Supporting distance vector routing over device discovery flows in the pervasive middleware PalCom

Amr Ergawy*, Boris Magnusson

Department of Computer Science, Lund University, Ole Römers väg 3, Lund SE-223 63, Sweden

Abstract

PalCom is a pervasive middleware that enables users to combine services from different devices into assemblies, or usage scenarios. The mechanism of device discovery in PalCom builds an ad-hoc network of interconnected devices while replacing cross-network periodic keep-alive messages with timely forwarded events that notify devices about the discovery and undiscovery of each other. Moreover, the device discovery mechanism is decoupled from any specific routing decisions mechanism. In this paper, we present a design that builds on this flexibility to add distance vector routing to PalCom while inheriting the event based cross-network device discovery for updating routing tables on interacting devices. The proposed design has been implemented and evaluated against a network with links that continuously change their availability.

Keywords: pervasive middleware; device discovery; ad-hoc networking.

1. Introduction

PalCom is a pervasive middleware that enables users to integrate services of different devices, possibly on different networks, into assemblies of use cases1. It provides a service discovery protocol that enables those devices to exchange their services descriptions. Using a tool that can display such descriptions in a user friendly interface, a user can write,
or compose, simple services assembly scripts that enable exchanging data and commands among those devices. We refer to that approach as a human, or user, in the loop, where users are given more influence on system configuration and control.

As a previous work, we defined the lowest layer of the PalCom stack that enables third-party developers to write Media Abstraction Objects, MAOs, which abstract different types of network interfaces to upper PalCom layers. As an extension, we detailed a device discovery mechanism for PalCom that enables devices to discover each other and to build an ad-hoc network among themselves. For development purposes, we assumed to route a message to a destination node via the first discovered route to it. In this work we introduce distance vector routing in PalCom.

In this paper, we explain the design challenges, principles, options and details of supporting distance vector routing in PalCom. In section 2, we summarize our previous work on the PalCom stack. Also, we survey ad-hoc routing protocols to compare their signaling and control overhead with that of a routing protocol over our device discovery mechanism. Then, in section 3, we detail our design before we summarize its implementation and evaluation in section 4. Finally we explain our conclusions and future work in section 5.

2. Previous work

We build the support of distance vector routing in PalCom on top of our previous work of networking media abstraction and device discovery in PalCom, which provides a signaling and control utility for supporting distance vector routing in PalCom.

2.1. Device discovery over networking media abstraction in PalCom

As shown in Fig. 1, our previous work defines the lowest layer of the PalCom stack as the Media Abstraction Layer, MAL. It enables the development and integration of media abstraction objects, MAOs, that abstract different

---

Fig. 1. The components and functionality of the Device Discovery Layer, DLL, as part of the PalCom stack.
network protocols to PalCom upper layers. As a result, PalCom nodes can communicate over heterogeneous networks. A level up, we define the Device Discovery Layer, DDL, that enables devices to discover each other while constructing an overlay network among themselves. As shown in Fig. 1, the DDL is composed of the Discovery State Machine, DSM, the Discovery Forwarding Manager, DFM, and the Routing Table, RT. The DSM uses periodic heartbeat messages to discover local-routes to devices on local networks. Also, the DSM uses forward discovery events to discover remote-routes on reachable remote-networks. As shown in Fig. 1, both of the route types are maintained in the RT.

To forward discovery events about a known device, the DFM arranges the network interfaces of the local node as introducers and advertisers of that device. As defined in Fig. 1, we refer to such arrangement as a Discovery Forwarding Flow, DFF. While a device is discovered, the DFM maintains at most two DFFs. As a result, for a single device discovery event, the maximum number of changes of the roles of the network interfaces, of the local node, as introducers and advertisers is two. This case happens when the known device disappears from one of the introducers of the two DFFs while it had appeared earlier via one of its advertisers. Such behavior minimizes the changes in the topology of the constructed ad-hoc network as a result of device discovery.

2.2. Challenges and design options for supporting distance vector routing in ad-hoc networks

A key challenge to distance vector routing protocols in ad-hoc networks is route discovery. From this perspective, these protocols are either table-driven protocols or on-demand protocols. Table-driven protocols are criticized for the overhead of exchanging routing table updates among nodes. Table-driven protocols include Destination Sequenced Distance Vector Routing, DSDV, and its fault-tolerant improved version, I-DSDV.

In on-demand routing protocols, a sender node attempts to discover a route to a destination node only on sending a message, which is criticized for route-setup latency. They include Ad-hoc On-demand Distance Vector Routing, AODV, and its variant of Ad-hoc On-demand Multipath Distance Vector Routing, AOMDV.

Compared to other table-driven protocols, distance vector routing in PalCom delegates the maintenance of routing tables to PalCom device discovery. In contrast to on-demand protocols, distance vector routing in PalCom avoids the route discovery stage, as it utilizes the PalCom device discovery mechanism.

Another challenge to distance vector routing protocols in ad-hoc networks is the need to flood a network with control messages. E.g. on-demand protocols may flood a network with route-requests. To address this challenge, these protocols may select a specific sub-group of nodes to forward a broadcast message. Also, such protocols may utilize information about the destination recent neighbors to forward route-requests. Compared to such approaches, PalCom device discovery limits periodic keep-alive messages to local networks while it forwards device discovery events across networks. Distance vector routing in PalCom inherits that optimization.

A third challenge that faces distance vector routing in ad-hoc networks is the detection of, and overcoming, broken links. These protocols may utilize information from the underlying Media Access Layer, MAC, protocols to detect broken links and reduce duplicate control messages that may be broadcasted to overcome them. Oppositely, a distance routing protocol in PalCom cannot make use of MAC layer information as networking protocols are abstracted by the media abstraction layer, as explained in section 2. Instead, device discovery in PalCom provides a device undiscovery mechanism that detects the disappearance of a device via a specific network interface, possibly indicating a broken link. Additionally, the PalCom media abstraction layer provides the ability to reach a device via different networking interfaces, which may overcome a broken link.

3. The proposed support for distance vector routing in PalCom

In distance vector routing, a source/intermediate node forwards a message to a destination node via the lowest cost route. Thus, the main goal of our design is to extend one of the components of the PalCom Device Discovery Layer, DDL, summarized above in section 2, to:

- Maintain route cost based natural ordering of the routes set to a known device, reflecting route cost updates that are conveyed in discovery messages that are received via introducer routes to that known device.
- Report route cost updates to neighbor nodes based on that natural ordering of the introducer routes.
Ideally, the logic of the chosen DDL component needs to be already designed to maintain some relation among the routes to a known device. We refer to such logic as *routes-set oriented*, as opposite to *device-state oriented* logic that makes decisions only based on device appearance, disappearance, and status updates. In this section, we start by analysing the DDL components as design options. Then, we emphasize our design principles and explain its details.

### 3.1. Design options

As mentioned above in this section, we aim to add the extension of distance vector routing to the component of the PalCom device discovery that already implements routes-set oriented logic more than device-state oriented logic.

#### 3.1.1. Analyzing the discovery state machine

We start by considering the discovery state machine, DSM, as the first component of PalCom device discovery. As mentioned in section 2, it implements both the local discovery logic as well as the cross networks discovery logic. By analyzing that logic, we find that the DSM makes its decisions based on the combination of:

- The parameters of the concerned device that are conveyed in an input device discovery message.
- The current state of the source route, to the concerned device, via which the input message was received.
- The current values of the parameters of the concerned device.
- The collective state of the concerned device via its routes set.

It is clear that the DSM does not maintain any relation among the routes to a known device and it makes its decisions only based on the device state. In conclusion, it is not feasible to extend the DSM to support distance vector routing.

#### 3.1.2. Analyzing the discovery forwarding manager

Alternatively, considering the discovery forwarding manager, DFM, summarized in section 2, it implements the logic that triggers forwarding discovery events to neighbor nodes and aligns the state of remote routes to that of the local routes of their <next-hop> parameter, as shown in Fig. 1. By analyzing that logic, we find that the DFM makes its decisions based on the combination of:

- A device discovery event that was triggered by the discovery state machine.
- The latest forwarded discovery event to neighbor nodes about the concerned device.
- The effect of the state of a local route to the concerned device on the state of a remote-route, to another device.

Similarly to the discovery state machine, DSM, the logic of the DFM does not maintain a relation among the routes to a known device. In conclusion, it is not feasible to extend the DFM to support distance vector routing.

#### 3.1.3. Analyzing the discovery forwarding flows

Finally, considering the discovery forwarding flows, DFFs, summarized in section 2, it implements the logic that arranges the network interfaces on a PalCom node into flows via which triggered discovery events are forwarded to neighbor nodes. It makes its decisions based on the combination of:

- The temporal order of the discovery of the introducer routes to the concerned device. In particular, as summarized in section 2, the interface of the first discovered route to the concerned device is the introducer of its first discovery flow, DFF1 while the interface of the second discovered route to that device is the introducer of its second discovery flow, DFF2.
- The current set of network interfaces on the local node.

To sum up, the logic of the DFFs maintains a relation among the interfaces of the local node that are associated with routes to a known device. In conclusion, it is feasible to extend the DFFs to support distance vector routing.

### 3.2. Design principles and properties

In this section, we list and discuss the three design principles that we follow to specify the details of extending the discovery forwarding flows to implement distance vector routing.
3.2.1. Discovery state consistency

As summarized in section 2, and shown in Fig. 1, the discovery forwarding flows, DFFs, advertise the discovery state of a known device in consistency with the discovery state machine, DSM, on the advertiser PalCom node. Our extension of the DFFs to support distance vector routing must preserve this consistency by maintaining two properties:

- To only function as a processor of discovery state events.
- To never influence or create discovery state events.

3.2.2. Separation of concerns

As we mentioned in section 2, and illustrated in Fig. 1, the discovery forwarding flows, DFFs, separate the discovery advertisement logic from the device discovery logic, which is implemented by the discovery state machine, DSM, and the discovery forwarding manager, DFM. For a better system modularity, the extension of the DFFs to support distance vector routing must preserve such separation of concerns by maintaining two properties:

- To only maintain the relations among interfaces on the local node and routes to discovered devices.
- To never implement or influence device discovery logic.

3.2.3. Minimizing routes dynamics

As mentioned in section 2, in response to a discovery event about a known device, the discovery forwarding flows, DFFs, ensures a maximum of two changes of roles of the network interfaces, of the local node, as introducers and advertisers of that known device, which minimizes the changes of the topology of the discovered ad-hoc network.

The extension of the discovery forwarding flows to support distance vector routing shall keep route dynamics within the current limits by maintaining two properties:

- To keep the maximum number of changes in interfaces roles, as introducers and advertisers, per discovery event as close as possible to the above mentioned maximum of two.
- To keep minimal changes to interfaces roles per interface-event, e.g. interface enabled or interface disabled.

3.3. Design details

We start this section by defining the modifications to the formats of discovery message to convey route cost values and how we update route costs in routing tables. Then, we define the necessary modifications of the interfacing of the discovery forwarding flows, DFFs, to other device discovery components, to be notified about route cost updates. Finally, we detail how the DFFs maintain the route cost based natural ordering of the routes set to a known device.

3.3.1. Modifications of the discovery messages formats and updating the routing table

PalCom discovery messages are composed of nodes. We define a message type as a dash separated sequence of nodes, where a node is symbolized as a capital letter. Forwarded device discovery events are formatted as either forwarded device information reply messages, noted as S-I, or forwarded heart-beat reply messages, noted as S-H, where an S-node marks a forwarded message, an I-node marks a device information reply message, and an H-node marks a heart-beat-reply message.

To convey route cost values, we define a new node, symbolized as C. When a PalCom node receives a discovery message that triggers a device discovery event, e.g. an S-I or an S-H message, then a C-node is inserted into that message before re-broadcasting it, to advertise the received event. If the received discovery message triggers no device discovery event, then only an S-C message is broadcasted to advertise the route cost update.

Moreover, according to the design principle of the separation of concerns, the discovery forwarding flows cannot modify the routing table to update route entries with route cost updates. Instead, we consider the route cost to be a route entry parameter that is updated by the discovery state machine and the discovery forwarding manager.

3.3.2. Defining the component interfacing of the discovery forwarding flows

We specify how the discovery state machine, DSM, and the discovery forwarding manager, DFM, shall be interfaced to the extended discovery forwarding flows, DFFs, and the necessary procedures of such interfacing. According to the design principle of discovery state consistency, a discovery event disseminates from the DSM, to the
DFM, and finally to the DFFs. Also, we specified above that only the DSM and the DFM may update a route cost update to a route entry in the routing table. From all of the above, we can preserve the current design properties where:

- The discovery forwarding flows, DFFs, need only to be interfaced to the discovery forwarding manager, DFM.
- Among the parameters of that interfacing, we may only pass route entries with already updated route cost values.

Thus, the interfacing procedures of the DFFs, defined in table 1, handle the events of adding, deleting, and updating routes, to discovered devices, and network interfaces, on the local node. As a last step, these procedures update the DFFs and advertise discovery events using the procedure `update_advertisements`, which takes two parameters:

- The type of the message node that indicates a device appearance or update that will be advertised via interfaces that become advertisers after updating the DFFs. If a caller procedure, from table 1, does not specify this parameter, we assume the default-appearence-or-update-node, i.e. an information-reply I-node.
- The type of the message node that indicates a device disappearance that will be advertised via detached advertiser interfaces of a disappearing introducer, after updating the DFFs. If a caller procedure, from table 1, does not specify this parameter, we assume the default-disappearance-node, i.e. an interface-closed Y-node.

Table 1. Procedures of the extended discovery forwarding flows to support distance vector routing.

<table>
<thead>
<tr>
<th>Event</th>
<th>Discovery flow element</th>
<th>Name</th>
<th>Details</th>
<th>Network interface</th>
</tr>
</thead>
</table>
| Add     | A newly discovered route to a known device, with updated route cost.                   | Processing | For the concerned known device:  
1. Add the parameter route as a introducer to the `route cost based sorted set` of introducers.  
2. `update_advertisements` (default-appearence-or-update-node, default-disappearance-node). |          | An enabled network-interface on the local node.                                     |
| Remove  | An introducer route via which a known device has disappeared.                           | Processing | For the concerned known device:  
1. Remove the parameter route from the `route cost based sorted set` of the introducers.  
2. `update_advertisements` (default-appearence-or-update-node, Y-node or X-node). |          | A disabled network-interface on the local node.                                     |
| Update  | An introducer route via which the parameters of a known device and/or the route cost to it has been updated.  
2. The node type in the advertisement discovery message that indicates the type of the device parameters update via this route. Only three values are expected, a heart-beat-reply H-node, an information-reply I-node, or a Null to indicate only route cost update.  
3. A boolean to indicate whether the discovery state machine and the discovery forwarding manager updated the route cost of that introducer route. | Processing | For the concerned known device:  
1. If the third parameter indicates an updated route cost, resort the `route cost based sorted set` of the device introducer routes.  
2. `update_advertisements` (H-node, I-node, or Null, default-disappearance-node). |          | N/A                                                                              |
3.3.3. Maintaining the routing cost based sorted routes set and the discovery forwarding flows

As we summarized in section 2, we define a discovery forwarding flow as the pair (introducer route, advertiser interfaces)\(^1\). Accordingly, for a known device, we define the two maintained route cost based discovery forwarding flows, DFFs, as follows:

- **DFF1** = (introducer= the network-interface, on the local node, with the lowest cost route to the known device, advertisers= the rest of the network-interfaces, on the local node).
- **DFF2** = (introducer= the network-interface, on the local node, with the 2nd lowest cost route to the known device, advertisers= the network-interface, on the local node, with the lowest cost route to the known device).

As explained above, for a known device, the DFFs maintain a routing cost based sorted set of the discovered routes to that device. On a call from one of the procedures in table 1, the sorted routes set is used to maintain DFF1 and DFF2 of the concerned device according to their above mentioned definitions, which have the same structure as the original definitions of DFF1 and DFF2 that we summarized in section 2. In turn, the extended DFFs inherits the implementation of the principle of minimizing routes dynamics.

4. Implementation and evaluation

We implemented our solution as part of the Java implementation of PalCom. We used the test setup in Fig. 2-a to evaluate routing decisions of a message stream with a sending rate of one second from device-1 to device-2 in the

---

Fig. 2. (a) test network for evaluating the implemented support of distance vector routing in PalCom; (b) the number of hops the received test messages, at the destination node, took in response to changing network availability at the source node.
condition of changing routes availability. At the start of the test, routes from device-1 to device-2 are available by enabling the connections via LAN-1, LAN-2, and LAN-4. Then, during a test round, we use an automated script to disable LAN-1, LAN-2, and LAN-4 in-sequence with a period of 15 seconds between each of these actions. Afterward, in the same round, we re-enable those networks in the reverse order.

In Fig. 2-b, we show the sequence of received test messages on device-2 during three test rounds, which have the same pattern of the number of hops that test messages took in response to changing network connectivity. In particular, on the availability of LAN-1, LAN-2, and LAN-4, test messages from device-1 took the single hop route via LAN-1 to device-2. On the availability of only LAN-2 and LAN-4, test messages from device-1 took the two hops route via LAN-2 to device-2. Finally, on the availability of only LAN-4, test messages from device-1 took the three hops route to device-2. According to these results, the introduced distance vector routing for PalCom makes proper and timely routing decisions in response to changing network connectivity.

5. Conclusion and future work

Supporting distance vector routing in PalCom over device Discovery Forwarding Flows, DFF, avoids the need for specialized signaling and control messages compared to other table-driven and compared to on-demand ad-hoc networking protocols. In particular the PalCom device discovery mechanism provides the necessary autonomous management of the ad-hoc network of discovered PalCom devices. The proposed solution was integrated into the Java implementation of PalCom. The evaluation results show that correct and timely routing decisions are made by the system in response to availability changes of the network connections.

As a future work, we plan to enhance the device discovery mechanism itself by adding a revision system of the routing table contents that enables re-synchronizing discovery information among PalCom devices after temporary disconnections that last for a time interval that is shorter than the configured heart-beat periods. Moreover, we plan to review the device discovery mechanism to prevent a noticed problem where a router node may forward an appearance discovery event about a remote route to a just disappeared neighbor node while that router node was the only gateway to that just disappeared neighbor node.

References