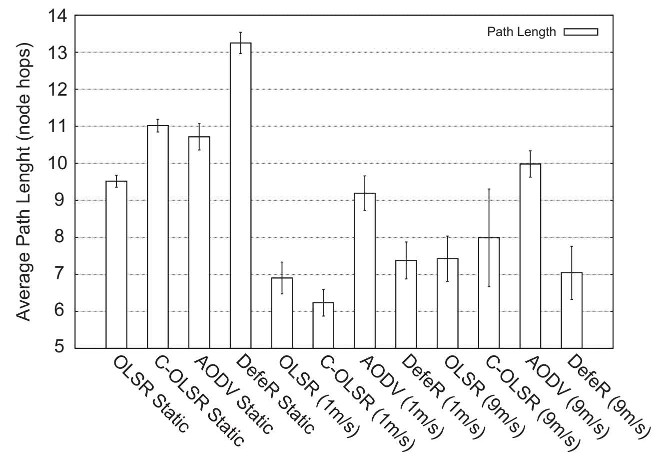


Fig. 10. Average path length (scenario 2).

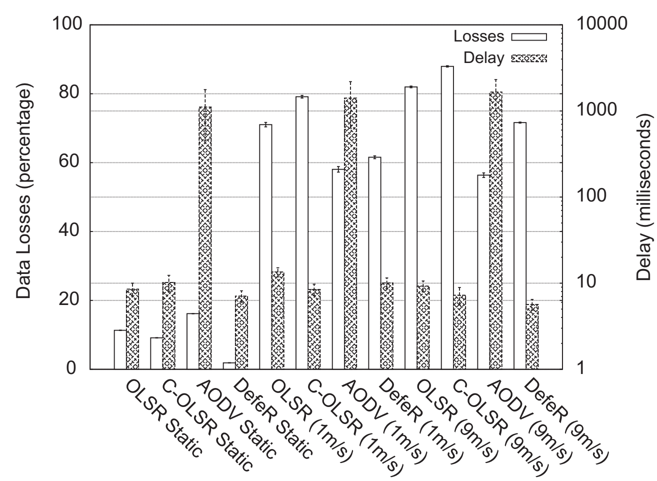


Fig. 11. Average losses and delay (scenario 2).

Fig. 11. Moreover, for a higher speed, the Deferred Routing protocol has a smaller hop count than all the other presented protocols.

4.2.2. Average losses and delay

Despite typically having a higher path length, the percentage of registered losses by the Deferred Routing scheme is, for the static scenario better than any of the presented protocols, as shown in Fig. 11. With less than 2% of losses the Deferred Routing outperforms the OLSR, C-OLSR and AODV protocols which register more than 9% of losses, up to 16%. Moreover, having a delay of 7 ms, presented in a base-10 logarithmic scale, it is also better than the other protocols in which the AODV protocol stands out for having a delay higher than 1 s.

For both mobile scenarios, the AODV protocol had the best traffic delivery performance but a great amount of end-to-end delay, taking sometimes more than 2 s. In addition to this, the Route Discovery Time of the AODV protocol was in average 1 s. The high confidence interval in AODV reveals some instability of the protocol, in particular regarding the obtained delay. Conversely, the Deferred Routing scheme has only 3% more losses than AODV at a speed of 1 m/s with much smaller delay (around 10 ms).

Still concerning the mobile versions of *Scenario 2*, the C-OLSR registered a higher number of losses when compared to the OLSR protocol. This may be due to a slow update of gateway information when nodes move from their position, which may result from the defined scheme in the distributed C-OLSR, where C-TC messages have to be propagated between all the nodes in the existing clusters [25].

4.2.3. Routing message overhead

In Fig. 12, the amount of sent and received routing traffic is presented. In the static scenario the Deferred Routing registers the lowest routing overhead. These results are even more significant when comparing with the AODV protocol which, being an on-demand protocol should have low routing traffic, but yet it is higher and presents a large confidence interval. This reflects the flooding mechanisms that take place in the AODV protocol during the Route Discovery Process.

Taking into account the scenario with lower mobility, the Deferred Routing scheme presents a slightly higher overhead than the AODV protocol. Even though this indicates that the AODV does in fact perform better in a moderate mobile scenario, it also suffers from higher route discovery and end-to-end delays when comparing with Deferred Routing. Moreover, not only does the Deferred Routing solution clearly outperform the proactive protocols, but it also outperforms the AODV protocol in the scenario with higher mobility.

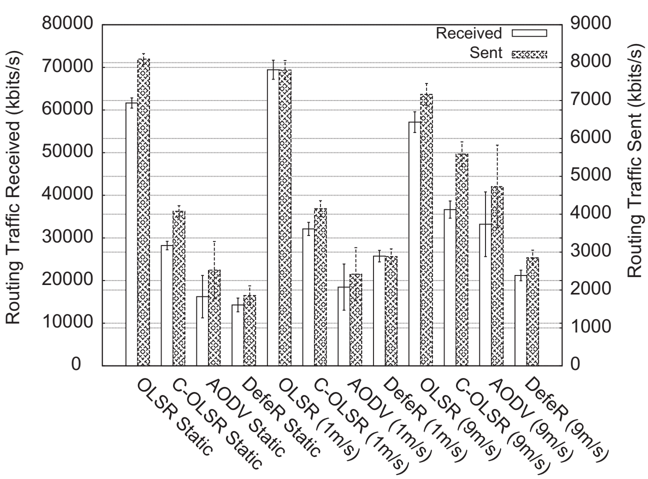


Fig. 12. Routing messages overhead (scenario 2).

4.2.4. Topology awareness

Fig. 13, presented in a base-10 logarithmic scale, shows that all the obtained results for the average number of network changes, neighbour additions and deletions, and also two hop additions and deletions, are much smaller in Deferred Routing, for both static and mobile scenarios, than the ones registered for the OLSR protocol. In fact, the number of topology changes is up to 10 times smaller, showing that Deferred Routing renders a much more accurate and stable network perspective. Moreover, apart from topology changes in the static scenario, when comparing the Deferred Routing protocol with the C-OLSR approach very little differences are noticed. The main difference between these two protocols is only registered in the number of topology changes which is lower for the C-OLSR protocol but which also has a larger confidence interval, revealing more instability.

Regarding the different speeds in the mobile versions of this scenario, the number of topology changes in C-OLSR and Deferred Routing seems similar due to used logarithmic scale, being the existing differences nearly imperceptible. In fact, the number of registered topology changes, contrary to what would be expected, is slightly lower at a higher speed. This is a consequence of how the periodic updates are used by OLSR, which does not cope well with mobility and fails to acknowledge topology changes in a timely manner. Even though a higher mobility should trigger more topology changes due to largest distances covered by nodes, it also creates more instability in the network, such that the OLSR protocol does not acknowledge topology changes. However, the total number of neighbour additions and deletions increases with speed, confirming that OLSR is not able to handle and deal with mobility efficiently, even though new links are detected and removed.

Fig. 14, presented in a base-10 logarithmic scale, also reflects the instability of the OLSR protocol generated by the increase of nodes' speed. In fact, both the OSLR and C-OLSR protocols register an increase of the number of processed topology changes per routing table calculation. This increase results from the new routes created by the moving nodes, however the Deferred Routing approach is not as susceptible to node mobility, maintaining a similar value of AToCRT at different speeds. Moreover, as previously shown, the DefeR Routing proposal has an increased traffic delivery, confirming its better route handling.

4.2.5. Average MPR Count and Sent TCs

As stated before, the *MPR* nodes used by the OLSR protocol avoid unnecessary Topology Control messages. In Fig. 15, it is possible to verify that Deferred Routing has similar number of *MPR*

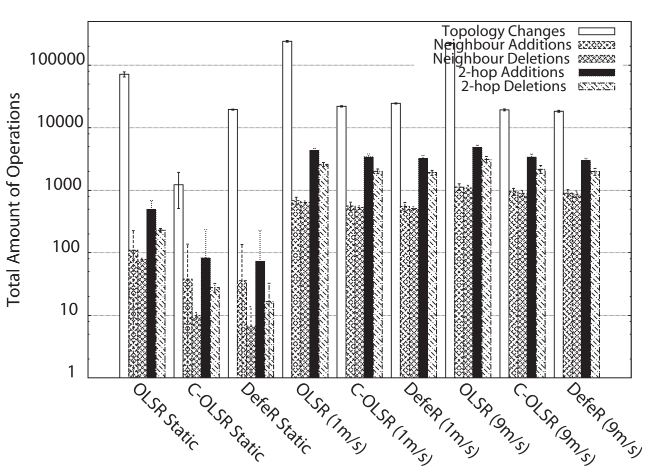


Fig. 13. Average number of network changes (scenario 2).

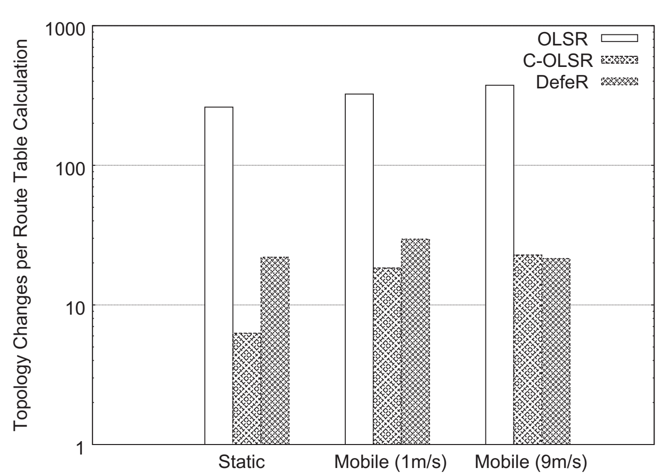


Fig. 14. Topology changes per routing table calculation (scenario 2).

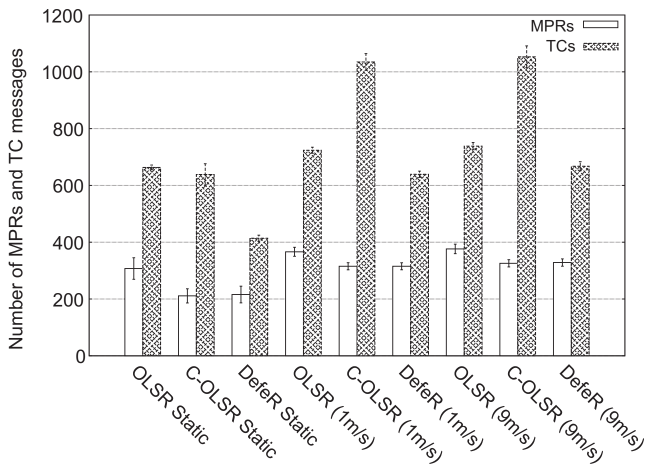


Fig. 15. Average number of *MPR* nodes and *TCs* (scenario 2).

nodes when compared with the C-OLSR protocol. However, it requires less *TC* messages, avoiding an additional routing overhead.

Comparing with OLSR, the Deferred Routing approach is capable of reducing the number of required *MPR* nodes considerably and, as a direct result, the number of sent *TC* messages is also smaller. On the other hand, for both mobile versions of the scenario, the C-OLSR has a higher transmission of *TC* messages when compared with the OLSR protocol, due to the transmission of C-TC messages, being only slightly affected by the increase of the speed of nodes.

In particular, with the increase of mobility, the connectivity amongst nodes decreases and consequently the number of *MPR* nodes increases. This leads to a higher number of sent *TCs*, revealing once again that the OLSR protocol has issues with mobility, being less scalable. However, due to the Deferred Routing hierarchical organization, making use of different granularity levels for the existing clusters, the *MPR* election algorithm is more efficient, and this increase of *TC* messages is less significant.

4.3. Summary of analysis

The proposed routing approach has been evaluated in two different scenarios, revealing that it is able to efficiently reduce the overall routing traffic required to manage a large scale network. This was achieved by using the hierarchical aggregated views of the network, increasing the reliability of the existing routing information and avoiding unnecessary network disruptions.

The Deferred Routing protocol presented a higher hop count in some scenarios when compared with other alternatives. However, the obtained results show that it is more stable and has less routing overhead, presenting better performance regarding traffic delivery and delay when compared with other protocols.

Another important conclusion obtained from the provided results is that a memory limited version of the OLSR protocol, in order to correctly operate in most of wireless capable devices, is not capable of providing an acceptable routing performance.

5. Related work

The creation of infrastructure-less wireless Ad-hoc has long been an ambition of several authors, leading to the development of different types of routing protocols, with different approaches. From the well known existing routing protocols for wired networks, proactive routing protocols such as OLSR [5] or DSDV [2] have been proposed, however, these suffered from large overheads, in particular in dynamic networks, exchanging unnecessary or inaccurate routing information. A different paradigm was then attempted, changing the routing protocol from proactive to reactive, where routing information would only be transmitted when a node wants to exchange data with another. Despite avoiding the storage of inaccurate routing information, reactive protocols can generate a heavy load on the network when flooding routing packets for route retrieval, especially if there are different traffic flows. Moreover, when a node desires a routing path, the wanted route retrieval always suffers from an initial delay, which may not be suitable to some types of traffic.

Two different classes of routing protocols for MANETs have also been defined. One of these routing classes combines both the proactive and reactive routing schemes, typically in conjunction with clustering algorithms, named hybrid routing. Another routing class relies on nodes' positions, either for cluster formation, or for routing with the use of graph theories to reach the desired destinations. However these protocols, such as [41] typically require the usage of GPS devices, which may not always be available. Both hybrid and location based routing approaches are known for having routing hierarchies, even though proactive and reactive hierarchical schemes also exist.

Some hierarchical protocols, such as DASH, are entirely proactive and still have an infrastructure associated to routing. In a similar way, some Reactive Routing approaches have also used hierarchies to increase their performance. Also, hybrid routing schemes are well known for their commonly present hierarchy definitions. The most relevant ones, regarding proactive, reactive and hybrid hierarchical approaches are described next.

5.1. Hierarchical routing protocols

The definition of specific hierarchies by different routing protocols has commonly been used aiming at keeping the protocols more scalable. In contrast with typical flat routing protocols, hierarchical protocols usually exchange their routing information in different ways, according to a cluster or node hierarchy level. Well defined hierarchies are usually more common in hybrid routing protocols, however, hierarchical routing can also be found in proactive and, even though less frequently, in reactive routing protocols.

5.1.1. Hierarchical proactive protocols

The usage of hierarchies in conjunction with proactive routing approaches can be observed as a hierarchy of clusters, as an organized tree of addresses, or even as trees of paths forming a topology. Several schemes exist and all attempt to efficiently handle routing with the least overhead possible, as presented next.

STAR. The "Source-Tree Routing in Wireless Networks Protocols", STAR [40], is a link-state protocol which has on average less overhead than on-demand routing protocols. Its bandwidth efficiency is accomplished by restraining the dissemination of link-state information only to the routers in the data path towards the desired destinations. STAR also creates paths that may not be optimal while avoiding loops, such that the total available bandwidth is increased. Moreover STAR has specific mechanisms to know when update messages must be transmitted to detect new destinations, unreachable destinations, and loops.

Despite being able to scale, as each node only maintains a partial topology graph of the network, the STAR may suffer from large memory and processing overheads in scenarios where constant mobility may report different source trees, and routing paths are too big due to the network size.

MMWN. In the work entitled "Multimedia support in Mobile Wireless Networks", MMWN [42], the authors propose an architecture consisting of two main elements, corresponding to different node types, which can either be switches or endpoints. Both of these can be mobile, however only switches can route packets and only endpoints can be sources of or destinations for packets. This protocol also keeps a cluster hierarchy as a location management scheme, capable of obtaining the address of an endpoint. This information is kept as a dynamic distributed database, such that in each node there is a location manager node.

The proposed hierarchy allows the necessary amount of routing messages to be reduced, such that only location managers are required to update their information and only then perform the location finding process. However, this aspect is also negative on the overall performance of the protocol, as routing is strongly related with the hierarchy of the network, making the routing process complex and being vulnerable to disruptions when location managers change.

CGSR. Another proactive hierarchical routing protocol is the "Cluster-head Gateway Switch Routing" protocol, CGSR [43], where nodes are also grouped into clusters. This protocol relies on a cluster-head node to keep routing information about its cluster, and all other nodes only need to know the routing path until their own cluster-head. Additionally, all the inter-cluster routing is also processed by the cluster-head which connects to remaining clusters' cluster-head nodes.

Even though the proposed cluster hierarchy may reduce the amount of flooding for dissemination of routing information, as only the cluster-heads are responsible for this task, the process of maintaining these clusters involves additional overheads, in particular the election of an appropriate cluster-head node. Moreover, this special node will always represent a bottleneck on each cluster, overloading it and possibly leading to a faster energy depletion, and consequent cluster-head re-election.

C-OLSR. The work entitled "Cluster-based OLSR extensions to reduce control overhead in mobile Ad-hoc networks", C-OLSR [25], proposes an extension to the OLSR protocol by introducing a cluster organized network. The authors propose a scheme where the existing clusters are considered as nodes themselves, using the *MPR* concept created by OLSR applied to clusters. This structure, in conjunction with the definition of C-HELLO and C-TC messages, allows the maintenance of paths among the existing clusters while reducing the required amount of routing information, as only *MPR* Clusters generate C-TC messages.

Even though this paper uses the OLSR protocol for intra-cluster routing, proposing the mentioned C-HELLO and C-TC extensions to support a clustered network, the propagation of these new messages across clusters may have a negative impact. Moreover, the proposed mechanisms may suffer from mobility phenomena which, as in other approaches, require an additional overhead of updating the entire network structure.

DART. Inspired on a previously work on a Dynamic Addressing paradigm, the authors propose DART, "Dynamic Address Routing for Scalable Ad-hoc and Mesh Networks" [4], a proactive hierarchical approach that efficiently manages the organization of nodes into zones for large scale networks. Address allocation and lookup are the main drawbacks of this proposal. However the published work presents schemes to tackle these problems showing how addresses can be allocated taking into account node positioning, building a tree with l levels where l is the number of bits used in the routing address. A clear distinction is made between routing address and the identity of a node (a unique identification tag) since the routing address is dynamic and changes with node movement, contrasting with the node identifier which is always the same.

The three most important functionalities in DART are, first, the address allocation responsible for maintaining one routing address per network interface according to the movement and current position of a node; second, the routing which determines how to deliver packets from source to destination and, finally, the node lookup which consists in a distributed lookup table in charge of mapping identifiers to network addresses.

The DART proposal reveals to be an efficient solution for routing in large scale Ad-hoc networks, however for small networks the Dynamic Address Heuristic has a strong overhead impact and in general it is difficult to implement, as the distributed lookup table is hard to manage.

5.1.2. Hierarchical reactive protocols

The usage of Hierarchical Reactive Protocols is modest when compared with proactive or hybrid routing approaches. This is most likely due to the fact that most well defined hierarchies require constant updates in order to be efficiently kept, going against the concept behind Reactive Routing, which only exchanges routing information when required. Nevertheless, some Hierarchical Reactive protocols do exist and are described in the following paragraphs.

CBRP. As an attempt to create a scalable proactive routing protocol, the "Cluster Based Routing Protocol", CBRP [44], proposes a variation of the "Min-Id" [45] for cluster formation, restraining the typical flooding required by proactive protocols within each cluster. By relying on flooding between cluster-heads in different clusters, adjacent clusters can be known, and thus routing overhead reduced.

As a 2-level hierarchy, this protocol can be scalable to a certain extent, however, the typical cluster formation and cluster-head election overhead still exists. Even though node mobility does not necessarily lead to inaccurate routing table calculations, as it would happen with a proactive approach, the inherent route retrieval propagation delay may lead to temporary loops.

Hi-AODV. As the name indicates, the "Hierarchical AODV Routing Protocol", Hi-AODV [46] is a hierarchical version of the well known AODV routing protocol, using a tree based on cluster-heads for the creation of the concept of virtual nodes, which correspond to a typical cluster. The cluster-head is the only node responsible for handling control packets and managing the routing table of its own internal cluster. Having a tree composed of clusters seen as a virtual node, allows Hi-AODV to reduce the number of control packets and avoid additional overhead,

In addition to the already mentioned challenges and overheads related to the maintenance of clusters and their cluster-heads, again, it is clear that even though routing overheads can be reduced, the cluster-head will always have to be part of any routing path, leading to non-optimal paths, and additional interferences in the vicinities of cluster-heads.

5.1.3. Hierarchical hybrid protocols

Quite a few Hybrid Routing protocols for Ad-hoc networks can be found in the literature, however, despite the fact that many rely on clusters or well defined zones, not many implement a hierarchical routing scheme. The following protocols propose a hybrid routing scheme capable of retrieving inter-cluster information in a reactive approach, avoiding the necessity of restraining routing information in cluster-heads to reduce the overall overhead. However, on a downside, inter-cluster communication may be subject to route retrieval delay if no previous path has been maintained in cache.

ZHLS. The "Zone-based Hierarchical Link-State" routing protocol, ZHLS [47], is characterized by dividing the network into non-overlapping zones where two different routing paradigms are used: proactive routing within the zones and reactive between different zones. This proposal alleviates single points of failure and bottlenecks by not being dependent on cluster-head nodes and, at the same time, by maintaining a scalable hierarchy based topology.

One important assumption, and a possible limitation from this protocol is that each node knows its own position (for instance by using GPS) and consequently its zone ID which is directly mapped to the node position. With this approach packets are forwarded by specifying in their header the zone ID and node ID of their destination.

The division of the network into a number of zones depends on factors such as node mobility, network density, transmission power and propagation characteristics. The geographic awareness is much more important in this partitioning process as it facilitates it when compared to radio propagation partitioning.

In addition to the limitation of requiring some positioning system, the ZHLS protocol requires that all nodes exchange inter-zone flooding information when only gateway nodes need this routing information for calculating the shortest path between different zones. Moreover, the ZHLS is susceptible to a route retrieval delay when establishing inter-zone paths, as reactive routing is used for this purpose.

DDR. Another hierarchical hybrid routing protocol, the "Distributed Dynamic Routing" algorithm, DDR [48], for mobile Ad-hoc networks, is a tree based routing protocol which consists of six different stages where an election of the preferred neighbour is made, followed by the forest construction which creates a suitable structure for the wireless network, allowing an improved resource utilization. Afterwards intra and inter tree clustering is performed, followed by zone naming and partitioning. Zones are responsible for maintaining the protocol scalable and reducing the delay.

While DDR creates and maintains a dynamic logical structure of the wireless network, the "Hybrid Ad-hoc Routing Protocol", HARP [49] finds and maintains routing paths. The HARP protocol aims at discovering the most suitable end-to-end path from a source to a destination by using a proactive intra-zone routing approach and a reactive inter-zone scheme, by performing an on demand path discovery and by maintaining it while necessary.

Even though the DDR algorithm does not require any sort of cluster-head for cluster maintenance, the possibility of some nodes being chosen as preferred neighbours by other nodes may lead to the creation of bottlenecks as they would be required to transmit an increased amount of both routing and data packets. It is important that the choice of preferred neighbours is balanced so that the overall performance of the protocol does not get compromised. Moreover maintaining the entire logical structure of the network may be somewhat heavy, depending on how dynamic nodes may be.

Hierarchical routing is expected to improve resilience to mobility [50]. However, to the extent of our knowledge there is still no hierarchical routing protocol which aggregates cluster information with different granularity levels, such as proposed by Deferred Routing, being the least disruptive approach taken by Hybrid Hierarchical protocols which use Reactive Routing for inter-cluster paths. The presented routing concept is more effective in supporting node mobility, as the changes of cluster only affect sibling clusters in the hierarchy, not disrupting the routing tables of any other clusters, while always having inter-cluster paths available with no inherent delay. The management of clusters also represents an additional overhead, even though small [15], to the operation of the Deferred Routing proposal. However, as opposed to some of the presented solutions, despite requiring a cluster organization, no specific clustering algorithm is required, having no need for the usage of cluster-heads for reducing routing traffic, as it is already reduced by aggregated views of the network.

6. Conclusion

An innovative routing approach named Deferred Routing has been proposed for infrastructure-less wireless networks. It stands out for supporting large scale networks with a reduced routing overhead. The presented scheme is resilient to mobility phenomenons, due to the proposed hierarchical aggregation of clusters, providing network stability with the achieved topology awareness. Moreover, this scheme does not require any additional routing messages, nor does it rely on any unrealistic assumptions.

The concept presented in this work can be applied to any typical link-state routing scheme, requiring minor changes in order to handle hundreds of mobile nodes. Also, regarding the formation of clusters, there is no specific assumption on the used clustering scheme, as it is flexible enough to support any available solution.

Results obtained in two different scenarios reveal that the Deferred Routing contribution is relevant for improving the traffic performance delivery, while reducing the required amount of routing traffic. The presented routing approach was compared against the proactive OLSR protocol, the reactive AODV protocol and the cluster-based C-OLSR protocol. The results showed Deferred Routing as being more efficient regarding traffic delivery, end-to-end delay and total routing overhead.

Acknowledgements

The authors thank the reviewers for their valuable comments which have helped to significantly improve the quality of this work. This work was supported by the Portuguese National Foundation for Science and Technology (FCT) through a PhD Scholarship (SFRH/BD/43790/2008) and by the National Project MORFEU (PTDC/EEA-CRO/108348/2008). The authors thank the OPNET University Program for the licenses provided for the OPNET Modeler Wireless Suite®.

References

[1] A. Cimmino, P. Donadio, Overall requirements for global information multimedia communication village 10th strategic workshop, Wireless Personal Communications 49 (3) (2009) 311–319, http://dx.doi.org/10.1007/s11277-009-9686-3.
[2] C.E. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers, SIGCOMM Computer Communications Review 24 (4) (1994) 234–244, http://dx.doi.org/10.1145/190809.190336.
[3] C.-C. Chiang, H.-K. Wu, W. Liu, M. Gerla, Routing in clustered multihop, mobile wireless networks with fading channel, in: IEEE Singapore International Conference on Networks, SICON'97, April 16–17, Singapore, IEEE, 1997, pp. 197–211. <http://www.ics.uci.edu/atm/adhoc/paper-collection/gerla-routing-clustered-sicon97.pdf>.
[4] J. Eriksson, M. Faloutsos, S.V. Krishnamurthy, Dart: dynamic address routing for scalable ad hoc and mesh networks, IEEE/ACM Transactions on Networking 15 (2007) 119–132, http://dx.doi.org/10.1109/TNET.2006.890092.
[5] T.Clausen, P.Jacquet, Optimized link state routing protocol (olsr), RFC 3626, Internet Engineering Task Force, October 2003. <http://www.ietf.org/rfc/rfc3626.txt>.
[6] D.B. Johnson, D.A. Maltz, Y.C. Hu, The dynamic source routing protocol for mobile ad hoc networks (dsr), RFC 4728, Internet Engineering Task Force MANET Working Group (February 2007). <http://tools.ietf.org/html/rfc4728>.
[7] C.E. Perkins, E.M. Belding-Royer, S.R. Das, Ad hoc on-demand distance vector (aodv) routing, RFC Experimental 3561, Internet Engineering Task Force (July 2003). <http://www.ietf.org/rfc/rfc3561.txt>.
[8] Chakeres, C. Perkins, Dynamic manet on-demand (dymo) routing, Internet-draft, Internet Engineering Task Force (March 2009). <http://www.ietf.org/internet-drafts/draft-ietf-manet-dy mo-17.txt>.
[9] L. Canourgues, J. Lephay, L. Soyer, A.-L. Beylot, A scalable adaptation of the OLSR protocol for large clustered mobile ad-hoc networks, in: IFIP Annual Mediterranean Ad hoc Networking Conference (MED-HOC-NET), Palma de Mallorca, 23/06/2008-27/06/2008, IFIP, vol. 265, Springer, 2008. <http://www.springerlink.com>.
[10] L. Villasenor-Gonzalez, Y. Ge, L. Lament, HOLSR: a hierarchical proactive routing mechanism for mobile ad hoc networks, IEEE Communications Magazine 43 (7) (2005) 118–125, http://dx.doi.org/10.1109/MCOM.2005.1470838.
[11] G.Pei, M.Gerla, T.-W. Chen, Fisheye state routing: a routing scheme for ad hoc wireless networks, in: IEEE International Conference on Communications, ICC 2000, vol.1, 2000, pp. 70–74. http://dx.doi.org/10.1109/ICC.2000.853066.
[12] G. Koltsidas, G. Dimitriadis, F.-N. Pavlidou, On the performance of the hsls routing protocol for mobile ad hoc networks, Wireless Personal Communications 35 (2005) 241–253, http://dx.doi.org/10.1007/s11277-005-3491-4.

[13] D. Milic, T. Braun, Fisheye: topology aware choice of peers for overlay networks, in: IEEE 34th Conference on Local Computer Networks, 2009. LCN 2009, 2009, pp. 467–474. <http://dx.doi.org/10.1109/LCN.2009.5355168>.

[14] A.D. DeClauzade DeMazieux, M. Marot, M. Becker, Correction, generalisation and validation of the max-min d-cluster formation heuristic, in: Proceedings of the 6th International IFIP-TC6 Conference on Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet, NETWORKING'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 1149–1152. <http://portal.acm.org/citation.cfm?id=1772322.1772448>.

[15] L. Conceição, D. Palma, M. Curado, A novel stable and low-maintenance clustering scheme, in: Proceedings of the 2010 ACM Symposium on Applied Computing, SAC'10, ACM, New York, NY, USA, 2010, pp. 699–705. http://doi.acm.org/10.1145/1774088.1774232. http://portal.acm.org/citation.cfm?id=1772322.1772448.

[16] H. Kim, M. Yoo, A scalable ad hoc routing protocol based on logical topology for ubiquitous community network, in: The 9th International Conference on Advanced Communication Technology. <http://dx.doi.org/10.1109/ICACT.2007.358597>.

[17] B. Zhou, Z. Cao, M. Gerla, Cluster-based inter-domain routing (cidr) protocol for manets, in: Wireless On-Demand Network Systems and Services, 2009, WONS 2009. Sixth International Conference on, 2009, pp. 19–26. <http://dx.doi.org/10.1109/WONS.2009.4801843>.

[18] D. Palma, M. Curado, Increasing reliability in large scale Ad-hoc networks, 4th ERCIM eMobility Workshop, 2010, pp. 39–49. <http://wiki.ercim.eu/wg/eMobility/images/f/fa/ERCIM-WS2010-Proceedings-4.pdf>.

[19] C.-K. Toh, A novel distributed routing protocol to support ad hoc mobile computing, in: Proceedings IEEE 15th Annual International Phoenix Conference on Computers and Communications, IEEE IPCCC 1996, March 27–29, Phoenix, AZ, USA, IEEE, 1996, pp. 480–486. www.ics.uci.edu/%7Eatm/adhoc/paper-collection/toh-distributed-routing-ipccc96.pdf.

[20] M. Shakeri, E. Hosseini, Simulation and evaluation of routing protocols in wireless mobile ad hoc network, in: Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on, 2010, pp. 603 –607. http://dx.doi.org/10.1109/WCINS.2010.5545215.

[21] S. Zhou, L. Ying, On delay constrained multicast capacity of large-scale mobile Ad-hoc networks, in: INFOCOM, 2010 Proceedings IEEE, 2010, pp. 1–5, http://dx.doi.org/10.1109/INFCOM.2010.5462257.

[22] M. Qin, R. Zimmermann, An adaptive strategy for mobile ad hoc media streaming, IEEE Transactions on Multimedia 12 (4) (2010) 317–329, http://dx.doi.org/10.1109/TMM.2010.2046275.

[23] Opnet simulator. <http://www.opnet.com/>.

[24] W. Navidi, T. Camp, Stationary distributions for the random waypoint mobility model, IEEE Transactions on Mobile Computing 3 (1) (2004) 99–108. <http://dblp.uni-trier.de/db/journals/tmc/tmc3.html>.

[25] F.J. Ros, P.M. Ruiz, Cluster-based olsr extensions to reduce control overhead in mobile ad hoc networks, in: Proceedings of the 2007 International Conference on Wireless Communications and Mobile Computing, IWCMC'07, ACM, New York, NY, USA, 2007, pp. 202–207. http://doi.acm.org/10.1145/1280940.1280985.

[26] S. Ortiz, IEEE 802. 11n: the road ahead, Computer 42 (7) (2009) 13–15, http://dx.doi.org/10.1109/MC.2009.224.

[27] G. Anastasi, E. Borgia, M. Conti, E. Gregori, Wi-fi in ad hoc mode: a measurement study, in: IEEE International Conference on Pervasive Computing and Communications, 2004, p. 145. http://dx.doi.org/doi.ieeecomputersociety.org/10.1109/PERCOM.2004.1276853.

[28] B. Xing, K. Seada, N. Venkatasubramanian, An experimental study on wi-fi Ad-hoc mode for mobile device-to-device video delivery, in: INFOCOM Workshops 2009, IEEE, 2009, pp. 1–6, http://dx.doi.org/10.1109/INFCOMW.2009.5072111.

[29] D. Palma, M. Curado, Inside-out olsr scalability analysis, in: P. Ruiz, J. Garcia-Luna-Aceves (Eds.), Ad-hoc, Mobile and Wireless Networks, Lecture Notes in Computer Science, vol. 5793, Springer, Berlin/Heidelberg, 2009, pp. 354–359.

[30] T.Hamma, T.Katoh, B.B. Bista, T.Takata, An efficient zhls routing protocol for mobile ad hoc networks, in: DEXA'06: Proceedings of the 17th International Conference on Database and Expert Systems Applications, IEEE Computer Society, Washington, DC, USA, 2006, pp. 66–70. http://dx.doi.org/10.1109/DEXA.2006.24.

[31] 3GPP, Spatial channel model for multiple input multiple output (mimo) simulations. technical specification group radio access network., Technical Report 7, 3rd Generation Partnership Project, 2008.

[32] Y. Wei-dong, S. Li-min, L. Ji-zhao, A realistic mobility model for urban traffic scenario in vehicular delay tolerant network, in: IET International Conference on Wireless Sensor Network, 2010, IET-WSN. http://dx.doi.org/10.1049/cp.2010.1033.

[33] Olsr daemon for quagga. <http://olsrdq.sourceforge.net>.

[34] A. Tonnesen, T. Lopatic, H. Gredler, B. Petrovitsch, A. Kaplan, S.-O. Tücke etal., Olsrd – an ad-hoc wireless mesh routing daemon, July 2010. <http://www.olsr.org>.

[35] D. Palma, M. Curado, Dash, deferred aggregated routing for scalable ad-hoc networks, in: Wireless Days (WD), 2010 IFIP, 2010, pp. 1–6. http://dx.doi.org/10.1109/WD.2010.5657744.

[36] S. Kristiansen, M. Lindeberg, D. Rodriguez-Fernandez, T. Plagemann, On the forwarding capability of mobile handhelds for video streaming over manets, in: Proceedings of the Second ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds, MobiHeld'10, ACM, New York, NY, USA, 2010, pp. 33–38. http://doi.acm.org/10.1145/1851322.1851332.

[37] D.-W. Kum, J.-S. Park, Y.-Z. Cho, B.-Y. Cheon, Performance evaluation of aodv and dymo routing protocols in manet, in: 2010 7th IEEE Consumer Communications and Networking Conference (CCNC), 2010, pp. 1–2. http://dx.doi.org/10.1109/CCNC.2010.5421657.

[38] X. Xiang, X. Wang, Y. Yang, Supporting efficient scalable multicasting over mobile ad hoc networks, IEEE Transactions on Mobile Computing 10 (4) (2011) 544–559, http://dx.doi.org/10.1109/TMC.2010.176.

[39] C.-C. Yang, L.-P. Tseng, Fisheye zone routing protocol: a multi-level zone routing protocol for mobile ad hoc networks, Computer Communications 30 (2) (2007) 261–268, http://dx.doi.org/10.1016/j.comcom.2006.08.014.

[40] J. Garcia-Luna-Aceves, M. Spohn, Source-tree routing in wireless networks, in: Seventh International Conference on Network Protocols, (ICNP'99) Proceedings, 1999, pp. 273–282.

[41] H. Nakagawa, K. Nakamaru, T. Ohta, Y. Kakuda, A hierarchical routing scheme with location information on autonomous clustering for mobile ad hoc networks, in: International Symposium on Autonomous Decentralized Systems, 2009. ISADS'09. 2009, pp. 1 –6. http://dx.doi.org/10.1109/ISADS.2009.5207383.

[42] K. Kasera, R. Ramanathan, A location management protocol for hierarchically organized multihop mobile wireless networks, in: 1997 IEEE 6th International Conference on Universal Personal Communications Record, Conference Record, vol. 1, 1997, pp. 158–162. http://dx.doi.org/10.1109/ICUPC.1997.625502.

[43] C. Ching-Chuan, W. Hsiao-Kuang, L. Winston, G. Mario, Routing in clustered multihop mobile wireless networks with fading channel, in: Proceedings of IEEE SICON, 1997, pp. 197–211.

[44] Y.T. Mingliang Jiang, Jinyang Li, Cluster based routing protocol(cbrp), Internet-draft, Internet Engineering Task Force, August 1999. <http://tools.ietf.org/html/draft-ietf-manet-cbrp-spec-01>.

[45] G. Zhu, X. Jiang, C. Wu, Z. He, A cluster head selection algorithms in wireless network based on maximal weighted independent set, in: 2010 Proceedings of the 5th International Conference on Ubiquitous Information Technologies and Applications (CUTE), 2010, pp. 1–6. http://dx.doi.org/10.1109/ICUT.2010.5678667.

[46] T. Ohta etal, A class of hierarchical routing protocols based on autonomous clustering for large mobile ad hoc networks, in: Proceedings of IEICE Transactions on Communications, vol. E8 7-B, 2004, pp. 2500–2510.

[47] M. Joa-Ng, I.-T. Lu, A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks, IEEE Journal on Selected Areas in Communications 17 (1999) 1415–1425.

[48] N. Nikaein, H. Labiod, C. Bonnet, Ddr-distributed dynamic routing algorithm for mobile ad hoc networks, in: 2000 First Annual Workshop on Mobile and Ad Hoc Networking and Computing, MobiHOC, 2000, pp. 19–27. http://dx.doi.org/10.1109/MOBHOC.2000.869209.

[49] N.Nikaein, C.Bonnet, N.Nikaein, Harp – hybrid ad-hoc routing protocol, in: Proceedings of IST: International Symposium on Telecommunications, 2001.

[50] M. Zhang, P.H.J. Chong, Performance comparison of flat and cluster-based hierarchical ad hoc routing with entity and group mobility, in: WCNC'09: Proceedings of the 2009 IEEE Conference on Wireless Communications & Networking Conference, IEEE Press, Piscataway, NJ, USA, 2009, pp. 2450–2455.

[51] ITU-T, Performance and quality of service requirements for International Mobile Telecommunications-2000 (IMT-2000) access networks, ITU-T, 2003.