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

Jerzy Brzeziński Michał Kalewski Marek Libuda

Poznań University of Technology, Institute of Computing Science
ul. Piotrowo 2, 60-965 Poznań, Poland
e-mail: Jerzy.Brzezinski@cs.put.poznan.pl, Michal.Kalewski@cs.put.poznan.pl, Marek.Libuda@cs.put.poznan.pl

$Id: ad_hoc_algorithms.lyx,v 1.12 2006/04/28 12:14:22 mkalewski Exp $

Abstract

Distributed ad hoc systems are a fast growing and promising research area in computing science with multiple new applications for dynamic and mobile environments. However, constructing systems for such environments require new algorithmic solutions. This paper presents an overview of basic problems and algorithms for ad hoc networks. First, the formal model of ad hoc system is shown, then the paper discusses the following topics: routing, replication and consistency with replica allocation, location management and group communication. Finally, the paper presents some further lines of investigation.

Key words: distributed ad hoc systems, routing, replication, group communication.

Contents

1 Introduction

Mobile ad hoc networks (MANETs) [Agg04, DHL+02, Per00, Toh01] 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 (Figure 1). If

*This work has been partially supported by the State Committee for Scientific Research grant no. 3 T11C 073 28.

hosts in MANET function as both: a computing host and a router, they form a *multiple hop ad hoc network*. 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 a highly unpredictable manner. Mobile hosts in an 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).

Fast development and spread of ad hoc networks is possible, among other things, by deployment of IEEE 801.15 (Bluetooth) and 801.11 wireless networks standards (with the ad hoc mode) [Tan02], third-generation cellular networks [HS03, PSS03], and commonly use of mobile devices like *personal digital assistant* (PDA) or laptops. MANETs find application in the following areas:

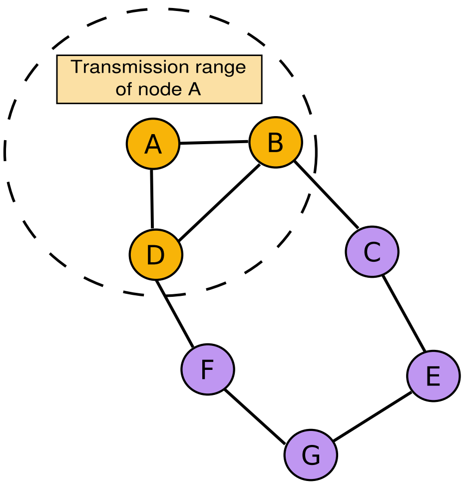
- civil, e.g. rescue workers at an earthquake that destroyed the infrastructure or a gathering of people with laptops lacking 801.11 (an example could be MIT roofnet project, an experimental network in which campus students deploy roof-mounted radio antennas for sharing of Internet access[1]);
- industrial, e.g. *sensor networks* – a number of sensors spread across a geographical area [Cal03, GZ04] (an example could be a sensor network in a water treatment plant for monitoring and control of process instruments deployed by Ember Corporation[2]);
- military, e.g. vehicles on battlefield or a fleet of ships at sea (an example could be *Mercury Wideband Network Radio* (MWNR), which is a military distributed tactical network[3]).

Other interesting examples of ad hoc systems implementation can be found in [Gac04] and [Mos04].

Moreover, ad hoc systems allow to build a new Internet service – *nomadic computing*, defined as a rich set of computing capabilities for nomadic or mobile users as they move from place to place in a transparent, integrated, adaptive and convenient way [Kle03]. Ad hoc networks are easy to speed deployment and allow to decrease dependence on infrastructure [Vai00].

MANETs and the expectations created by advancement of mobile devices bring many technological and algorithmic challenges that can not be dealt with, as it has been done in traditionally wired networks and distributed systems. The most important challenges are a dynamic nature of topology with frequent disconnections and complete distribution of control; other – also important – issues attached with ad hoc systems are: low battery life, greater security vulnerabilities and limited memory and computational resources.

Figure 1: An example of an ad hoc network.



The goal of this work is to systematize basic and current interesting algorithmic problems in ad hoc distributed systems. This paper presents the up-to-date state of art of main distributed algorithms

[1]Internet: `http://pdos.csail.mit.edu/roofnet/`
[2]Internet: `http://www.ember.com/`
[3]Internet: `http://acd.itt.com/tactical_networking.htm`

and points out some new interesting research directions. It is organized as follows: first, the model of ad hoc systems is defined in Sec. 2. The routing protocols for MANETs are presented in Sec. 3; Sec. 4 discusses replication in ad hoc systems with replica allocation/relocation (Sec. 4.1), location management (Sec. 4.2) and consistency management (Sec. 4.3). Deterministic and probabilistic group communication is shown in Sec. 5 and finally the paper is shortly concluded in Sec. 6 with some new research directions.

2 Ad hoc system model

As mentioned above, we define MANET as a network with the following properties: (i) self-organization – distributed control of all operations, (ii) no fixed infrastructure, (iii) dynamic topology with partitioning, (iv) limited bandwidth, memory and other computational resources, (v) limited security [Gac04]. Consequently, the topology of the ad hoc network is modeled by an 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 the nodes that can communicate directly). Since the nodes are mobile, the E set changes with time. The links between two adjacent nodes are always bi-directional and every node $v \in V$ has a unique *physical address* or *id*. The following properties of graph G are defined:

- G_P: graph G can get partitioned[4] and reconnected;
- G_{NP}: graph G does not get disconnected (partitioned) but nodes are still mobile;
- V_C: the V set can change in time – that is the number of nodes that take part in processing can change in time;
- V_{NC}: the V set does not change in time – that is the number of nodes that take part in processing is constant.

Finally we receive four graphs with different assumptions: $G_{C|P}$ (a graph can get partitioned and the number of nodes can change) – this is the most fluent model, $G_{NC|P}$ (a graph can get partitioned but the number of nodes is constant), $G_{C|NP}$ (a graph does not get disconnected but the number of nodes can change) and $G_{NC|NP}$ (a graph does not get disconnected and the number of nodes is constant) – this is the most stable one. V_C property means that every node can *stop* its processing (without warnings and at any time) and never go back to the system. On the other hand, V_{NC} property means that any node will not *stop* its processing – however, it is still possible that any node will be temporary *unreachable* due to the partitioning (with G_P).

This generic model can be extended with others and more specific parameters and assumptions, like e.g. an existence of *Global Positioning System* (GPS) [IMTW02], synchronous communication [BGP04] or *Geosensor* component that maintains current location of each mobile node and current real time [DGL+04]. Many algorithms designed for ad hoc systems in general, allow to be used in $G_{C|P}$ model, but when the system executes particular algorithm's operations then the connections (set E) – or part of the connections – ought to stay stable to correct finish computation.

Other interesting model of ad hoc network based on a pathset algebra is developed in [Pat01] – the author uses pathsets to model paths in the network, modal algebra for modeling a multi-agent system (the applications, actions and services) and operation analogous to the scalar multiplication as a function that couples the two specifying algebras. There are also available mobility models for ad hoc networks, useful for MANETs simulations: e.g., [BCD02, LNR04].

3 Routing

MANETs may employ traditional TCP/IP protocols to provide communication between nodes and to create multiple hop ad hoc networks but, due to the properties of such networks, each layer in a

[4] Partitioning fragments the graph into isolated subgraphs called partitions – there is a path in E for any two nodes in the same partition but there is not a path in E for any two nodes in different partitions.

TCP/IP model may require modifications – especially routing protocols. 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 possibly indefinitely. With MANETs frequent topology changes caused by nodes' mobility involve spontaneous changes of paths' validity. That makes routing in ad hoc networks different from routing in their fixed counterparts and creates a challenging problem, which has received a lot of attention from researches (as pointed out in [Tan02] "ad hoc routing is a red-hot research area").

Routing protocols for ad hoc networks are classified into three different groups: *proactive* (or *global*), *reactive* (or *on-demand*) and *hybrid* [ADW04]. In proactive routing algorithms, nodes predetermine the routes to all destinations at the startup and then they respond to changes in the network to state consistent network view (an update process may be also be periodic). Proactive protocols may consume considerable amount of bandwidth, as each node must maintain full routing information, so the aim of reactive algorithms is to reduce the overheads by determining routes for nodes that require sending data to a particular destination (on demand). A route discovery process in reactive protocols usually uses flooding special *route request* packets through the network. The route information is sent back (using *link reversal*) when the destination – or a node with a route to the destination – is reached or when all the possible route permutations have been examined. Packets in reactive protocols may carry a complete source to destination addresses (*reactive source routing*) or only destination and source addresses (*reactive point-to-point routing*). Hybrid routing algorithms combine proactive and reactive approaches to increase scalability – the nodes with close proximity may co-operate with each other to provide some sort of backbone (with proactive approach) to reduce the route discovery process (reactive approach) overheads for far away nodes. Examples of routing protocols are shown in Table 1, reviews of ad hoc routing protocols are presented in [ADW04, HP04, RT99].

Table 1: Examples of routing protocols (based on [ADW04]).

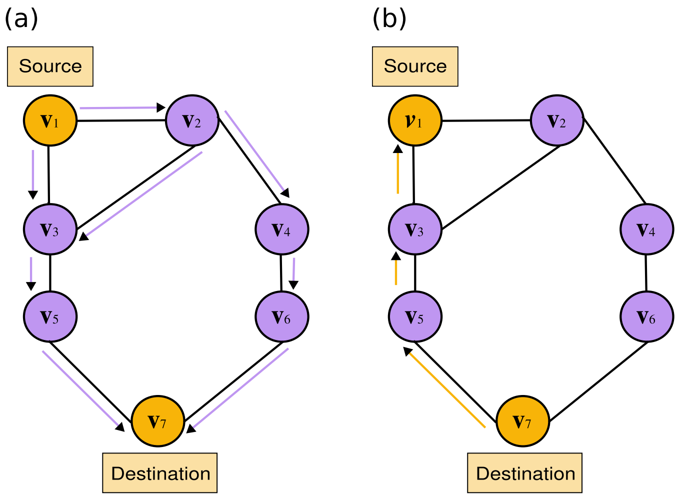
Proactive protocols	Reactive protocols	Hybrid protocols
Destination-sequenced distance vector (DSDV)	Ad hoc on-demand distance vector protocol (AODV)	Zone routing protocol (ZRP)
Cluster-head gateway switch routing (CGSR)	Cluster-based routing protocol (CBRP)	Zone-based hierarchical link state (ZHLS)
Wireless routing protocol (WRP)	Dynamic source routing (DSR)	Distributed spanning trees based routing protocol (DST)
Global state routing (GSR)	Routing on-demand acyclicmulti-path (ROAM)	Distributed dynamic routing (DDR)
Fisheye state routing (FSR)	Light-weight mobile routing (LMR),	Scalable location update routing protocol (SLURP)
Source-tree adaptive routing (STAR)	Temporally ordered routing algorithm (TORA)	
Distance routing algorithm for mobility(DREAM)	Associativity-base routing (ABR)	
	Signal stability adaptive (SSA)	

One of most popular routing protocols, is reactive *Ad Hoc On-Demand Distance Vector Routing* (AODV) [PR99a], described in RFC 3561[5]. AODV builds routes using a route request and route reply messages query cycle. When a node wants to send a packet to a destination for which it does not have a route, it broadcasts a route request message (*RREQ*) across the network. The *RREQ* message contains several key bits of information: source address (*RREQ.src_addr*), request identifier (*RREQ.req_id*) – incremented whenever a *RREQ* is broadcast, destination address (*RREQ.dst_addr*), the most recent of destination's sequence value that source is aware of (*RREQ.dst_seq*) and hop counter (*RREQ.hop_count*). Each node receiving a route request message checks if this request has already been processed (by *RREQ.src_addr* and *RREQ.req_id*), if this is the case, it is discarded. Otherwise,

[5] Available at http://www.ietf.org/rfc/rfc3561.txt

the node may send – unicast – a route replay message (*RREP*) if it is either the destination or if it has a route to the destination with a corresponding sequence number greater than or equal to that contained in the *RREQ* message (*RREQ.dst_seq*). Else, it stores data from the message as a new entry in local *reverse route table* (a timer is also started for newly-made route reverse entry) and rebroadcasts the *RREQ*. As the *RREP* propagates back to the source using reverse route table, the nodes update local routing table if one or more of the following conditions are met: (i) no route to the destination is known, (ii) the sequence number for the destination in the *RREP* is greater than the value in the routing table, (iii) the sequence numbers are equal but the new route is shorter (*RREP* also contains hop counter *RREP.hop_count*). Periodically, each node broadcasts a *Hello* message for route maintenance, if a link break occurs while the route is *active* (that is a route entry in the routing table marked as valid), the node upstream of the break propagates a route error message (*RERR*) to the source node with information about the new unreachable destination. Figure 2 presents AODV route discovery process.

Figure 2: AODV route discovery (a) propagation of *route request* (b) *route reply.*



The original AODV protocol has actually many implementations and extensions like IETF Internet drafts: *Secure AODV*[6] (SAODV) [Zap02], *Multicast AODV*[7] (MAODV) [PR99b] and *AODV for IPv6*[8] [MNP+02].

4 Replication and consistency

Since in ad hoc networks, the hosts move freely, disconnections may occur frequently and this may cause frequent network partitioning. Consequently, data and services accessibility in MANETs is 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 a 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. 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: (i) *replica allocation/relocation*, (ii) *location management* and (iii) *consistency management.*

[6] Available at http://www.cs.ucsb.edu/~ebelding/txt/saodv.txt
[7] Available at http://people.nokia.net/charliep/txt/aodvid/maodvid.txt
[8] Available at http://www.cs.ucsb.edu/~ebelding/txt/aodv6.txt

4.1 Replica allocation/relocation

Replica allocation and relocation in the field of fixed networks and distributed systems has been an extensive research topic. It determines how many replicas of each data item are created and to which nodes and how these replicas are allocated. In fixed networks, the 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 [JW92]. The communication cost is the average number of messages required for a read or write of the data item and the communication time of a 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 [JW92]. The problem of finding optimal allocation scheme has been proved to be NP-complete (for different cost models) for both general static distributed systems [MW91, CPP02] and ad hoc networks [JKXY04]. 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 a network topology.

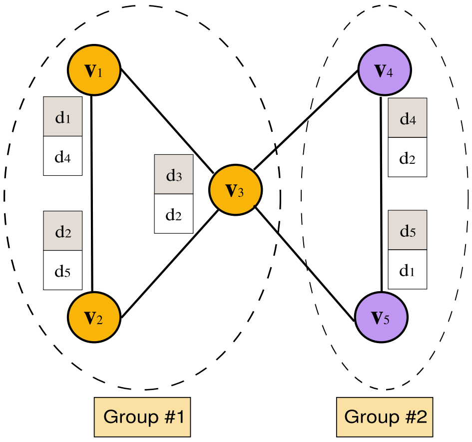
Several strategies have been proposed for data replication schemes and replica allocation in MANETs [Har01, Har03, HM04, HMN04, IMTW02, JKXY04, LW02]. In [Har01], three heuristic methods for replica allocation have been proposed. In all three proposed heuristics, replicas are relocated in a specific period (called *relocation period*) and replica allocation is determined based on the access frequency from each node to each data item and additionally in the two methods (DAFN and DCG) the network topology at the moment. First method, called *Static Access Frequency* (SAF), allocates replicas of data in descending order of the access frequencies from each node (within the limit of its own memory space). Since the nodes do not need to exchange any information with each other, SAF method allocates replicas with low traffic but when hosts have similar access characteristics, it excesses duplicates data item among them. To solve the problem of many replica duplication in SAF, the second method *Dynamic Access Frequency and Neighborhood* (DAFN), eliminates the replica duplication among directly connected nodes. If there is a replica duplication of data item between two neighboring nodes, the node with the lower access frequency (relative to the data item) changes the replica to a replica of different data item. It is accomplished by a repeated procedure of the breadth first search (from the host with the lowest *id*) in each set of nodes which are connected to each other. The last proposed method, *Dynamic Connectivity based Grouping* (DCG), shares replicas in larger groups of mobile nodes than the DAFN. The groups used for replica allocation in MANETs should be stable, that is e.g., the group is not easily partitioned due to changes of network topology. DCG method uses *biconnected components* [AHU74] to create groups of nodes in MANET. A biconnected component is defined here as a maximal partial graph G' of G which cannot be divided by deleting any vertex $v \in G$. DCG protocol is executed at every relocation period and initially each node broadcasts its *id* and information on access frequencies to data items, which 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 into a *group*. Hosts that belong to more than one biconnected component can only belong to one, e.g. to one which was found first. 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 (Table 2 shows an example). 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). Table 2 and Figure 3 show an example of executing the DCG protocol (rectangles denote original data and replicas allocated at nodes); in the presented case each node v_i ($i = 1, ..., 5$) holds data d_i as an original and has a capability to store two data items.

The methods SAF, DAFN and DCG are extended in [Har03, HM04, HHN04, HMN04] to perform probabilistic analysis of the update intervals and correlation among data items to increase accessibility and the number of successful access requests. In [JKXY04] a dynamic and adaptive replica allocation algorithm is proposed, which minimizes the *communication cost* (hops between two nodes) of object access.

Table 2: Access frequencies to data items.

Data items	Nodes					Groups	
	v_1	v_2	v_3	v_4	v_5	Group #1	Group #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

Figure 3: An example of executing the DCG protocol in the environment given by table 2.



4.2 Replica 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. As pointed out above, location changes in MANETs are caused by replica allocation/relocation and unpredictable network topology changes. Therefore, dedicated techniques to manage location of replicas to efficiently forward access requests in such a dynamic environment have been proposed in the literature [Har05, LVW03]. The main purpose of the approaches proposed in [Har05], is to reduce the traffic – the number of broadcasts – for data requests. Some of these methods are specialized for the replica allocation heuristics presented in section 4.1 (that is for DAFN and DCG) and one method (AL) is more generic and can be adopted for any replica allocation methods (with relocation periods). There are five methods and all of them can be explained in the two clauses: *recording location information* and *data access* as follows:

- *Access Log* (AL) 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. The size of each $AL.list_i$ is limited to L. (**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, an access request is broadcast in the network.

- *Neighborhood Management* (NM) method – (approach for DAFN method) each node holds *location management* (LM) *table* similar to the AL table. (**Recording location information phase**) At every relocation period, all information in the LM table that has been recorded before the current period is discarded. Next, after DAFN method execution, all neighboring hosts record the location of data items and replicas held by them in their local LM tables – so, each node knows the location of data items and replicas held by neighbors at the moment. (**Data access phase**) Similar to the AL data access phase.

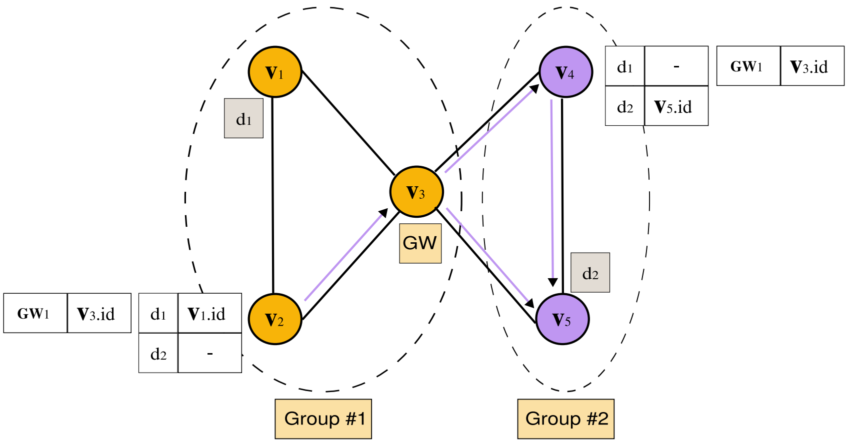
- *Group Management* (GM) method – (approach for DCG method) each node holds a LM table and a *gateway* (GW) *information table*, which consists of node identifiers (*id*) that connects to at least one node belonging to other groups (that is to other biconnected components). (**Recording location information phase**) After DCG method execution, each coordinator of the group notifies all nodes in the group about the information on (i) location of data items and replicas (its recorded as LM table) and (ii) 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 a 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 has been recorded before the current period is discarded. (**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 the previous requests fail, the node broadcasts the request in the network.

  Figure 4 presents an example of execution of the GM location method – here mobile host v_2 requests data item d_2, which is located at the node v_5 in other group. So, in this case, node v_2 uses gateway node (v_3) to forward its access request into group #2.

- AL-MN method – hybrid of the AL and NM methods. (**Recording location information phase**) The AL table is updated in the same way as in the AL method, but at every relocation period, all information in the AL table that has been recorded before the current period is discarded and the AL table is updated in the same way as LM table in the NM method. (**Data access phase**) Similar to the AL data access phase.

- AL-GM method – hybrid of the AL and GM methods. (**Recording location information phase**) The AL table is updated in the same way as in the AL method. At every relocation period, all information in the AL table and GW table that has been recorded before current period is discarded and both tables are updated in the same way as in the GM method. (**Data access phase**) Similar to the GM data access phase.

Figure 4: Data access in the GM method (intergroup communication).



Different approach to the location management problem is presented in [LVW03] – this solution uses a *biquorum system*, that is a generalization of the traditional quorum system. Here a dynamic quorum construction scheme is proposed, which uses a heuristic that keeps track of the unreachable nodes when quorum is constructed. Replicas are placed and located by performing operations on the subset (quorum) of all nodes in MANET.

4.3 Consistency management

The topology of ad hoc systems can change with time and get partitioned and reconnected – this creates a problem of maintaining consistency if a replication scheme is used in the system, particularly when partitioning occurs. In such a case, availability and correctness can simply be ensured – availability by allowing all nodes to process computation event when partitioning occurs (but this may compromise correctness) and correctness by suspending operation in all but one partition and forwarding updates at reconnect (but such a approach compromises availability). Generally, the strategies that have been proposed for consistency management in partitioned networks can be classified into four classes [DGMS85]: (i) *pessimistic strategies* – preserve consistency by limiting availability; (ii) *optimistic strategies* – do not restrict availability; (iii) *syntactic approaches* – use one-copy serializability[9] as a criterion for consistency correctness; (iv) *semantic approaches* – use the semantic of data items (or transactions) in defining correctness criterions. This classification and approach examples are shown in Table 3.

Table 3: Classification of strategies for consistency management in partitioned networks (based on [DGMS85]).

	Syntactic Approaches	Semantic Approaches
Optimistic Strategies	Version Vectors The Optimistic Protocol	Log Transformations Data-Patch
Pessimistic Strategies	Primary Site Copy Tokens Voting Missing Writes Accessible Copies Algorithm Class Conflict Analysis	General Quorum Consensus

Examples of optimistic and weak consistency protocols designed for MANETs are *Local Observation Consistency* (LOC) and *Global Observation Consistency* (GOC) [BHR02]. The replication scheme in ad hoc system is defined to be *local observation consistent* if for each original data item d:

$\forall_{d_i \in Copies(d)}$:
(C1) d_i will eventually converge to the most recently propagated state of d;
(C2) once d_i has reached state d_j^l, it will no longer accept a state d_j^k with $k < l$.

Where $Copies(d)$ is a set of all replicas of d, and k and l indicates succeeding states of the same replica. Conditions (C1) and (C2) of LOC ensure that all replicas of any data item d converge into most recently propagated state and that a propagated state of a given replica is never overwritten by an older propagated state of the same replica. The replication scheme in the ad hoc system is defined to be *global observation consistent* if it is local observation consistent and additionally for each original data item d:

$\forall_{d_i \in Copies(d)}$:
(C3) once d_i has reached state d_m^k, it will no longer accept a state d_n^l if $d_n^l <_x d_m^k$ and for any m and n.

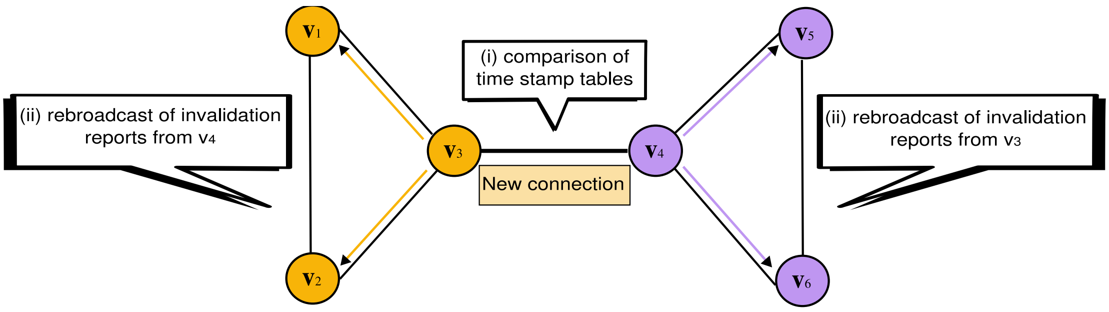
Where the relation $<_x$ is originally defined as follows: e *occurs before* e' – that is expressed as $e <_x e'$ – if e was finished before e' was started. Here, the major components of the system are *perceivable objects*, *observers* and nodes. Observers can sense perceivable objects that reside in its communication range and propagate (propagation is based on flooding mechanism) states of that objects – each node holds its own replicas of perceivable objects. To prevent that concurrent observers are propagating the

[9] One-copy serializability is originally defined as follows: the concurrent execution of operations on replicated data items is equivalent to some serial execution on non-replicated data item.

same state of an object and to cope with requirements of the $<_x$ relation a randomization technique is used – each observer waits random time before it starts to propagate a new state of an object.

On the other hand, a consistency protocol may, instead of update propagation, transmit invalidation messages. Examples of such an approach are: *Update Broadcast Method* (UBM) and *Connection Rebroadcast Method* (CRM) [HHN03]. It's assumed in these methods that each data item is updated only by the node that holds the original and after updating, the node broadcasts *invalidation report* that cancels all other replicas. In UBM invalidation report ($ir(d_i, t_s)$) includes d_i – data item identifier, and t_s – update time (time stamp). When a node receives an $ir(d_i, t_s)$, it refers to its own *time stamp table*: if t_s in ir is greater than t_s in time stamp table for replica of d_i, then the node discards a replica from its own memory, updates the time stamp in the time stamp table and rebroadcasts $ir(d_i, t_s)$. Else, if t_s in ir is identical to the one in the local table, the node discards the report and does not broadcast it. CRM is similar to the UBM but in addition, every time two nodes newly connect with each other, they compare its own time stamp tables and rebroadcast invalidation reports that they have already received. If v_i and v_j are newly connected nodes then the node with greater id transmits its own time stamp table to the other one, say v_i. Next, v_i compares the local time stamp table with the received table and updates its own. After that, v_i broadcasts invalidation reports for each data item whose t_s in local table are smaller than the ones held by v_j to its connected nodes, but except to the v_j. For each data item whose t_s held by v_j are smaller than the ones in local table, v_i sends information on update time stamps to the v_j. The node v_j broadcasts invalidation reports for these data items to its neighbors, except v_i. Figure 5 presents an example of executing CRM.

Figure 5: Example of Connection Rebroadcast Method.



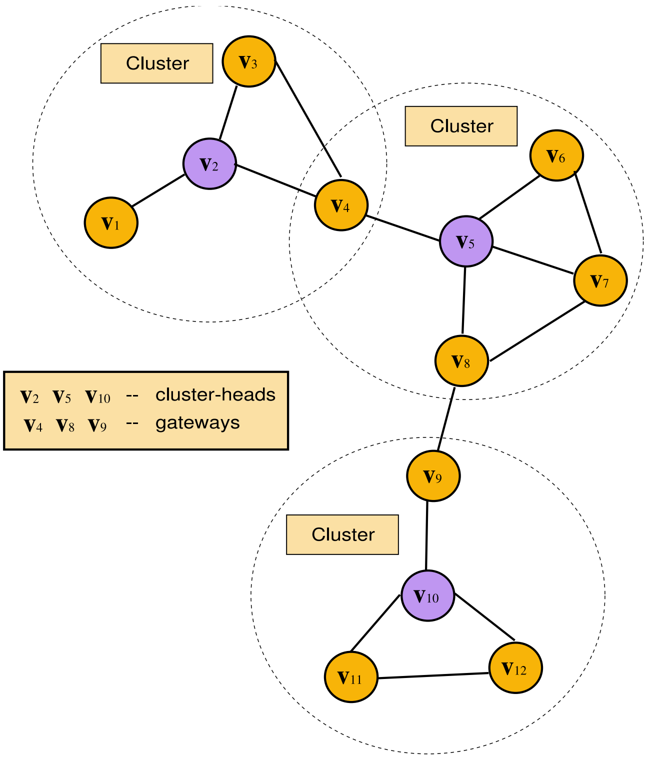
Other approach to consistency problem in mobile distributed computing is shown in [GH02] – here operational transformations are used to define a consistency criteria that lies between traditional strong and week criteria. Operational transformations consist in transforming operations to execute them at a different point in the history, that is to include or exclude the effect of other operations.

5 Group communication

Broadcasts and multicasts are a basic communication requirement to construct other and more advanced distributed algorithms. However, such a group communication should provide some delivery (reliability) guarantee beyond best-effort. Again, MANETs present a different network model with topology reconfiguration and partitioning from static distributed systems. Thus, generally current approaches to reliable group communication in ad hoc systems can be divided into two strategies [EV03]: (i) deterministic protocols, which provide strong delivery guarantees ("all or nothing") to a group or all nodes and (ii) probabilistic protocols, which provide guarantee delivery with a certain probability.

The first deterministic reliable broadcast algorithm introduced explicitly for ad hoc networks was *Reliable Broadcast* (RB) [PR97]. The RB algorithm provides *exactly once semantic* for messages delivery and tolerances nodes mobility, disconnections and network partitioning. So, a reliable broadcast here [PR97] is originally defined as follows: given a set V of N nodes in which each node at any time can generate a broadcast message, messages are delivered by the nodes so that the following properties are guaranteed: (i) if a node broadcasts a message m then all nodes in V will eventually deliver m and (ii) for each message m, every node delivers m at most once, and only if m has been broadcast by some node. To guarantee the protocol termination, the algorithm requires that the network partitions are temporary by nature. The RB protocol assumes the existence of a clustering algorithm – all nodes are grouped into *clusters* and a clustering algorithm is supposed to recognize the set of interconnected clusters. There are three types of nodes in each cluster: *cluster-head* (coordinates the transmission within the cluster), *gateway* (a node that is able to communicate between different clusters) and ordinary cluster member; an example of such an organization of an ad hoc network is presented in Figure 6. A node that wishes to broadcast a message m_b (each message is uniquely identified by the pair: node *id* and *sequence number* $< v_i.id$, $seq\_id >$) sends it to the cluster-head of the cluster it is currently in – the sending node remains blocked only the time needed to know that the message has been correctly received by its cluster-head. Next, the cluster-head resend the message m_b to all the nodes in a local cluster and waits for acknowledgement messages m_b^{ack} from all cluster members. The nodes that act as gateways forward the message m_b onto other clusters (by addressing the relevant cluster-head) and delay sending of the m_b^{ack} message until they receive acknowledgements from other clusters. This, in turn, could involve a recursive communication by propagation a message to other clusters. As soon as cluster-head receives all the local m_b^{ack} messages, it prepares cluster acknowledgement message $m_b^{cluster\,ack}$ that must be send to the originator cluster-head. If a gateway node, in touch with the predecessor cluster-head, is still active, the message $m_b^{cluster\,ack}$ is sent to this node. To deal with node mobility, the algorithm will switch to flooding acknowledgements back to the cluster-head of the cluster with the origination node.

Figure 6: An example of clustered ad hoc network.



Probabilistic protocols are not as precise as deterministic but they typically have less restrictive assumptions and reduced overhead. An example of such an algorithm is probabilistic multicast protocol *Anonymous Gossiping* (AG) [CRB01]. AG is added to MAODV but should also work with any on-demand routing protocol. *Gossiping* here is a technique of addition, that is outside the normal delivery, exchange of messages to increase the reliability of the system. AG involves two concurrent phases: (i) in the first one, a suitable primary multicast protocol is used to unreliably multicast the message m and (ii) in second phase, gossip technique is used to recover missed messages m from other members of the group that might receive it. The second phase consists of periodic rounds of the following steps (*gossiping*):

1. Node v_i randomly chooses other member of the group, say v_j.

2. v_i sends v_j a message history, that is the information about messages it has received (or not received).

3. v_j checks to verify if it has received any of the messages listed by v_i.

4. v_i and v_j could exchange messages which are not a part of each other's message history.

In the AG algorithm, nodes are randomly selected out of all neighbors (i.e., in the mulsticast tree of MAODV [PR99b] protocol) to send a *gossip invitation message* – so the nodes do not need to know the membership of their group. If the receiving node is a member of the multicast group then it decides randomly to accept invitation or redirect it forward accordingly to the same procedure. Next, the accepting node unicasts a *gossip reply message* to the initiator of the invitation to start gossiping.

Group communication in the case of MANETs is also studied in [GS99, GPSS02, GLOT02] (deterministic algorithms) and [ELH03] (gossip protocol).

6 Summary

Mobile ad hoc networks form distributed systems with arbitrary topology, where all nodes are free to move (in highly unpredictable manner) and arrange themselves as required. So, creating systems for such dynamic environments require new algorithmic solutions. In this paper, we have presented a short survey of basic algorithmic problems in distributed ad hoc systems: routing protocols, replications with replica allocation, location and consistency management, deterministic and probabilistic group communications. For each discussed problem, we have also shown suitable solutions and algorithms. There are also other and more complex algorithmic issues in MANETs, like self-stabilization (e.g. [GS03]), mutual exclusion (e.g. [CW02]), failure detectors (e.g. [KW03]), efficient message ordering in group communication (e.g. [DK96]) or new synchronization problems (e.g. [BGP04]) – but the presentation of these problems requires separate study and it would be interesting to investigate these issues further.

References

[ADW04] M. Abolhasan, E. Dutkiewicz, T. A. Wysocki. A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks*, 2(1):1–22, 2004. Cited on page(s): 4

[Agg04] G. Aggelou. *Mobile ad hoc networks: from wireless lans to 4g networks*. McGraw-Hill Professional, 2004. Cited on page(s): 1

[AHU74] A. V. Aho, J. E. Hopcroft, J. D. Ullman. *The design and analysis of computer algorithms*. Addison Wesley, 1974. Cited on page(s): 6

[BCD02] J. Boleng, T. Camp, V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2(5):483–502, 2002. Cited on page(s): 3

[BGP04] S. Baehni, R. Guerraoui, B. Pochon. The driving philosophers. *Proceedings of the 3rd IFIP International Conference on Theoretical Computer Science*, 2004. Cited on page(s): 3, 12

[BHR02] Ch. Becker, J. Hähner, K. Rothermel. Consistent update diffusion in mobile ad hoc networks. Raport instytutowy 2002/04, University of Stuttgart : Special Research Area SFB 627 (Nexus: World Models for Mobile Context-Based Systems), University of Stuttgart, Institute of Parallel and Distributed High-Performance Systems, Distributed Systems, July 2002. Cited on page(s): 9

[Cal03] E. H. Callaway. *Wireless sensor networks: architectures and protocols*. Auerbah Publications, 2003. Cited on page(s): 2

[CPP02] S. A. Cook, J. K. Pachl, I.S. Pressman. The optimal location of replicas in a network using a read-one-write-all policy. *Distrib. Comput.*, 15(1):57–66, 2002. Cited on page(s): 6

[CRB01] R. Chandra, V. Ramasubramanian, K. P. Birman. Anonymous gossip: improving multicast reliability in mobile ad-hoc networks. *Proceedings of the 21st International Conference on Distributed Computing Systems (ICDCS)*, strony 275–283, 2001. Cited on page(s): 11

[CW02] Y. Chen, J. L. Welch. Self-stabilizing mutual exclusion using tokens in mobile ad hoc networks. *DIALM '02: Proceedings of the 6th international workshop on Discrete algorithms and methods for mobile computing and communications*, strony 34–42, New York, NY, USA, 2002. ACM Press. Cited on page(s): 12

[DGL+04] S. Dolev, S. Gilbert, A. N. Lynch, E. Schiller, A. Shvartsman, J. Welch. Virtual mobile nodes for mobile ad hoc networks. *International Conference on Principles of DIStributed Computing*, strony 230–244, 2004. Cited on page(s): 3

[DGMS85] S. B. Davidson, H. Garcia-Molina, D. Skeen. Consistency in partitioned networks. *ACM Computing Surveys*, 17(3):341–370, 1985. Cited on page(s): 9

[DHL+02] J. Deng, Z. Haas, B. Liang, P. Papadimitratos, S. Sajama. Wireless ad hoc networks. Z. J. Haas and J. Deng and B. Liang and P. Papadimitatos and S. Sajama. Wireless ad hoc networks. In John Proakis, editor, Encyclopedia of Telecommunications (to appear). John Wiley, December 2002, 2002. Cited on page(s): 1

[DK96] D. Dolev, I. Keidar. Efficient message ordering in dynamic networks. *PODC '96: Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing*, strony 68–76, New York, NY, USA, 1996. ACM Press. Cited on page(s): 12

[ELH03] P. Th. Eugster, Jun Luo, J.-P. Hubaux. Route driven gossip: probabilistic reliable multicast in ad hoc networks. *Proceedings IEEE INFOCOM 2003, The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, 2003. Cited on page(s): 12

[EV03] P. Ezhilchelvan, E. Vollset. A survey of reliable broadcast protocols for mobile ad-hoc networks. Raport instytutowy CS-TR-792, University of Newcastle upon Tyne, 2003. Cited on page(s): 10

[Gac04] J. Gacnik. Technologies and sucessful applications for direct and multihop ad hoc networks. Seminar on new network and information technologies and infrastructures, Institute for Information Systems Research, University of Hannover, Germany, 2004. Cited on page(s): 2, 3

[GH02] R. Guerraoui, C. Hari. On the consistency problem in mobile distributed computing. *POMC*, strony 51–57. ACM, 2002. Cited on page(s): 10

[GLOT02] M. Gerla, S.-J. Lee, K. Obraczka, K. Tang. Congestion controlled adaptive lightweight multicast in wireless mobile ad hoc networks. *ISCC '02: Proceedings of the Seventh International Symposium on Computers and Communications (ISCC'02)*, strony 967–972, Washington, DC, USA, 2002. IEEE Computer Society. Cited on page(s): 12

[GPSS02] T. Gopalsamy, D. Panda, P. Sadayappan, M. Singhal. A reliable multicast algorithm for mobile ad hoc networks. *ICDCS '02: Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)*, strony 563–570, Washington, DC, USA, 2002. IEEE Computer Society. Cited on page(s): 12

[GS99] S. K. S. Gupta, P. K. Srimani. An adaptive protocol for reliable multicast in mobile multi-hop radio networks. *WMCSA '99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, strona 111, Washington, DC, USA, 1999. IEEE Computer Society. Cited on page(s): 12

[GS03] S. K. S. Gupta, P. K. Srimani. Self-stabilizing multicast protocols for ad hoc networks. *Journal of Parallel and Distributed Computing*, 63(1):87–96, 2003. Cited on page(s): 12

[GZ04] L. Guibas, F. Zhao. *Wireless Sensor Networks. An Information Processing Approach.* Elsevier/Morgan Kaufmann Publishers, 2004. Cited on page(s): 2

[Har01] T. Hara. Effective replica allocation in ad hoc networks for improving data accessibility. *Proceedings IEEE Infocom 2001*, strony 1568–1576, 2001. Cited on page(s): 6

[Har03] T. Hara. Replica allocation methods in ad hoc networks with data update. *Mobile Networks and Applications*, 8(4):343–354, 2003. Cited on page(s): 6

[Har05] T. Hara. Strategies for data location management in mobile ad hoc networks. *ICPADS (1)*, strony 147–153. IEEE Computer Society, 2005. Cited on page(s): 7

[HHN03] T. Hara, H. Hayashi, S. Nishio. Cache invalidation for updated data in ad hoc networks. Robert Meersman, Zahir Tari, Douglas C. Schmidt, redaktorzy, *CoopIS/DOA/ODBASE*, wolumen 2888 serii *Lecture Notes in Computer Science*, strony 516–535. Springer, 2003. Cited on page(s): 10

[HHN04] T. Hara, H. Hayashi, S. Nishio. Replica allocation considering data update intervals in ad hoc networks. *MWCN*, strony 131–142. Springer, 2004. Cited on page(s): 6

[HM04] T. Hara, S. K. Madria. Dynamic data replication using aperiodic updates in mobile adhoc networks. *DASFAA*, wolumen 2973 serii *Lecture Notes in Computer Science*, strony 869–881. Springer, 2004. Cited on page(s): 6

[HMN04] T. Hara, N. Murakami, S. Nishio. Replica allocation for correlated data items in ad hoc sensor networks. *SIGMOD Rec.*, 33(1):38–43, 2004. Cited on page(s): 6

[HP04] Y-Ch. Hu, A. Perrig. A survey of secure wireless ad hoc routing. *IEEE Security & Privacy*, 2(3):28–39, 2004. Cited on page(s): 4

[HS03] G. Heine, H. Sagkob. *GPRS Gateway to Third Generation Mobile Networks.* Artech House Publishers, 2003. Cited on page(s): 2

[IMTW02] S. Ishihara, T. Mizuno, M. Tamori, T. Watanabe. A replica distribution method with consideration of the positions of mobile hosts on wireless ad-hoc networks. *ICDCS Workshops*, strony 331–335. IEEE Computer Society, 2002. Cited on page(s): 3, 6

[JKXY04] Z. Jing, Y. Kan, L. Xicheng, W. Yijie. A dynamic adaptive replica allocation algorithm in mobile ad hoc networks. *PerCom Workshops*, strony 65–69. IEEE Computer Society, 2004. Cited on page(s): 6

[JW92] S. Jajodia, O. Wolfson. Distributed algorithms for dynamic replication of data. *PODS '92: Proceedings of the eleventh ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, strony 149–163, New York, NY, USA, 1992. ACM Press. Cited on page(s): 6

[Kle03] L. Kleinrock. An internet vision: the invisible global infrastructure. *Ad Hoc Networks*, 1(1):3–11, 2003. Cited on page(s): 2

[KW03] S.-Y. Kuo, S.-C. Wang. Communication strategies for heartbeat-style failure detectors in wireless ad hoc networks. *2003 International Conference on Dependable Systems and Networks (DSN 2003), 22-25 June 2003, San Francisco, CA, USA*, strony 361–370, 2003. Cited on page(s): 12

[LNR04] G. Lin, G. Noubir, R. Rajaraman. Mobility models for ad hoc network simulation. *Proceedings of Infocom*, 2004. Cited on page(s): 3

[LVW03] H. Lee, N. H. Vaidya, J. L. Welch. Location tracking using quorums in mobile ad hoc networks. *Ad Hoc Networks*, 1(4):371–381, 2003. Cited on page(s): 7, 8

[LW02] B. Li, K. H. Wang. Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks. *IEEE Joint Conference of Computer and Communication Societies (INFOCOM 2002)*, strony 1089–1098, New York, NY, June 23–27 2002. Cited on page(s): 6

[MNP+02] J. T. Malinen, A. Nilsson, Ch. E. Perkins, A. J. Tuominen, R. Wakikawa. Aodv and ipv6 internet access for ad hoc networks. *SIGMOBILE Mobile Computing and Communications Review*, 6(3):102–103, 2002. Cited on page(s): 5

[Mos04] C. Moss. The nodes revolution. *IEE Communications Engineer*, (February/March 2004):18–21, 2004. Cited on page(s): 2

[MW91] A. Milo, O. Wolfson. The multicast policy and its relationship to replicated data placement. *ACM Trans. Database Syst.*, 16(1):181–205, 1991. Cited on page(s): 6

[Pat01] P. A. Patsouris. Algebraic modeling of an ad hoc network for mobile computing. *J. Parallel Distrib. Comput.*, 61(7):884–897, 2001. Cited on page(s): 3

[Per00] Ch. E. Perkins. *Ad Hoc Networking*. Addison-Wesley Professional; 1st edition, 2000. Cited on page(s): 1

[PR97] E. Pagani, G. P. Rossi. Reliable broadcast in mobile multihop packet networks. *MobiCom '97: Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking*, strony 34–42, New York, NY, USA, 1997. ACM Press. Cited on page(s): 10

[PR99a] Ch. E. Perkins, E. M. Royer. Ad-hoc on-demand distance vector routing. *WMCSA '99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, strony 90–100, Washington, DC, USA, 1999. IEEE Computer Society. Cited on page(s): 4

[PR99b] Ch. E. Perkins, E. M. Royer. Multicast operation of the ad-hoc on-demand distance vector routing protocol. *Proceedings of the 5th Annual ACM/IEEE international Conference on Mobile Computing and Networking*, strony 207–218, New York, NY, USA, 1999. ACM Press. Cited on page(s): 5, 12

[PSS03] P. J. Pietri, P. Savelli, E. Seurre. *GPRS for mobile Internet*. Artech House Publishers, 2003. Cited on page(s): 2

[RT99] E. M. Royer, C.-K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6(2):46–55, 1999. Cited on page(s): 4

[Tan02] A. S. Tanenbaum. *Computer networks*. Prentice Hall; 4 edition, 2002. Cited on page(s): 2, 4

[Toh01] C. K. Toh. *Ad hoc mobile wireless networks: protocols and systems*. Prentice Hall PTR; 1st edition, 2001. Cited on page(s): 1

[Vai00] N. H. Vaidya. Tutorial on mobile ad hoc networks: routing, mac and transport issues. The 6th Annual International Conference on Mobile Computing and Networking, MobiCom Tutorials, Boston, MA: ACM, 2000. Cited on page(s): 2

[Zap02] M. G. Zapata. Secure ad hoc on-demand distance vector routing. *SIGMOBILE Mobile Computing and Communications Review*, 6(3):106–107, 2002. Cited on page(s): 5