An Efficient Filter-based Addressing Protocol for Autoconfiguration of Mobile Ad Hoc Networks

IEEE INFOCOM 2009
Fernandes, N. C.  Moreira, M. D. D.  Duarte, O. C. M. B.

Presented by Hsi-Min Lin
7/2, 2009
OUTLINE

- INTRODUCTION
- RELATED WORK
- FILTER-BASED ADDRESSING PROTOCOL (FAP)
  - Sequence Filters
  - Bloom Filters
  - Procedures of FAP
- ANALYSIS
- SIMULATION
- CONCLUSIONs
INTRODUCTION

- Address allocation problem in ad hoc network
  - Auto-configures address
  - Distributed
  - Address collision problem
  - Address allocation delay problem
  - Partitions problem
  - Control overhead constraint
    - Life time of nodes
    - Available bandwidth to send data
  - Storage constraint
RELATED WORK

- Duplicate Address Detection (DAD)
  - Randomly chooses an address
  - Address Request message (AREQ)
    - For allocated address
  - Address Reply message (AREP)
    - Handle address collision
  - Does not handle network partitions
RELATED WORK

- Duplicate Address Detection (DAD)
RELATED WORK

- Duplicate Address Detection (DAD)
  - Randomly chooses an address
  - Address Request message (AREQ)
    - For allocated address
  - Address Reply message (AREP)
    - Handle address collision
  - Does not handle network partitions
RELATED WORK

- Fan and Subramani propose a protocol based on DAD
  - Called FS in this paper
  - Use Hello messages
  - Use random numbers as network identifiers to detect network merging

- Fazio et al. also propose a protocol
  - Based on network identifiers
  - Works in a reactive fashion

- MANETconf
  - Allocated IP list and allocated-pending list
  - Partition detection depends on periodic flooding
  - Large number of control messages and required to have a high storage capacity.

- Dynamic Address assignment Protocol (DAP)
  - High control load required to have a high storage capacity.
FILTER-BASED ADDRESSING PROTOCOL (FAP)

Aims
- Dynamically autoconfigure addresses, identifying and solving addresses collisions with a low control load

Filters
- Represent the current set of allocated addresses.
- Hash of the filter (filter signature) as a partition identifier.
Sequence Filters

146.164.69.1 0 0 0 .......... 0 0 0

Initial address Number of available address
Sequence Filters

146.164.69.1 0 0 1 ........ 0 0 0

146.164.69.1 146.164.69.254

146.164.69.3

Network prefix

Host suffix
Bloom Filters

Element to be inserted

146.164.69.3

Hash functions

\[ H_1, H_2, \ldots, H_K \]

Filter

\[ \begin{align*}
0 \\
0 \\
0 \\
\ldots \\
0 \\
0 \\
0 \\
0 \\
\end{align*} \]

\[ m \]
Bloom Filters

Hash functions

\[ H_1 \quad H_2 \quad \ldots \quad H_K \]

<table>
<thead>
<tr>
<th>Filter</th>
<th>Counters</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 0 )</td>
<td>( 0 )</td>
</tr>
<tr>
<td>( 0 )</td>
<td>( 0 )</td>
</tr>
<tr>
<td>( \ldots )</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( 0 )</td>
<td>( 0 )</td>
</tr>
<tr>
<td>( 0 )</td>
<td>( 0 )</td>
</tr>
<tr>
<td>( 0 )</td>
<td>( 0 )</td>
</tr>
<tr>
<td>( 0 )</td>
<td>( 0 )</td>
</tr>
</tbody>
</table>

146.164.69.3
Bloom Filters

Hash functions

<table>
<thead>
<tr>
<th>H₁</th>
<th>H₂</th>
<th>H₃</th>
<th>H₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>146.164.69.3</td>
</tr>
</tbody>
</table>

Filter

<table>
<thead>
<tr>
<th>Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

Counters

<table>
<thead>
<tr>
<th>Counters</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
</tr>
</tbody>
</table>
## Bloom Filters

### Hash functions
- \( H_1 \)
- \( H_2 \)
- \( H_3 \)
- \( H_4 \)

### Example Entry
- 146.164.69.3

<table>
<thead>
<tr>
<th>Filter</th>
<th>Counters</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1</td>
<td>0 0 1</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0</td>
</tr>
</tbody>
</table>

1
Bloom Filters

Hash functions

146.164.69.3

Filter

1
0
1
0
0
0
0
0

Counters

0 0 1
0 0 0
0 0 1
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0

1 1
Bloom Filters

Hash functions

146.164.69.3

Filter

<table>
<thead>
<tr>
<th>1</th>
<th>0</th>
</tr>
</thead>
</table>

Counters

<table>
<thead>
<tr>
<th>001</th>
<th>000</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>0110</th>
<th>0100</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>0000</th>
<th>0000</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>0000</th>
<th>0000</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>0000</th>
<th>0000</th>
</tr>
</thead>
</table>
Bloom Filters

Hash functions

146.164.69.3

Filter
1
0
1
0
0
1

Counters
0 0 1
0 0 0
0 1 0
0 0 0
0 0 0
0 0 0
0 0 1
Procedures of FAP
Procedures of FAP

(a) Hello.

(b) Address Request (AREQ).

(c) Address Filter (AF).

(d) Partition.
Procedures of FAP

A partition’s Filter

| 146.164.69.1 | 0 | 0 | 1 | ... | 0 | 0 | 0 |

B partition’s Filter

| 146.164.69.1 | 0 | 0 | 1 | ... | 0 | 1 | 0 |

Hash function for Filter Signature

A partition’s Filter Signature

52ED879E

B partition’s Filter Signature

46042841
Procedures of FAP

Network Initialization

<table>
<thead>
<tr>
<th>Type</th>
<th>Reserved</th>
<th>Source</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) Hello.

(b) Address Request (AREQ).

Flood $N_R$ times

A: 3

B: 5

C: 6
Procedures of FAP

Join Network

(b) Address Request (AREQ).

AF with I = 1

A

B

100110

100110

100110

100110

Type | Reserved
Source
Node forwarding the packet
Sequence Number
Signature
Procedures of FAP

Partition merging

<table>
<thead>
<tr>
<th>Type</th>
<th>M</th>
<th>Counter</th>
<th>Seq. Num</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address filter of the other partition</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Signatures of the other partition</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>I</th>
<th>Counter</th>
<th>R</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address Filter</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Signatures</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(d) Partition

```
1 0 1 1 0 0
1 0 1 0 1 1
```

(c) Address Filter (AF)

```
1 0 1 0 1 1
1 0 1 1 0 0
```

AF with R = 0

Partition

```
1 0 1 1 0 0
```

C

A

D

B

23
Procedures of FAP

Partition merging (with collision)

<table>
<thead>
<tr>
<th>Type</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td></td>
</tr>
<tr>
<td>Node forwarding the packet</td>
<td></td>
</tr>
<tr>
<td>Sequence Number</td>
<td></td>
</tr>
<tr>
<td>Signature</td>
<td></td>
</tr>
</tbody>
</table>

(b) Address Request (AREQ).
(a) Partition.

AREQ

Partition
Address should become available

- The nodes could flood the network with a notification to release its address (removal of its address from the address filter of each network node)

- Otherwise, the address remains allocated on the filters
ANALYSIS

- The available addresses scarce after many departures
- Message loss
- Two different nodes generate Address Requests (AREQs) with the same address and the same sequence number
- Two isolated partitions have exactly the same filters and try to merge
SIMULATION

Initialization control load

Delay to allocate an address on the initialization
SIMULATION

Number of address collisions with 50 nodes on a partition merging.

Control load with 50 nodes on partition mergings and two transmissions of flooding messages.
CONCLUSIONS

- Filter-based Addressing protocol (FAP)
  - Handles nodes joining/leaving the network and partition mergings in a **distributive way**
  - Uses **address filters** to allocate addresses
    - Reduce the **control load**
    - Reduce the **delay**
    - Accurate **partition** merging detection
    - Increase the protocol **robustness**
      - Robust to messages losses