Management of Ad-Hoc Networks

Remi Badonnel, Radu State, Olivier Festor

This chapter presents a state of the art of the management of ad-hoc networks and is structured as follows. We introduce ad-hoc networking and the challenges of ad-hoc networking management in Section 1. We present the main management paradigms and the underlying architectures in Section 2. Several application domains including monitoring, configuration and cooperation control are covered by Section 3. Finally, Section 4 concludes the chapter with a synthesis of research orientations in this domain.

1 Introduction

The increasing need of mobility in networks and services leads to new networking paradigms including ad-hoc networking [27, 39, 31] where the mobile devices can establish direct and multi-hop communications among them.

Definition 1 A mobile ad-hoc network (MANET) is a self-organizing network that can be spontaneously deployed by a set of mobile devices (laptops, phones, personal assistants) without requiring a fixed network infrastructure.

The deployment of an ad-hoc network is performed dynamically by the mobile devices (or nodes) themselves: each mobile node is both a terminal that communicate with the others and a router that can forward packets on behalf of the other nodes. If two network nodes cannot communicate directly because they are not localized in the same one-hop neighborhood, a multi-hop communication is established using intermediate nodes. Ad-hoc networks provide localized and temporal-limited connectivity in dedicated target environments: search and rescue services, intervehicular communications, last mile Internet delivery and geographically challenging environments. The emergence of new applications is favored by the advances in hardware technologies and by the standardization of ad-hoc routing protocols.

Compared to fixed networks, ad-hoc networks permit to reduce the costs of deploying and maintaining the infrastructures but they present other additional constraints such as:

- dynamic topology: such networks do not require a fixed infrastructure.

They are spontaneously deployed from mobile nodes and are then highly dynamic, since nodes might come and go based on user mobility, out-of-reach conditions and energy exhaustion.
heterogeneity of devices: a typical ad-hoc network is composed of various devices such as laptops, personal digital assistants, phones and intelligent sensors. These devices do not provide the same hardware and software capabilities but they have to interoperate in order to establish a common network.

distributed environment: the network running is based on the cooperation of independent nodes distributed over a given geographical area. The network maintenance tasks are shared among these nodes in a distributed manner based on synchronization and cooperation mechanisms.

resource constraints: bandwidth and energy are scarce resources in ad-hoc networks. The wireless channel offers a limited bandwidth, which must be shared among the network nodes. The mobile devices are strongly dependent on the lifetime of their battery.

limited security: ad-hoc networks are more vulnerable than fixed networks because of the nature of the wireless medium and the lack of central coordination. Wireless transmissions can be easily captured by an intruding node. A misbehaving node can perform a denial of service attack by consuming the bandwidth resources and making them unavailable to the other network nodes.

The key concept of ad-hoc networking is the routing mechanism. Ad-hoc networks rely on a routing plane capable to fit with their dynamics and resource constraints. In particular, an ad-hoc routing protocol must be capable to maintain an updated view of the dynamic network topology while minimizing the routing traffic. Numerous ad-hoc routing protocols were proposed in the last ten years. The MANET working group [22] manages the standardization of ad-hoc IP routing protocols at the IETF and classifies them in two major routing approaches: reactive routing and proactive routing.

**Definition 2** A reactive routing protocol is a routing protocol which establishes the routes in an on demand manner only when a route is requested by a node, by initiating a route discovery process.

In the reactive approach, a source node had to flood a request for a route to a destination node in the ad-hoc network. The intermediate nodes are discovered and memorized during the flooding. When the destination node receives the request, it uses this reverse path to contact the source node and to provide the requested route. A typical example of reactive protocol is the Ad-hoc On-demand Distance Vector (AODV) protocol described in [30].

**Definition 3** A proactive routing protocol is a routing protocol which periodically maintains the network topology information on each network node, by managing local routing tables.

In the proactive approach, each node maintains a routing table that is regularly updated by exchanging topology information with the other nodes. An
example of a proactive protocol is the Optimized Link State routing protocol (OLSR) [15]. Each node performs two main operations: it determines the list of direct-connected neighbor nodes by accomplishing link sensing through periodic emission of beaconing hello messages and exchanges link state information with the other nodes by flooding topology control messages. The OLSR heuristic reduces the control overhead by selecting a subset of nodes called Multi-Point Relays (MPRs) responsible for forwarding broadcast messages during the flooding process.

The two routing approaches do not present the same properties and the choice of the one or the other is done according to the particular application. The reactive routing approach reduces significantly the control overhead since it does not require to flood topology information periodically. However, the proactive routing approach offers a lower latency as the route discovery has not to be done. Hybrid routing protocols provide a trade-off between control overhead and latency.

While the routing plane is progressively converging towards the use of standardized protocols, the management plane raises interesting research challenges towards providing a management framework to ad-hoc networks.

### 1.1 Management Requirements

A fixed network management model is not directly appropriate for ad-hoc networks. New management paradigms and architectures must be defined to cope with the dynamics and resource constraints of such networks. Managing ad-hoc networks presents us with different research issues which are not encountered in the same way in the common fixed networks:

- **management domain related:** since ad-hoc networks are dynamically formed by loosely coupled entities, the notion of administrative domain, responsible for the overall management is difficult to define. Although, from a technical point of view, cooperative and role-based schemes are possible such that the manager/agent roles are assigned dynamically, the management layer has to implement higher level business policies, which are hard to define in blurry shaped administrative domain.

- **relevant management information:** what should be monitored and what actions to be taken is still unknown in this case. The challenge is to identify the essential pieces of information relevant in this context and to determine the possible corrective configuration operations in a close-loop management paradigm.

- **management agent reliability and willingness to cooperate:** the management agent is constrained by its inherent limits (connectivity, battery, failures) so that the provided monitored data might be biased. Additionally, malicious or non-cooperative nodes might provide false data or no data at all. The key issue is to define management paradigms, where biased or faulty information can be out-weighted or discarded as such.
• cost of management: resources (essentially network bandwidth and battery power) are already scarce in ad-hoc networks and monitoring and management actions are additional consumers in an already resource-limited context. Defining lightweight schemes, where already available information is used and resource availability and consumption can be minimal, are the main issues.

• self-organization: the dynamic nature of mobile ad-hoc networks requires that the management plane is able to adapt itself to the heterogeneous capacity of devices and to the environment modifications. The ad-hoc network is deployed in a spontaneous manner, the management architecture should be deployed in the same manner in order to minimize any human intervention.

• robustness and scalability: the management plane must be easy to maintain and must remain coherent, even when the ad-hoc network size is growing, when the network is merging with another one or when it is splitting into different ones.

New management paradigms were considered to deal with the research issues posed by ad-hoc networks. They also had to be integrated into the common management approaches of fixed networks.

1.2 Management Organizational Models

An ad-hoc network corresponds to a distributed environment where different organizational models can be considered for the management purposes. A management organizational model defines the role (manager or agent) of each component of the management plane and specifies how each component interacts with the others. Typically, we can distinguish four different management organizational models presented on figure 1:

• centralized model (figure 1(a)): a single network manager manages all the nodes in the whole network. The significant advantage of this model is to simplify the management task by considering a single point of control. However, the centralized model can generate a high management traffic and is not robust since the failure of the manager leads to the failure of the whole management system.

• centralized hierarchical model (figure 1(b)): the network manager uses local managers as intermediary to perform some management tasks. Each local manager manages a subset of network nodes and is assigned with a given degree of responsibility. The manager is the central point of control since it has the highest degree of responsibility.

• distributed model (figure 1(c)): the management is performed by a set of managers communicating according to a peer-to-peer scheme. Each manager has the same degree of responsibility and is fully responsible for
Manager

(a) Centralized management

Manager

(b) Centralized hierarchical management

Manager

(c) Distributed management

Manager

(d) Distributed hierarchical management

Figure 1: Organizational models for ad-hoc network management
managing a subset of nodes in the network. The usage of multiple managers reduces the overhead of management messages and improves the robustness of the management system. However, a cooperative mechanism must be defined in order to ensure the coherence of the management operations taken by managers.

- distributed hierarchical model (figure 1(d)): this model is a distributed model where each of the managers can delegate part of its management tasks to local managers.

These organizational models are not specific to ad-hoc networking but are recurrent in network management. Because of the nature of ad-hoc networks, the role assignment and the relationships among components are highly dynamic compared to regular fixed networks.

2 Management Paradigms

This section presents the main management paradigms for mobile ad-hoc networks and describes the underlying architectures able to cope with the constraints of such dynamic and resource scarce networks.

2.1 SNMP-based Management

Current network management systems use a set of available management protocols (CLI [6], SNMP [38], XML-based [18]) to exchange management information between a network manager and the agents. The Simple Network Management Protocol (SNMP) [38] is among the most common used protocol for network monitoring and configuration. Integrating ad-hoc networks into common agent/manager models require the compatibility with this protocol, but also requires in the same time a reorganization of the management plane in order to limit the underlying network overhead.

2.1.1 ANMP approach

Among the pioneering management approaches for ad-hoc networks, the Ad-Hoc Network Management Protocol (ANMP) [14] architecture defines a centralized hierarchical solution based on the SNMP protocol. The compatibility relies on using the same protocol data unit structure and the same management information base structure as in SNMP. The major difference comes from the architectural model (presented in figure 2), which combines clusters and hierarchical levels. ANMP comes within the scope of management by delegation. It clusters the management plane and organizes it in a three-level hierarchy composed of: individual managed nodes interacting as agents at the lowest level, clusterheads interacting as local managers at the middle level and a central network manager at the top level.
The hierarchical model reduces the network traffic overhead by collating management data at the intermediate levels, but is limited to three hierarchical levels in order to facilitate the maintenance of the hierarchy in such dynamic networks. Two clustering algorithms are defined at the application layer to organize the management plane:

- The first approach is based on graph-based clustering and forms clusters of one-hop neighbor nodes based on the network topology. In the network topology seen as a graph, each mobile device is represented by a node identified by a unique ID (MAC address, serial number) and a bidirectional link exists between two nodes if and only if they are within the transmission range of each other.

  The algorithm performs in a distributed manner the organization of one-hop clusters where the node with the minimum ID among its neighbors is elected as the clusterhead. Each node implements internal data structures including the list of one-hop neighbors, the list of nodes in the cluster and a ping counter indicating last time a ping was received from the clusterhead.

- The second approach called geographical clustering forms clusters of up to three-hop neighbors based on the spatial density. The approach requires that node positions are known through a geographical positioning system.
such as GPS. It is based on two algorithms: a centralized periodic processing for cluster definition and a distributed maintenance processing to manage node mobility.

The centralized processing considers the ad-hoc network as a rectangular box and calculates horizontal and vertical stripes in order to split the network into rectangular clusters of different sizes (up to three times the transmission range) but of homogeneous spatial density. In each of these rectangular clusters, the ad-hoc node located in the box center is defined as the clusterhead.

The clustering processing is executed periodically and a maintenance processing manages node mobility between two clustering updates. The maintenance processing performs at the cluster level defines an announcement protocol for nodes to leave and join a cluster and provides guest facility in each cluster for nearby unmanaged nodes.

ANMP is not limited to the two presented approaches but can be extended to other clustering algorithms. In particular, while ANMP makes a distinction between clustering at the application layer for management purposes and clustering at the routing layer, it may be suitable to propose a lightweight management approach by piggybacking an ad-hoc network routing protocol.

A management information base (MIB) is defined by ANMP as an additional group to the standardized MIB-II of SNMP. This group called anmpMIB includes security information data to implement the Level-based Access Control Model (LACM) in order to secure management operations and management information access. In the management plane, each node has a clearance level: the higher the level is, the lower the security level is. The group also defines topology information including two information subgroups for each clustering approach, and agent information statistics maintained by managers. Finally, the group describes objects related to device battery and power consumption, including a drain function of the battery used for predictive purposes.

ANMP defines a complete SNMP-compatible management approach including a security model. The clustering schemes can ensure that over 90 percents of nodes are managed while keeping a low network management traffic overhead.

GUERRILLA approach

An alternative solution is proposed in [37] by the GUERRILLA management architecture, which relies on two concepts: management plane self-organization and dynamic deployment of management related code. GUERRILLA proposes a context-aware approach where management intelligence is spread over the network nodes according to their capabilities. It aims at making management operational in highly dynamic networks by maintaining connectivity in the management plane (and enabling disconnected management) and by minimizing management costs on node resources.
This distributed architecture (presented in figure 3) is composed of nomadic managers (higher tier) and active probes (lower tier):

- A nomadic manager maintains the connectivity in the management plane with the other managers and updates the management plane configuration in an autonomous way.

- Deployed by a nomadic manager, an active probe is an extension to an SNMP agent. It integrates an execution environment for hosting monitoring scripts and performs the localized management operations.

The clustering algorithm relies on the Cluster-based Topology Control (CLTC) approach which minimizes the consumption of the node resources. Moreover, the nomadic managers implement an auto-configuration mechanism to determine the management actions to be executed in an adaptive manner. In a formal way, a nomadic manager determines the management action $a$ – from a set of possible actions $A$ – that maximizes the objective function $f(a)$ defined by equation 1.

$$f(a) = \frac{U(T(s, a)) - U(s)}{C(a)}$$  

In this equation, $s$ represents the view of the domain state perceived by the manager and $T(s, a)$ is a transition function that provides the new state of the
domain predicted by the manager if the action $a$ is executed. $U(s)$ defines a utility function possibly derived from policies, while $C(a)$ represents the management cost function. By maximizing the function $f(a)$, the nomadic manager aims at improving the managed environment utility with a minimal cost.

GUERRILLA comes within the scope of management by delegation. It proposes fundamental good assumptions such as self-organization, disconnected management and takes into account heterogeneous devices.

2.2 Policy-based Management

Policy-based management (PBM) consists in defining high-level management objectives based on a set of policies that are enforced in the network. Architectures for policy-based management rely on two main functional components:

- the Policy Decision Point (PDP) component which aims at converting the high-level policies into low-level device-dependent policies and performing their distribution,

- the Policy Enforcement Point (PEP) component which enforce the low-level policies they receive from the PDP component.

The Common Open Policy Service (COPS) protocol was specified to standardize the distribution policy through the PDP-PEP communications. Several research work in [26, 32, 13] focus on the adaptation of the policy-based scheme to ad-hoc networking constraints.

2.2.1 Decentralized COPS-PR approach

A first management framework is introduced by Phanse in [32, 33] to extend the policy-based management for quality of service in ad-hoc networks. Phanse decentralizes COPS for provisioning (COPS-PR). The latter is an extended protocol which combines the outsourcing and provisioning models allowing to support hybrid architectures.

This distributed hierarchical approach is described on figure 4. It relies on three key mechanisms:

- clustering and cluster management: the clustering of the ad-hoc network relies on a simple k-hop clustering scheme: during the initial deployment, some Policy Decision Point nodes called super-PDPs are present in the network and form clusters with the PEP client nodes located in the k-hop neighborhood. The number of intermediate nodes between a super-PDP server and its PEP client nodes is therefore bounded to k-hops and the management system can expect to provide acceptable response time performance.

- dynamic service redundancy: a mechanism for dynamic service redundancy is used when the super-PDP servers are not enough numbered to serve all the clients in the network, or when clients are moving from one
Figure 4: Policy-based management with decentralized COPS-PR

cluster to another one. It consists of a delegation service and a redirection service. The delegation service answers the lack of super-PDP servers in the network, by allowing a super-PDP server to delegate part of its decisions to another node interacting as a regular PDP server. In this case, the super-PDP server keeps a hierarchical control on the regular PDP servers. The redirection service manages the PEP clients moving among network clusters by allowing a PDP server to dynamically redirect a client to a more appropriate less distant PDP server.

• service location discovery: a discovery mechanism is proposed to automatically discover new policy servers but it is not yet designed.

The management framework supports multi-domain policy negotiation for PEP client visitors from other domains. An inter-PDP server signaling is introduced in COPS messages to exchange policies among PDP servers from different domains. If the visited PDP server cannot find policies for a visitor PEP client, it queries the PEP client to get its home PDP server address and negotiates the policies with the home PDP server.

Phanse’s approach is an innovating solution for policy management in ad-hoc networks, which was prototyped and validated experimentally in a multi-hop environment. However, the considered communication protocol COPS-PR is not
widely used in today’s network management systems. Some improvements could be possible for the multi-domain policy negotiation mechanism. In particular, the policy could be transferred during the redirection and the PEP clients could come with their own policies.

### 2.2.2 DRAMA approach

![DRAMA architecture](image)

DRAMA defines a centralized hierarchical policy-based management system for ad-hoc networks in [13, 12]. This three-level architecture described on figure 5 is composed of a set of policy agents having the same basic structure. The Global Policy Agent (GPA) at the top level manages multiple Domain Policy Agents (DPA). A Domain Policy Agent manages in turn multiple Local Policy Agents (LPA). Network management policies are propagated hierarchically from the GPA agent to the DPAs and from the DPAs to the LPAs.

The conceptual architecture is relatively simple, but allows to perform an exhaustive set of experiments: the management system was prototyped and demonstrated in a realistic environment to illustrate several use cases including CPU utilization reporting, server reallocation upon failure, reconfiguration of bandwidth allocation.
2.2.3 Node-to-node approach

A node-to-node approach is defined in [26] in the framework of corporate ad-hoc networks presenting uniform management and company-wide policies.

**Definition 4** The node-to-node policy distribution defines a fully distributed policy distribution where a network node exchanges and synchronizes the policies with the neighboring nodes (see figure 6).

This distributed architecture works over the DiffServ quality-of-service model providing the network layer differentiated service: the communications among network nodes can be marked with different priority levels.

In this architecture, each ad-hoc device exports both DiffServ functions: the edge router monitors and marks local application traffic and the core router classifies and schedules routed packets. From the management plane point of view, the ad-hoc devices embed both PDP and PEP functionalities including a local policy repository to store the policy rules applicable to the device.

The key concept of the architecture is based on a two-phase policy distribution using a synchronization mechanism:

- In the first phase of the distribution, the ad-hoc device of the administrator interacts as a Policy Decision Point (PDP) and distributes the policies to the other devices behaving as Policy Enforcement Point (PEP) and storing the policies on their local repository. This phase is not sufficient to configure all the ad-hoc nodes if these were not connected to the network at the distribution time.

- To propagate the policies to all the devices, the second phase consists in a permanent synchronization among two neighboring nodes. An already-configured node can interact as a local PDP to configure a device which has not received the latest policies.

An authority mechanism allows a management hierarchy with delegation. Each manager has an authority level known by all. Network policies are tagged with manager identifiers. The more authoritative policies always replace the less authoritative ones. The approach describes an interesting device-level management framework for enterprise networks. It assumes that in the future quality-of-service mechanisms will be available at the MAC layer. However, the synchronization protocol and the consistency problems of the policies are not discussed.

2.2.4 Scalability evaluation

Burgess evaluates the scalability of policy-based configuration management for ad-hoc networks in [9]. The performance evaluation relies on an analytical framework which takes into account the unreliable nature of the communication medium and the mobility factors in ad-hoc networks.
Hypothesis 1 Formally, an ad-hoc network can be represented by a connectivity matrix $A$ where rows/columns stand for an ad-hoc node. A matrix entry $A[i,j]$ indicates whether node $i$ is connected to node $j$. The connectivity among two peers of nodes undergoes time and mobility caused variations in the context of ad-hoc networks. The matrix entry value can be therefore defined by a probability value.

The connectivity matrix serves as an analytical basis for analyzing the network traffic overhead generated by different policy-based management models and for evaluating their scaling behavior.

The organizational models presented in Section 1 are refined for the context of policy-based management and leads to a set of six different management models. Each of these models corresponds to a different degree of decentralization with respect to three main criteria:

- policy decision: the deployment of policies can be controlled by a central manager or by a set of distributed managers. The final hosts can either strictly respect these policies or can freely modify some aspects of them.
- policy exchange: the policies can be received from a manager, from an intermediate manager in a hierarchical manner, or from another node in a node-to-node scheme,
• policy enforcement: the policies can be transmitted by a manager or can be locally enforced by the hosts.

The scalability analysis relies on a key assumption: the probability of a successful configuration depends on the regularity of the maintenance operations, which in turn depends on the reliability of the communications between the policy source and the hosts. In order to estimate the efficiency of the configuration, the average rate of errors generated by an host is compared to the average rate of repair maintenance operations which can be done in function of the communication channel size and reliability. The configuration is successful if and only if all the host errors have been repaired. Therefore, the average rate of possible repair operations must be higher than the average rate of host errors.

The results of this analysis provide a stochastic quantification of the scalability of policy-based management schemes. They confirm the limits of a centralized management due to the communication bottleneck caused by the centralized manager. Distributed management schemes offers a better robustness and scalability but introduce policy convergence issues.

2.3 Middleware for Management

An ad-hoc network is an heterogeneous set of nodes with diverse capabilities in terms of energy and processing facilities but also in terms of operational protocols and services. A programmable middleware defined by [20] provides a common communication infrastructure capable to align dynamically the software capabilities of network nodes. The middleware defines a programmable scheme where each node is able to share, download and activate loadable plugins offering specific operational functionalities such as an application service or a routing protocol. The key idea is to elect a common set of loadable plugins and to distribute it among the network nodes in order to facilitate their communications.

The middleware for management is organized into clusters, where the cluster head coordinates the election and distribution of loadable plugins. In a given cluster, each node exposes several loadable plugins and stores locally the executable code in its plugin repository. The cluster head initiates the election process by contacting the cluster nodes and requesting them to advertise the candidate plugins. Based on the set of advertised plugins $P$, the cluster head uses an election algorithm which determines the most appropriate(s) plugin(s) for the given context. The selection of plugins aims at selecting the plugins which minimize the cost function $g(p)$ defined in equation 2.

$$g(p) = \sum_{i=1}^{n} w_i \times C_i(p)$$

where $w_i \in [0, 1], \sum_{i=1}^{n} w_i = 1$ and $C_i \in [0, 10], \forall i \in [1, n]$

The characteristics $C_i(p)$ of a given plugin $p$ are rated from 0 to 10 and are assigned with a mathematical weight $w_i$ by the cluster head. This cost function considers the characteristics such as CPU usage, memory consumption and plugin size to converge to a common basis of plugins among the cluster nodes. The
cluster head initiates then the plugin flooding by requesting the cluster nodes that own the plugins to distribute them to their one-hop neighbors in the cluster in a peer-to-peer manner. Each cluster mate stores the common plugins into the plugin repository and activates them. The cluster head is capable to configure and to modify on-the-fly the parameters of the current activated plugins.

The middleware infrastructure was implemented by combining a lightweight XML-based message-oriented protocol and the Java platform for small and medium devices. The first evaluation performance shows the management middleware is efficient in terms of convergence and scalability. The middleware was limited to a centralized scenario with a single cluster and must be extended to multiple clusters with cluster-to-cluster communications.

### 2.4 Probabilistic Management

A pure and sound management approach, where all nodes are managed at any time is too strict for ad-hoc networks.

**Definition 5** Instead of addressing the management of the whole network, a probabilistic management approach focus on a subset of network devices or on a subset of network parameters to derive the behavior of the whole network with a given probability guarantee.
A probabilistic management approach is proposed in [3] for ad-hoc networks based on probabilistic guarantees. A subset of nodes is considered to be managed in order to provide a light-weight and reliable management. These nodes are determined based on their network behavior so as to favor subsets of well-connected and network participating nodes. The assumption of the probabilistic management approach was motivated by the simple observation that ad-hoc network management should not comprise the totality of network elements (as it is typically the case in fixed networks).

The requirements on the management plane are relaxed and are limited to a subset of nodes called a spatio-temporal connected component. This component is defined as a subset of the ad-hoc network, such that nodes within such a component have a high probability of being directly connected. The term spatial derives from this neighborhood notion, while the term temporal is related to the temporal behavior of this neighborhood. In a store and forward oriented architecture, such nodes are also capable to inter-communicate at a higher hop-count. The probabilistic management is limited to the first and eventually second largest spatio-temporal connected component. With respect to such a selective management scheme, probabilistic guarantees are determined in function of the percentage of nodes to be managed. The underlying algorithmic describes the extraction of spatio-temporal components from a probability

Figure 8: Probabilistic management
matrix similar to the matrix $A$ defined in Section 2.2.4. The election of manager nodes in each component can be performed with a centrality measure: the most central nodes of a component are elected as managers.

**Definition 6** The centrality is a measure capable to quantify the relative importance of a node in a network or a graph structure. Several variants of this measure have been defined in social networking [7].

A spatio-temporal component is a non-oriented connected graph associated to a $m$-dimensional matrix $M$. A typical centrality measure that can be applied on this graph is the degree centrality $c_{\text{degree}}$ defined by equation 3. A node $i$ is considered as central if it is directly connected to a maximal number of other nodes.

$$c_{\text{degree}}(i) = \sum_{j=1}^{j=m} M(i,j) \text{ for a given node } i$$

The degree centrality is a local measure limited to the one-hop neighborhood information. An alternative centrality measure is the eigenvector centrality defined by Bonacich in [4]. The key idea is to define the centrality measure in a recursive way: the more a node is directly connected to central nodes, the more the node is central. For such a non-oriented connected graph, Bonacich demonstrates that the problem can be reduced to the typical eigenvector problem defined by equation 4.

$$M \times \vec{x} = \lambda \times \vec{x}$$

The vectors that solve this equation are called eigenvectors and are paired with corresponding eigenvalues. The principal eigenvector $\vec{v}_{\text{principal}}$ is the solution vector which is paired with the highest eigenvalue. The eigenvector centrality $c_{\text{eigenvector}}$ of node $i$ corresponds to the $i$-th element of the principal eigenvector of matrix $M$, as mentioned in equation 5.

$$c_{\text{eigenvector}}(i) = \vec{v}_{\text{principal}}(i) \text{ for a given node } i$$

A generalized version of the eigenvector centrality has been defined in [5] in the case of oriented connected graphs.

Another probabilistic scheme is designed at the level of the information model in [16]. It aims at improving the acquisition of monitoring information in an ad-hoc network. In this scheme, monitoring information is not directly queried from the network devices but must be processed by a correlation-based probabilistic model. The direct access to raw monitoring data is not possible. An information query must be formulated to the acquisition module which integrates the probabilistic model. This model relies on time-varying multivariate normal distributions and allows to correlate multiple raw monitoring data through statistical techniques in order to improve the information reliability. Disparate and faulty monitoring raw data may generate biased approximated information. The correlation mechanism thus provides information approximations with probabilistic confidences.
The scheme allows to reduce the number of nodes involved in the data monitoring according to the asked confidence. The architecture defines a novel approach to integrate an information base with a correlation probabilistic model and provides a trade-off between information confidence and data acquisition cost. However, the efficiency of the architecture is directly correlated to the availability of efficient probabilistic models for a given monitored metric.

3 Application Domains

This section details the research work addressing management applications in MANETs including network monitoring, configuration of IP addresses and control of node cooperation.

3.1 Monitoring

Monitoring ad-hoc networks resides in measuring performance metrics, collecting the measurement data and processing these measurements.

3.1.1 WANMON approach

A local monitoring approach called WANMon (Wireless Ad-Hoc Network Monitoring Tool) is proposed in [29] for determining the resources usage of an ad-hoc node. The architecture is locally deployed such that an agent is responsible for monitoring the resources of an ad-hoc device. It monitors the resource consumption generated by ad-hoc network-based activities: network usage, power usage, memory usage and CPU usage. In particular, the tool aims at distinguishing both the resources consumed for the node itself and the resources consumed by the routing activity on the behalf of other nodes. WANMon is one of the first monitoring tools dedicated to ad-hoc networking, but presents some limitations. The tool cannot provide real-time information and is limited to a local monitoring activity.

3.1.2 DAMON approach

DAMON described in [34] defines a Distributed Architecture for MONitoring multi-hop mobile networks. This generic monitoring system is composed of monitoring agents and of data repositories called sinks that store monitored information. A centralized deployment with a single monitoring sink is supported for small networks, but DAMON provides also in a distributed scheme seamless support for multiple sinks for larger networks. In DAMON, each agent is hosted by an ad-hoc device, monitors the network behavior and sends the monitoring measurements to a data repository. DAMON defines a robust solution adapted to device mobility by supporting sink auto-discovery by agents and resiliency of agents to sink failures. The auto-discovery of repositories relies on a periodic beaconing mechanism initiated by sinks to advertise their presence and broadcast instructions to agents. A network agent receives the beacon messages,
Figure 9: Distributed monitoring with DAMON

maintains a list of available data repositories and associates with the closest sink as its primary repository in order to send the collected information. The distance between the agent and the primary repository is possibly more than one hop. In this case, monitoring data from multiple agents can be aggregated by an intermediate agent to facilitate the data collection. DAMON distinguishes two types of monitoring information: time dependent information (list of neighbors, energy left on a device) which require to be frequently transmitted to sinks and time independent information (packet logs, daily traffic statistics).

The resiliency to sink failures relies on a simple switching operation to a different sink when the primary sink undergoes a failure. An agent can detect the failure of the primary sink because of the absence of beaconing messages on a given time period. In this case, the agent updates the list of available data repositories and switches to a new primary sink in order to send the collected information. In order to prevent intermittent switching between two sinks, the agent can modify its primary repository only after the successful reception of a determined number of beaconing messages. The architectural framework of a monitoring agent is composed of a packet classifier module, categorizing in input the packets according to their types and a set of packet handler modules performing specific processing on the received packets such as beacon listening.
and time dependent information aggregation.

DAMON proposes a distributed scheme for monitoring ad-hoc networks based on multiple data repositories. A complete implementation of DAMON was demonstrated with the on-demand distance vector (AODV) routing protocol. But, the architecture supports a wide range of other protocols and devices. While the solution is robust and maintainable with little effort, the proximity-based association mechanism should be improved to ensure an homogeneous distribution of agents with sinks.

3.1.3 Synchronization of monitoring data

By definition an ad-hoc network is dynamic and temporal such that synchronization mechanisms are required to maintain the coherence and to maximize the life time of monitoring information in such a distributed environment. The problem of time synchronization in sparse ad-hoc networks cannot be solved with regular clock synchronization algorithms because of the restriction of node resources and of the unreliability of network links.

A synchronization method is described in [35] to synchronize information in distributed ad-hoc networks with a low message overhead as well as a low resource consumption. The solution is an adaptation of a time stamps based algorithm using unsynchronized clocks of local nodes. If the monitoring information is not duplicated, the information maintained by a local node disappears when the node is failing or is leaving the ad-hoc network. A dissemination process is therefore defined in [41] to both optimize the information survival within the network and minimize the cost of the information dissemination. A monitoring information should be defined with an expiration time as this information cannot be maintained indefinitely without any update. In particular, the statistical distribution of timescales and the impact on ad-hoc network performances is discussed in [17].

3.1.4 Processing of monitoring data

In order to process the monitoring information from an ad-hoc network, an analytical method based on distributed contrast filtering mechanisms and graph dependency analyses is proposed in [2]. The goal of this information processing is to highlight network traffic patterns in order to evaluate the network behavior. The contrast filtering can be performed in a centralized manner by the network manager but a distributed scheme where the filtering is processed locally by individual nodes is preferred for scalability and performance purposes. The filtering process aims at making in evidence disparity among nodes. Applied for a given network metric, it consists in a local comparison of the normalized value of the metric among network nodes located in a same neighborhood. A concrete usage relies on applying the contrast filter on the routed packet number per node in order to reveal routing paths and to detect network nodes interacting in the routing task.

A second method is defined in the same paper based on dependency graphs
to estimate node influence in an ad-hoc network. This more refined approach is a direct adaptation to ad-hoc networks of the work described in [8] related to the estimation of nodes influence in fixed wired networks. The method is initially inspired from research in social sciences, where social relationships are studied to detect individuals capable to influence important parts of a social network. In an ad-hoc network, the graph analysis identifies the core ad-hoc nodes participating in the network functioning. The analytical approach in [2] comes within the scope of a more global analytical framework introduced in [9] for modeling ad-hoc networks and for quantifying their behavior. However, the experimental work must be thorough with multiple scenarios implying an extensive set of network parameters.

Corrective management operations can be decided from the monitoring information. For instance in [1], a network-wide correction can be defined based on a local monitoring to improve a quality-of-service differentiated service. In this approach, each network node monitors independently the rate of high priority flow. When the rate falls out of a given threshold value, a network node is capable to signal corrective operations to the other nodes such as a selective rejection of lower priority messages in order to achieve quality-of-service guarantees. Many other configuration operations can be envisioned to optimize network performance in a close-loop management framework.

3.2 Configuration

The configuration of ad-hoc networks [36] must be performed in a self-organized manner to ensure robustness and scalability. A basic expectation in configuring ad-hoc networks lies on the autoconfiguration [40, 25] of network devices including the allocation of their IP addresses. While in a fixed wired network the device configuration is often done in a centralized manner using a dynamic host configuration (DHCP) server, such a centralized approach with fixed allocated addressed server cannot be deployed in ad-hoc networks. Alternative address allocation approaches are needed based on a distributed scheme to ensure that two nodes do not have the same address in an ad-hoc network. Allocation approaches are distinguished in [42] into three categories: conflict-free allocation, conflict-detection allocation and best effort allocation.

3.2.1 Conflict-free IP address configuration

The conflict-free IP address configuration consists in ensuring an IP address of a new node is not already assigned. This is done by using dedicated disjoint pools of addresses. A typical example of conflict-free allocation can be outlined with the IP version 6 stateless autoconfiguration. The latter relies on the statement that dedicated pools of MAC level addresses are assigned to network card producers such as an ethernet card is uniquely identified. This address can be used as a unique identifier to define the IP address in order to avoid an allocation conflict. Indeed, the stateless mechanism allows the mobile node to generate its own unique IP address using a combination of the network prefix and a suf-
fix which is locally computed based on the unique MAC address. However, this hardware-based addressing scheme does not consider that the physical addresses of a mobile node can be easily modified by reprogramming the network cards.

![Figure 10: IP address configuration with MANETconf](image)

3.2.2 **Conflict-detection IP address configuration**

The conflict-detection IP address configuration consists in selecting a supposed-to-be available IP address and assigning it after having requested a confirmation from the other network nodes that this address is effectively not in use. MANETconf [28] defines a conflict-detection distributed allocation protocol for ad-hoc networks. No centralized server can ensure the uniqueness of the allocated address. The address allocation relies therefore on a distributed mutual exclusion algorithm performed by the network nodes. Each of the latter maintains the set of the IP addresses already allocated.

The assignment process is performed in an indirect way through an intermediate node called initiator node and is defined by the following process:

- In order to obtain an IP address, a new node chooses in its neighborhood a node already part of the network and requests him to interact as an initiator.
• The initiator node selects an IP address considered to be available for use according to its local view of allocated IP addresses.

• It then performs an address query on the behalf of the new node to the other network nodes.

• At the reception of the address query, a network node sends an affirmative or negative reply if he considers the address is or is not available.

• If the initiator receives a positive reply from the network nodes, the new node is configured with the chosen address, otherwise a new address must be selected and queried by the initiator node.

MANETconf tolerates network partitioning and network merging. On one hand, the ad-hoc network partitioning is managed by an update mechanism: each time an address query is performed by a network node, the latter is capable to update the set of network nodes in his partition by identifying the nodes which did not send back any reply messages. The set of allocated addresses thus corresponds to the set of nodes part of the current partition. On the other hand, when a network merging occurs, the ad-hoc nodes from different partitions exchange and merge their set of allocated addresses. Conflicting addresses which are allocated to two nodes at the same time are identified and reconfigured by the allocation protocol.

MANETconf offers a robust protocol defining a decentralized dynamic host configuration protocol. It supports network partitioning and merging and also resolves concurrent initiation of address allocations. The solution considers a standalone ad-hoc network but could take advantage of considering a hybrid architecture where a gateway connected to an external network would be available. While MANETconf manages both message losses and node crashes, security issues such as malicious behavior of nodes remain open.

3.2.3 Best-effort IP address configuration

The best-effort IP address configuration is based on assigning by default a supposed-unused IP address to a node until a conflict is detected. The PROPHET [42] allocation defines such a best-effort scheme using the random properties of a stateful function noted $f(n)$. The stateful function is configured with an initial state value called seed and is capable to randomly generate sequences of different addresses such that the probability of having duplicate addresses is kept low. During the allocation of addresses, the function is propagated towards network nodes with different seed values as follows:

• In order to join the ad-hoc network, a new node contacts a network node and requests a new IP address.

• The node generates the new IP address using the stateful function $f(n)$ configured with its own seed. It generates also a seed value for the new node.
• The latter can in turn use the stateful function configured with this seed value to generate new IP addresses.

This distribution protocol combined with the properties of the stateful function guarantees a low probability of duplicate addresses.

PROPHET offers an allocation scheme adapted to large scale networks by defining a low communication overhead for address distribution. However, the design of the stateful function should be refined. The solution does not fully avoid address duplication and must be complemented by a conflict detection mechanism.

3.3 Cooperation Control

The deployment of an ad-hoc network is based on the cooperation of mobile nodes and on the mutual utilization of their resources. Management mechanisms are therefore required to stimulate the participation of mobile nodes, such as their involvement to share the medium access and to perform packet forwarding on the behalf of the other nodes, but also to prevent the network overload by misbehaving nodes in order to guarantee a fairness cooperation.

3.3.1 Medium access control

A misbehaving node may not respect the contention resolution mechanisms defined by a wireless medium access control protocol in order to monopolize the bandwidth resources by forcing a not equitable share of the channel. A corrective scheme is defined in [21] over the 802.11 wireless protocol to detect and handle misbehaving nodes at the medium access layer. In this wireless protocol, the contention resolution mechanism requires that a mobile node waits a backoff time interval before starting the transmission of network packets. The backoff interval value is defined by a specific random process ensuring a fair share of the channel. The misbehaving node may force a short backoff interval value such that the channel is systematically assigned to the node before the other ones. The proposed scheme defined by some modifications of the wireless protocol consists in improving the resolution mechanism by implementing some control capabilities to the node which receives the network packets.

• In a regular scenario, the sender node which wants to transmit packets to a receiver node is the node waiting during the shortest backoff interval.

• In the corrective scheme, the receiver node is capable to monitor the packet traffic generated by the sender nodes and to detect among them a misbehaving node. The receiver identifies the deviations from the protocol by monitoring the frequency and duration of the access channel by sender nodes.

If a sender node deviates from the wireless protocol, the receiver node can in a first time penalize the sender by assigning him a larger backoff interval. If the
sender node definitively does not respect the protocol, the receiver can refuse all the transmissions initialized by the misbehaving node.

The corrective scheme allows to improve a fair share of the wireless channel and facilitates the detection of misbehaving nodes by modifying the contention resolution mechanisms of the 802.11 wireless protocol. However, the solution focuses on a specific misbehavior detection and does not address for instance the case of misbehaving nodes using multiple physical addresses to acquire more bandwidth resources.

![Figure 11: Cooperation stimulation with a virtual currency](image)

### 3.3.2 Routing control

An efficient cooperation model cannot be limited to the medium access layer but must also be defined at the higher network layers. At the routing layer, a control scheme is introduced in [23] in order to detect and limit the impact on the routing process of mobile nodes that agree to forward network packets but fail to perform the task.

This control architecture is composed of two main components both deployed over network nodes: a watchdog capable to monitor failing network nodes and a pathrater capable to mitigate routing misbehavior.

- The watchdog detects misbehaving nodes according to the following mech-
anism. After a mobile node forwards the packets to an intermediate node, it listens to the intermediate node to check out that this node continues the forwarding process. The watchdog component maintains a buffer of the recently sent packets and compares it to the network packets emitted by the intermediate node. If a network packet is kept in the buffer more than a predefined timeout, the watchdog considers that the intermediate node fails to perform packed forwarding and sends in this case a notification message to the source node.

- The pathrater component receives the notification messages and maintains a rating of each network node to define their capability to forward network packets. High rating values are equivalent to misbehaving nodes. Both misbehavior rating information and regular link layer reliability information are combined to define a new route in the ad-hoc network.

The routing mechanism is improved by taking into account the behavior of intermediate nodes and by avoiding the definition of new routes through misbehaving nodes. However, the watchdog-pathrater model has some limitations. In particular, the misbehavior detection may be biased because of the physical environment conditions such as ambiguous collisions and limited transmission power.

### 3.3.3 Economical model

An economical model is proposed in [10, 11] at the service layer in order to both stimulate the cooperation and prevent the network overloading. A virtual currency called nuggets is introduced to enforce service availability and in particular to reward the nodes participating in the packet forwarding service. Two economical models are defined in this approach: the packet purse model and the packet trade model.

- In the packet purse model, the packet forwarding service is covered by the source node according to the destination node to reach. The source node loads the network packet with a sufficient number of nuggets. During the network packet route, each intermediate node charges one to several nuggets from the packet for the forwarding service. The deployment of this model is limited by the problem of avoiding an intermediate node to charge all the nuggets loaded to a packet, and also to the difficulty of estimating the number of nuggets required to reach a given destination.

- In the packet trade model, the packet is traded by intermediate nodes in an hop-to-hop manner and does not require to carry nuggets. The packet to be forwarded is processed by an intermediate node in two steps. The packet is first bought from the previous node for a given nugget price and then is sold to the next node for an higher price. In this scheme, the total price of the forwarding process is payed by the destination node and the price of the forwarding process is not required to be estimate by the source.
node. However, the Packet Trade Model does not prevent a network node to overload the network by flooding.

4 Conclusions

Ad-hoc networks are a potential solution for the increasing need of mobility and ubiquitous computing. These self-configuring networks are spontaneously deployed by a set of heterogeneous nodes of various capacities and resources, without requiring any preexisting infrastructure. The dynamic and distributed nature of ad-hoc networks raises new challenges towards monitoring and controlling them. New management paradigms capable to cope with the constraints of such unreliable networks were presented in this chapter. They were completed by the description of different management application domains including network monitoring, auto-configuration of IP addresses and control of node cooperation.

From an organizational point of view, several models can be considered for the purposes of ad-hoc network management. The management approaches were classified according to the organizational model in figure 12. The approaches aim at reorganizing the management plane with different degree of decentralization. The figure shows the predominance of two antagonist models for managing ad-hoc networks:

- the centralized hierarchical model: the centralized model simplifies the management operations by considering a single point of control but suffers from communication bottleneck with the central manager. The use of intermediate nodes in a hierarchical scheme reduce the management load for the central manager.

- the distributed (flat) model: it provides better scalability performance by using a set of distributed managers. However, it requires a minimum degree of autonomy from the network nodes, such that trust and convergence issues are posed.

From a chronological point of view, the figure 12 shows the publication date of each management approach. These approaches are relatively recent, since the first cited approach is ANMP which was published in 1999. Historically, the concept of ad-hoc networking is not new, since the first infrastructure-less multi-hop network was proposed in the 1970s with the ALOHA project [24]. However, the development of telecommunications has augmented the interest in ad-hoc networking for civil applications such that an impressive number of routing protocols were proposed in the last ten years. The recent standardization of ad-hoc routing protocols clarifies the context of ad-hoc networking for defining management approaches. It defines an easier way for the piggybacking and the instrumentation of ad-hoc routing protocols for management purposes.

From a functional point of view, the areas covered by each management approach are synthesized in figure 13. The traditional classification defines
five FCAPS functional areas: fault, configuration, accounting, performance and security. Fault management is not a well-covered area. Issues related to observability are challenging: fault management can be hindered by the impossibility to observe a given node in ad-hoc networks. A node that does not reply to legitimate polling in an fixed network can be considered as not functional. In an ad-hoc network, a node might not be reachable because it is moving and is out of reachability, or because it is not functioning properly. Configuration management is a relatively well-covered area. In particular, an analytical model has been defined by [9] to quantify the scalability performance of various configuration scheme in ad-hoc networks. Accounting management is the less covered area: ad-hoc networking does not fit within traditional accounting schemes. An ad-hoc network relies on the common sharing of resources among devices such that incentive model must be preferred to stimulate the cooperation. The performance management and the security management could be further re-
searched when quality-of-service and security mechanisms will be defined in the lower network layers.

<table>
<thead>
<tr>
<th></th>
<th>Fault</th>
<th>Configuration</th>
<th>Accounting</th>
<th>Performance</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANMP</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>GUERRILLA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Decentralized</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>COPS-PR PBM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRAMA PBM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Node-to-Node PBM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Middleware</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>for Mgmt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Probabilistic Mgmt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>WANMON</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>DAMON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>MANETconf</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROPHET</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Misbehavior Mgmt</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Watchdog-Pathrater</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Virtual Currency</td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
<td>✔</td>
</tr>
</tbody>
</table>

Figure 13: Classification by management functional area

Ad-hoc networks are commonly considered as a particular type of networks. However, the constraints of future networks may reverse the tendency: all the optimization approaches defined for ad-hoc networking could be applied to the networks of the future. The growing dynamics of these ones might generate an edge network management crisis. Consequently as it was pointed out by [19], it is of major importance for the future to define a scalable ad-hoc routing protocol that can manage itself automatically in order to tame this crisis.
References


