A short survey of basic algorithmic problems in distributed ad hoc systems

Michal Kalewski

Institute of Computing Science
Poznan University of Technology

$Id: adhoc-alg.lyx,v 1.3 2006/03/07 07:58:17 mkalewski Exp $

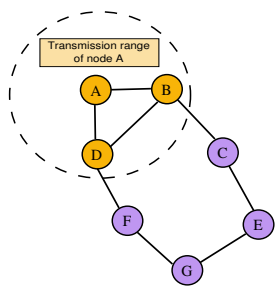
Outline

Ad hoc networks

- *Mobile ad hoc networks* (MANETs) are composed of autonomous and mobile hosts (or communications devices), which communicate through wireless links.
- Each pair of such devices, whose distance is less than their transmission range, can communicate directly with each other – a message sent by any host may be received by all the hosts in its vicinity.
- If hosts in MANET function as both a computing host and a router they form a *multiple hop ad hoc network*.

Ad hoc networks

- Hosts can come and go or appear in new places, so with an ad hoc network, the topology may be changing all the time and can get partitioned and reconnected in highly unpredictable manner.
- Mobile hosts in ad hoc network can exchange information in areas that do not have preexisting infrastructure in decentralized way (the control of the network is distributed among the hosts).



Ad hoc network:

- no fixed infrastructure,
- dynamic topology with partitioning,
- self-organization – distributed control of all operations,
- limited bandwidth, memory and other computational resources,
- limited security.

Ad hoc networks

Ad hoc network:

- no fixed infrastructure,
- dynamic topology with partitioning,
- self-organization – distributed control of all operations,
- limited bandwidth, memory and other computational resources,
- limited security.

Michal Kalewski Ad hoc networks [4/34]

Ad hoc networks

Ad hoc network:

- no fixed infrastructure,
- dynamic topology with partitioning,
- self-organization – distributed control of all operations,
- limited bandwidth, memory and other computational resources,
- limited security.

Michal Kalewski Ad hoc networks [4/34]

Ad hoc networks

Ad hoc network:

- no fixed infrastructure,
- dynamic topology with partitioning,
- self-organization – distributed control of all operations,
- limited bandwidth, memory and other computational resources,
- limited security.

Michal Kalewski Ad hoc networks [4/34]

Ad hoc networks

Ad hoc network:

- no fixed infrastructure,
- dynamic topology with partitioning,
- self-organization – distributed control of all operations,
- limited bandwidth, memory and other computational resources,
- limited security.

Ad hoc network model

The topology of the ad hoc network is modeled by a undirected graph $G = (V, E)$, where V is the set of nodes (hosts-routers) and E is the set of links between neighboring nodes (that is between nodes that can communicate directly).

Since the nodes are mobile, the E set changes with time.

Links between two adjacent nodes are always bi-directional and every node $v \in V$ has a unique *physical address* or id.

Ad hoc network model

The following properties of graph G are defined :

- G_P: graph G can get partitioned and reconnected;
- G_{NP}: graph G does not get disconnected (partitioned), but nodes still are mobile;
- V_C: the V set can change with time – that is the number of nodes that take part in processing can change in time;
- V_{NC}: the V set does not change with time – that is the number of nodes that take part in processing is constant.

Ad hoc network model

Finally we receive four graphs with different assumptions:

- $G_{C|P}$ – graph can get partitioned and the number of nodes can change;
- $G_{NC|P}$ – graph can get partitioned but the number of nodes is constant;
- $G_{C|NP}$ – graph does not get disconnected but the number of nodes can change and
- $G_{NC|NP}$ – graph does not get disconnected and the number of nodes is constant.

Routing

In a fixed network, if a router has a valid path to some destination – missing the point of failure – that path continues to be valid possible indefinitely.

With MANETs frequent topology changes caused by nodes mobility involve that validity of paths can change spontaneously.

That makes routing in ad hoc networks different form routing in their fixed counterparts and a challenging problem.

Routing

Routing protocols for ad hoc networks are classified into three different groups:

- global/proactive – the routes to all the destinations are determined at the start up and maintained by using periodic route update process;
- on-demand/reactive – the routes are determined when they are required by the source using a route discovery process;
- hybrid – combine the basic properties of the first two classes of protocols into one.

AODV Routing

One of most popular routing protocols is reactive **Ad Hoc On-Demand Distance Vector Routing** (AODV) – described in RFC 3561.

AODV builds routes using a route request and route reply messages query cycle.

▸ AODV

AODV Routing – algorithm

If P_i does not have entry in routing table for P_j, it has to discover a route to P_j by construct and broadcast ROUTE_REQUEST packet.

```
struct ROUTE_REQUEST {
  src_addr;
  request_id;  //incremented whenever a ROUTE_REQUEST is broadcast
  dest_addr;
  dest_seq;    //the most recent value of Pj's sequence value that Pi has seen
  hop_count    //keep truck of how many hops the packet has made
};
```

```
struct ROUTE_REPLY {
  src_addr;
  dest_addr;  //copied from incoming ROUTE_REQUEST
  dest_seq;   //taken from local counter in memory
  hop_count;
  lifetime    //controls how long the route is valid
};
```

AODV Routing – algorithm

When a $\mathrm{ROUTE\_REQUEST}$ packet arrives at a node P_k $(k \neq j)$, it is processed in the following steps:

1. $<\mathrm{src\_addr}, \mathrm{request\_id}>$ is looked up in a local *history table*: (i) if this request has already been processed, it is discarded and processing stops; (ii) if it is not a duplicate, the pair is entered into the *history table*.
2. P_k looks up the destination in its route table: if $\mathrm{rt.dest\_seq} \geq \mathrm{dest\_seq}$ a $\mathrm{ROUTE\_REPLY}$ packet is sent back to the source ($\mathrm{rt.dest\_seq}$ – sequence number stored in the local routing table); else step (3) is executed.
3. P_k increments the $\mathrm{hop\_count}$ field and rebroadcasts the packet; data from packet are stored as a new entry in local *reverse route table* (a timer is also started for newly-made reverse route entry).

AODV Routing – algorithm

When a $\mathrm{ROUTE\_REQUEST}$ packet arrives at a node P_k $(k \neq j)$, it is processed in the following steps:

1. $<\mathrm{src\_addr}, \mathrm{request\_id}>$ is looked up in a local *history table*: (i) if this request has already been processed, it is discarded and processing stops; (ii) if it is not a duplicate, the pair is entered into the *history table*.
2. P_k looks up the destination in its route table: if $\mathrm{rt.dest\_seq} \geq \mathrm{dest\_seq}$ a $\mathrm{ROUTE\_REPLY}$ packet is sent back to the source ($\mathrm{rt.dest\_seq}$ – sequence number stored in the local routing table); else step (3) is executed.
3. P_k increments the $\mathrm{hop\_count}$ field and rebroadcasts the packet; data from packet are stored as a new entry in local *reverse route table* (a timer is also started for newly-made reverse route entry).

Michal Kalewski Ad hoc networks [12/34]

AODV Routing – algorithm

When a $\mathrm{ROUTE\_REQUEST}$ packet arrives at a node P_k $(k \neq j)$, it is processed in the following steps:

1. $<\mathrm{src\_addr}, \mathrm{request\_id}>$ is looked up in a local *history table*: (i) if this request has already been processed, it is discarded and processing stops; (ii) if it is not a duplicate, the pair is entered into the *history table*.
2. P_k looks up the destination in its route table: if $\mathrm{rt.dest\_seq} \geq \mathrm{dest\_seq}$ a $\mathrm{ROUTE\_REPLY}$ packet is sent back to the source ($\mathrm{rt.dest\_seq}$ – sequence number stored in the local routing table); else step (3) is executed.
3. P_k increments the $\mathrm{hop\_count}$ field and rebroadcasts the packet; data from packet are stored as a new entry in local *reverse route table* (a timer is also started for newly-made reverse route entry).

Michal Kalewski Ad hoc networks [12/34]

AODV Routing – algorithm

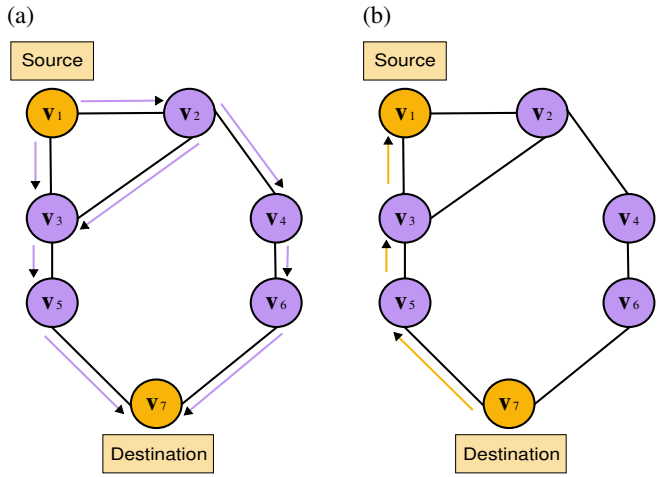
In response to the incoming request, P_j builds a $\mathrm{ROUTE\_REPLY}$ packet and unicast it to the node that the $\mathrm{ROUTE\_REQUEST}$ packet came from.

At each node on the way back ($\mathrm{ROUTE\_REQUEST}$) reverse route table is used to unicast the packet to the source; local host also updates its routing table if one (or more) of the following three conditions are met:

- no route to P_j is know,
- the sequence number for P_j in the $\mathrm{ROUTE\_REPLY}$ packet is greater than the value in the routing table,
- the sequence numbers are equal but the new route is shorter.

Periodically, each node broadcast a Hello message for route maintenance.■

AODV Routing – example



Replication and consistency

Since in ad hoc networks hosts move freely, disconnections may occur frequently and this may cause frequent network partitioning.

Consequently, data and services accessibility in MANETs are lower than in fixed networks and distributed systems.

One possible approach to improve this is to replicate data (or services) among nodes in ad hoc network, but replication creates problem of inconsistency when partitioning occurs with nodes containing replicated data.

Thus when designing a replication protocol for dynamic systems with partitioning, the competing and not independent goals of *availability* and *correctness* must be met.

Replication and consistency

Availability indicates data and services accessibility with system's normal function disrupted as little as possible and correctness indicates accuracy of replicated nodes (with respect to coherency protocol).

To ensure this replication in MANETs requires that the following three main issues should be addressed:

- *replica allocation/relocation*,
- *location management* and
- *consistency management.*

Replica allocation/relocation

Replica allocation determines how many replicas of each data item are created and to which nodes and how these replicas are allocated.

In fixed networks optimal replication scheme depends only on the read-write pattern for each item and in many researches the *communication cost* and *communication time* of replication scheme are used as the cost functions.

The communication cost is the average number of messages required for a read or write of the data item and the communication time of replication scheme is the average communication time of read or write of the data item – communication time of an operation is its longest message path.

Replica allocation/relocation

The problem of finding optimal allocation scheme has been proved to be NP-complete (for different cost models) for both general static distributed systems and ad hoc networks.

Unlike traditional distributed systems where the cost functions are independent of network topology, in MANETs location of replicated data items should be allocated (also) as a function of network topology.

Replica allocation methods

- Static Access Frequency (SAF): each node allocates replicas of data items in descending order of the access frequencies.
- Dynamic Access Frequency and Neighborhood (DAFN): the access frequency to each data item and the neighborhood among nodes are taken into account.
- Dynamic Connectivity based Grouping (DCG): shares replicas in larger groups of nodes than the DAFN method; in order to share replicas effectively, each group should be stable, so the DCG creates groups that are biconnected components. ▸ DCG

Biconnected component:

- denotes a maximum partial graph G which is connected (not divided) if an arbitrary node in the graph G is deleted.

Replica allocation methods

- Static Access Frequency (SAF): each node allocates replicas of data items in descending order of the access frequencies.
- Dynamic Access Frequency and Neighborhood (DAFN): the access frequency to each data item and the neighborhood among nodes are taken into account.
- Dynamic Connectivity based Grouping (DCG): shares replicas in larger groups of nodes than the DAFN method; in order to share replicas effectively, each group should be stable, so the DCG creates groups that are biconnected components. ▸ DCG

Biconnected component:

- denotes a maximum partial graph G which is connected (not divided) if an arbitrary node in the graph G is deleted.

Michal Kalewski Ad hoc networks [19/34]

Replica allocation methods

- Static Access Frequency (SAF): each node allocates replicas of data items in descending order of the access frequencies.
- Dynamic Access Frequency and Neighborhood (DAFN): the access frequency to each data item and the neighborhood among nodes are taken into account.
- Dynamic Connectivity based Grouping (DCG): shares replicas in larger groups of nodes than the DAFN method; in order to share replicas effectively, each group should be stable, so the DCG creates groups that are biconnected components. ▸ DCG

Biconnected component:

- denotes a maximum partial graph G which is connected (not divided) if an arbitrary node in the graph G is deleted.

Michal Kalewski Ad hoc networks [19/34]

Replica allocation (DCG)

- DCG protocol is executed at every *relocation period* and initially each node broadcasts its id and information on access frequencies to data items, this allows hosts to recognize the connected nodes.
- In each partition the node with the lowest id executes an algorithm to find biconnected components with the known network topology at the moment – each biconnected component is put to a *group*.
- Hosts that belongs to more than one biconnected component can only belong to one, i.e., to one which was found first.

Replica allocation (DCG)

- DCG protocol is executed at every *relocation period* and initially each node broadcasts its id and information on access frequencies to data items, this allows hosts to recognize the connected nodes.
- In each partition the node with the lowest id executes an algorithm to find biconnected components with the known network topology at the moment – each biconnected component is put to a *group*.
- Hosts that belongs to more than one biconnected component can only belong to one, i.e., to one which was found first.

Michal Kalewski Ad hoc networks [20/34]

Replica allocation (DCG)

- DCG protocol is executed at every *relocation period* and initially each node broadcasts its id and information on access frequencies to data items, this allows hosts to recognize the connected nodes.
- In each partition the node with the lowest id executes an algorithm to find biconnected components with the known network topology at the moment – each biconnected component is put to a *group*.
- Hosts that belongs to more than one biconnected component can only belong to one, i.e., to one which was found first.

Michal Kalewski Ad hoc networks [20/34]

Replica allocation (DCG)

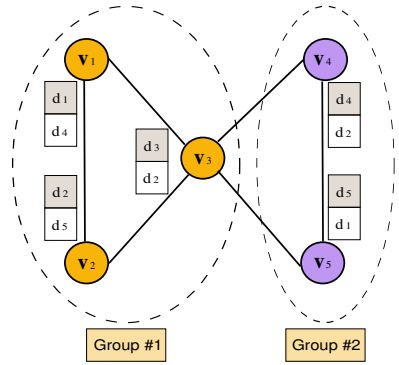
- Next the host with the lowest id in each group (*coordinator of the group*) calculates an access frequency of the group to each data item as a summation of access frequencies of all nodes in the group to the data.
- Then the coordinator determines replicas allocation in the group in the order of the access frequencies of the group – each replica is allocated at the host whose access to the corresponding data item is the highest (allowing hosts free memory space).■

Replica allocation (DCG)

- Next the host with the lowest id in each group (*coordinator of the group*) calculates an access frequency of the group to each data item as a summation of access frequencies of all nodes in the group to the data.
- Then the coordinator determines replicas allocation in the group in the order of the access frequencies of the group – each replica is allocated at the host whose access to the corresponding data item is the highest (allowing hosts free memory space).■

Replica allocation (DCG) – example

Data items	Nodes					Groups	
	v_1	v_2	v_3	v_4	v_5	#1	#2
d_1	0,6	0,2	0,4	0,3	0,5	1,2	0,8
d_2	0,4	0,6	0,5	0,3	0,2	1,5	0,5
d_3	0,3	0,4	0,5	0,2	0,1	1,2	0,3
d_4	0,3	0,1	0,2	0,6	0,4	0,6	1
d_5	0,4	0,5	0,3	0,2	0,7	1,2	0,9



Location management

The location management or location tracking problem in ad hoc networks is to gather information on the state and location of each node, i.e., to track the location of data items and replicas.

Location changes in MANETs are caused by replica allocation/relocation and unpredictable network topology changes.

The main purpose of location management approaches is to reduce the traffic – the number of broadcasts – for data requests.

AL GM

Location management (AL)

AL (*Access Log*) method – each node holds *access log* (AL) *table*, which consists of pairs

$$< AL.d_i, AL.list_i >,$$

where: $AL.d_i$ is data identifier (d_i) and $AL.list_i$ is a list of node identifiers (id) that holds the replica of data item corresponding to the data d_i.

Size of each $AL.list_i$ is limited to L.

Location management (AL)

(**Recording location information phase**)
When a node successfully accesses a data or its replica d_i on node v_i, then it inserts id of v_i at the top of the list corresponding to d_i in local AL table – if id of v_i already exists in the list, the old one is deleted.

At every relocation period each node clears its AL table.

(**Data access phase**)
Access requests to item d_j are unicast to the nodes according to the order in the $AL.list_j$ list. If all the requests fail, access request is broadcast in the network.■

Location management (GM)

GM (*Group Management*) method – (approach for DCG method) each node holds a LM table and *gateway* (GW) *information table*, which consists of node identifies (id) that connects to at least one node belonging to other groups (that is to other biconnected components).

Gateway:

- node that connects to at least one node belonging to other groups (that is to other biconnected components).

Location management (GM)

(**Recording location information phase**)
After DCG method execution, each coordinator of the group notifies all nodes in the group about the information on

- location of data items and replicas (its recorded as LM table) and
- gateway nodes (its recorded as GW table) in the group.

Node identifiers are inserted in both tables $LM.list_i$ (if a data item or its replica is allocated to more than one node) and GW in descending order of the hop count – this is possible because of node identifiers broadcast at the beginning of the DCG method.

At every relocation period all information in the LM and GW tables that have been recorded before current period is discarded.

Location management (GM)

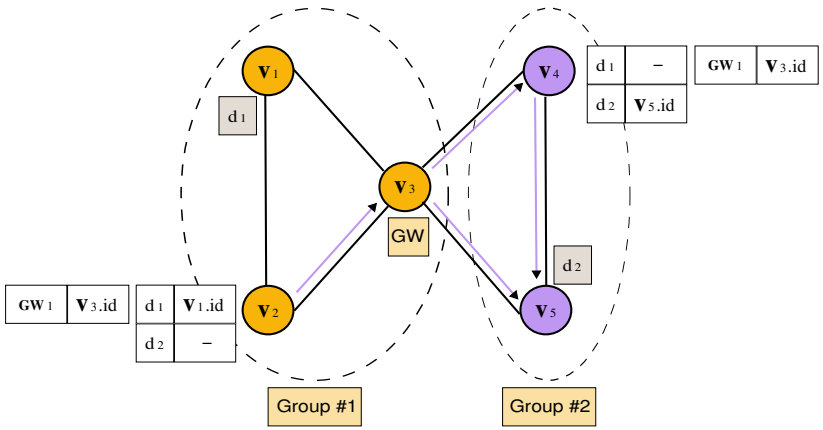
(**Data access phase**)
Access requests to item d_j are unicast to the nodes according to the order in the $LM.list_j$ list.

If all the requests fail, the node unicasts the request to gateway nodes until the request succeeds.

Gateway nodes forward the received request messages to neighboring nodes in other groups, which execute the same procedure in a group using local LM and GW tables.

Finally, if all previous requests fail, the node broadcasts the request in the network.■

Location management (GM) – example



Consistency management

Earlier solutions that do consider network partitioning approach the problem from the direction of replica consistency in distributed systems and databases; while the problems are similar, several instances of information dissemination in ad hoc networks are different.

Strong consistency may result in MANETs in poor availability if the presence of frequent network partitioning; therefore, weaker consistency levels have been proposed to increase the availability.

Consistency management (LOC & GOC)

Local observation consistent (LOC)
$\forall_x \forall_{x_n \in Copies(x)}$:

- (C1) x_n will eventually converge to the most recently propagated state of x;
- (C2) once x_n has reached state x_j^k, it will no longer accept state x_j^l with $l < k$.

Global observation consistent (GOC)

- (C3) once x_n has reached state x_i^k, it will no longer accept x_j^l state with $x_j^l \rightarrow x_i^k$

Consistency management (broadcast methods)

- Update Broadcast Method (UBM): a node holding an original data item broadcast an invalidation report to connected hosts every time it updates the data item.
- Connection Rebroadcast Method (CRM): similar to UBM, but in addition, every time two nodes newly connect with each other, they rebroadcast invalidation reports that they have already receive.

Consistency management (broadcast methods)

- Update Broadcast Method (UBM): a node holding an original data item broadcast an invalidation report to connected hosts every time it updates the data item.
- Connection Rebroadcast Method (CRM): similar to UBM, but in addition, every time two nodes newly connect with each other, they rebroadcast invalidation reports that they have already receive.

Self-stabilization

Self-stabilization is amenable to the layered approach.

We could denote that if S is self-stabilizing with respect to P then $TRUE \dot{=} P$ in S.
The relation $\dot{=}$ is transitive, if $Q \dot{=} P$ and $P \dot{=} R$, then $Q \dot{=} R$.

Informally we can see how transitivity corresponds to the technique of layering:

- given S_1 satisfying $Q \dot{=} P$ and S_2 satisfying $P \dot{=} R$, we combine S_1 and S_2 such that S_2 reads from the variables of S_1 to produce a new program satisfying $Q \dot{=} R$.

Most interesting self-stabilizing algorithms for MANETs: mutual exclusion, spanning tree construction and other graph theory problems.

Conclusions

- Failure detectors and gossiping.
- Replica allocation: different groups (DCG – biconnected component).
- Group communication: i.e., efficient casual order.
- Self-stabilization: layered approach.