Policy-based Self-Management of Hybrid Ad hoc Networks for Dynamic Channel Configuration

Antonis M. Hadjiantonis, George Pavlou
Centre for Communication Systems Research
Dept. of Electronic Engineering, University of Surrey
Guildford, UK
{initial.surname}@surrey.ac.uk

Abstract— **Wireless networks have become a ubiquitous reality and ever more surround our everyday activities. They form and disappear around us spontaneously and have become new means for social interaction. At the same time, their increased complexity and heterogeneity have become barriers to their wider adoption and ease of use. It has become clear that self-management capabilities are vital for truly pervasive network management. To cater for the needs of both service providers and users, we propose a policy-based solution based on the ad hoc networking paradigm. By allowing limited infrastructure support, the notion of "hybrid ad hoc networks" emerges as a flexible extension to today's networks. A Distributed Policy Repository provides the support for policy-based device management and devices can autonomously adapt by examining local conditions and taking corrective actions in real-time. We design a system with self-management capabilities, able to self-configure and self-optimize connectivity parameters. Our aim is to simplify the management of hybrid ad hoc networks, enabling their coexistence and integration with wireless local area networks (WLANs). Implementation and deployment on a wireless testbed have demonstrated significant performance improvement for ad hoc networks in the presence of interference. Our experiments have measured a 33% goodput increase, after switching to a non-interfering channel in real-time.**

Keywords: policy-based management, hybrid ad hoc networks, LDAP, ubiquitous systems, self-management, interference, 802.11.

I. INTRODUCTION

Pervasive management is receiving intense interest from academia and industry, aiming to simplify and automate ubiquitous network operations. Synonymous to pervasive, autonomic management aims to vanish inside devices, relieving users from tedious configuration and troubleshooting procedures. Ideally, autonomic elements have self-configuration, self-optimization, self-protection and self-healing capabilities. When combined, these capabilities can lead to adaptive and ultimately autonomic systems. In reality, the deployment of ubiquitous networks is withheld from several obstacles that need to be overcome in order to realize such a vision. This provides motivation for our work to realize a system with self-management capabilities, in an effort for gradual transition to pervasive management.

Among various wireless technologies, Mobile Ad hoc Networks (MANETs) have received intense interest, especially from the research community. This interest however has not led to significant industrial exploitation or widespread adoption. According to [1], the major reason for the negligible market impact of the "pure general-purpose MANET" paradigm is the lack of realism in the research approach. As a result, MANETs are normally deployed in labs or by a few experienced users. To avoid such pitfalls, we base our research on realistic assumptions and tightly couple our design with implementation and deployment on a wireless testbed. We adopt the notion of "hybrid mobile ad hoc networks" [1] by relaxing the main constraints of pure general-purpose MANET, i.e. we consider the deployment of a network that consists of user devices with limited infrastructure support and connectivity. This assumption allows our design to be applied to several interesting paradigms and cases studies. We refer to this paradigm as *hybrid ad hoc* for the rest of this paper.

The deployment of *hybrid ad hoc* networks suffers from limitations in wireless link connectivity and capacity, due to the design of Physical (PHY)/ Data Link (MAC) layers and the wide use of TCP/IP which is optimized for fixed networks. The capacity and throughput are limited and severely degrade as the user population and number of hops grow [2]. Intermittence and interference amplify the problem, since enabling wireless technologies need to share the same spectrum and ISM (industrial, scientific and medical) frequency bands are by definition subject to interference. In spite of these drawbacks, the percentage of ad hoc network in cities worldwide accounts for an average 10% of total WLAN deployments [3], reaching a 13% in Paris. In addition, the results of field measurements during CeBIT in 2006 (trade show for Telco and IT), have counted 291 wireless connections of which 42% were in ad hoc mode [3]. Based on these facts, we argue that there is an increased demand for self-management of ad hoc networks. By facilitating easy and efficient deployment of ad hoc networks, we can take advantage of MANET routing protocols and mesh principles to deploy hybrid ad hoc networks, on top of which services can be provided.

We attempt to tackle the mentioned management problems of hybrid ad hoc networks using a policy-based approach and providing nodes with self-management capabilities. The basis of our solution is a context-aware policy-based framework [4]. The work presented here significantly extends and enhances the aforementioned framework, since it addresses a new application domain and evaluates the design on a wireless testbed. We introduce new policies to further improve and extend the Policy Repository and include self-management capabilities. The implemented case study deals with the

The research work in this paper was partly supported by the EU EMANICS Network of Excellence on the Management of Next Generation Networks (IST-026854).

deployment of hybrid ad hoc networks and effectively reduces co-channel interference by enforcing policies for self-configuration and self-optimization.

The rest of this paper is organized as follows: Section II introduces the problems related to wireless networks, while Section III describes the designed policy-based framework to anticipate these problems with detailed policy examples. Section IV presents the evaluation results and measurements on a testbed implementation. We conclude the paper in Section V.

II. BACKGROUND

Different enabling technologies have been considered as the basis for ubiquitous networks. Mobile Ad Hoc Networks (MANETs) offer fast and cheap deployment without the need of existing infrastructure while emerging Mesh technologies attempt to combine the benefits of MANETs with the support of wired access points. We consider the hybrid ad hoc paradigm as a promising solution for the deployment of ubiquitous networks [1]. Managing hybrid ad hoc networks and MANETs in general is an extremely challenging task. If we depart from cases of special-purpose deployments such as emergency scenarios and military operations, these networks typically consist of heterogeneous devices deployed by their users spontaneously in order to serve a relatively short-term purpose, e.g. file-sharing, online gaming or internet connection sharing. These devices cannot be fully controlled from a network manager and this fact provides a fruitful ground for self-management solutions. Traditionally, network managers have authority over managed devices (routers, switches), but in ubiquitous networks, users own the managed devices (laptops, PDAs). Critical differences also relate to the timescale of condition changes and the fluidity of network topology. The heterogeneity of devices enlarges relevant problems.

Policy-based management (PBM) simplifies the complex management tasks of large scale systems, since high-level policies monitor the network and automatically enforce appropriate actions in the system [5,6,7,8]. In general, policies are defined as Event-Condition-Action (ECA) clauses, where on event(s) E, if condition(s) C is true, then action(s) A is executed. Policy-based management (PBM) approaches for wireless networks have been proposed in [4,7,9] and industry envisions autonomic computing as dynamically managed by business rules and policies [10]. In [7,11], cases are examined where no absolute control from an authority is accepted, discussing whether all policies should apply to all users and how their preferences should be respected.

The components of a PBM system are shown in Fig. 1 in UML notation. The Policy Repository (PR) is an integral part of every policy-based system because it encapsulates the management logic to be enforced on all networked entities. It is the central point where policies are stored by managers using a Policy Management Tool (PMT) and can be subsequently retrieved either by Policy Decision Points (PDP) or by one or more PMTs. Once relevant policies have been retrieved by a PDP, they are interpreted and the PDP in turn provisions any decisions or actions to the controlled Policy Enforcement Points (PEP). Although a PR is a centralized concept, various techniques exist to physically distribute its contents. The reasons for distribution are obviously resilience and load balancing [8,12,13]. Typical implementations of a PR are based on Lightweight Directory Access Protocol (LDAPv3, RFC4511 [14]) Servers, also known as Directory Servers (DS). We will refer to a DS with its directory content (i.e. policies) as a *directory*. A single point of failure would make policy-based systems vulnerable; therefore replication features of directories are often exploited. When designing a Policy Repository for the policy-based management of wireless networks, there exist additional requirements that need to be taken into account, e.g. tolerance to connection intermittence and multi-hop communications. These issues are examined in the proposed framework and motivate the design of a Distributed Policy Repository (DPR).

The basic connectivity settings for devices joining existing WLANs, e.g. public hotspots or home networks, are automatically provisioned by the controlling wireless access point (WAP). Lower levels (PHY/MAC) are automatically configured by the wireless hardware drivers, based on the WAP control packets (beacons). For ad hoc networks, the apparent obstacle is how to establish communication in the absence of a WAP. In general, one of the ad hoc devices assumes the role of a master, acting as a WAP for the rest of the devices. In most cases, initial MAC/PHY configuration is arbitrarily set at the master device by adopting default software driver and/or hardware dependent parameters. Because of the variety of software drivers and hardware-specific implementations, many wireless configuration problems arise during the initial MAC/PHY setup. The use of "*default*" settings may work for isolated networks, but in cases of simultaneous network deployments can lead to interference and performance degradation. Imagine a conference venue, where different groups attempt to form ad hoc networks for file exchange, using the *default* settings. Most likely they will use the same channel (frequency), causing severe interference to each other and throughput decrease.

We consider the family of IEEE 802.11 standards [15] for WLAN, since it is the most widely deployed technology. Devices based on 802.11(a,b,g,n) are operating in ISM radio bands and can arbitrarily use any of the defined channels for deployment. The design of appropriate MAC layer algorithms makes these technologies fairly tolerant to interference and noise, but this comes at a price. Speed and performance are sacrificed in order to allow multiple stations to share the same wireless medium, i.e. the available spectrum. CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocols attempt to reduce the collision probability by sensing the wireless channel and backing off if it is sensed busy. The classic problem of hidden terminal is quite common. An additional measure to prevent collisions is used, the RTS/CTS handshake (Request To Send / Clear To Send), but this introduced the exposed terminal problem [16]. The use of

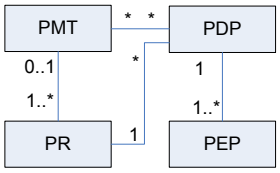

Figure 1. Basic components of a policy-based management system

Spread Spectrum modulation techniques can cause increased collisions due to interference between different channels (co-channel interference). This happens because channel spacing is overlapping for maximum frequency reuse. Depending on the enabling technology and modulation, different channels are likely to interfere with each other and interference increases the nearer the channels are. For example, 802.11bg technology defines 14 channels in the 2.4GHz ISM band, with center frequency separation of only 5 MHz and an overall channel frequency occupation of 22 MHz. Recommended deployments in FCC region use three non-overlapping channels (1,6,11)[17].

We argue that a policy-based system can be a future-proof solution for network management, since high-level policies need not be changed and their low level enforcement can be implemented independently as technology changes.

III. SELF-MANAGEMENT FRAMEWORK

The designed policy-based framework provides a highly distributed management environment that can cater for the configuration and optimization of user devices with minimum or no intervention. Management logic is encapsulated in policies that are transparently enforced to devices. Network Operators and Service Providers use the policy-based system to introduce the appropriate policies, aiming to set guidelines for the management of numerous user devices. Contrary to traditional management systems, the designed system does not require the mandatory enforcement of policies and tight control of managed devices. Instead, the system physically and logically distributes the policies among devices, making them available to vast numbers of users that voluntarily choose to enforce the relevant policies that would eventually relieve them from manual configuration.

To achieve the above, the DPR is designed, as an extension of the traditional PR. It is responsible for the distribution of policies in the network and for logically connecting the devices that collaboratively participate in the managed domain. These features were deemed necessary for the management of user owned networks, such as ad hoc networks, because of their spontaneous nature and the different ownership relation between networked devices and the network manager. In dense WLAN deployments (e.g. conferences, stadiums), users manually initiate ad hoc networks without relying on any infrastructure support. This results in poor performance and interference problems among WLANs, even regulatory violations. By making available appropriate policies in the DPR, user devices are assisted by receiving guidelines that transparently configure the ad hoc network, choosing the best available wireless channel to avoid interference and dynamically switching channels if performance degrades.

A. Distributed Policy Repository

The motivation for a DPR lies in the need for provisioning large-scale user-owned networks without the need for over-provisioning management resources, e.g access points, bandwidth or human effort. Because the deployment of such networks varies significantly in terms of spatial and temporal parameters, accurate planning and pre-provisioning is extremely difficult. Hence we propose the distribution of management tasks among PDPs hosted on user devices and based on policy guidelines stored in DPR. The DPR is a set of distributed and/or replicated LDAP *directories* (replicas), configured to store policies. Our design is based on the advanced replication and distribution features of modern LDAP servers. The innovation lies in the adoption and customization of such features for policy-based management in a wireless environment.

The diverse nature of wireless networks prevents the unmodified adoption and deployment of a Policy Repository (PR) using the various techniques targeting fixed networks. This motivates our research efforts for an enhanced PR, the Distributed PR. The policy-controlled DPR idea was introduced in [4] where different replication states were enforced depending on the network mobility. The work presented here further extends and enhances those concepts with sophisticated policies and applies them to a new domain. Depending on user density and population, devices are organized in clusters and each Cluster Head hosts a PDP, facilitating the policy provisioning for its cluster's PEPs. The selected Cluster Heads collaboratively share management tasks as well as the hosting of the DPR [4,18]. In order to decide where to place the DPR replicas, all Cluster Heads (PDPs) execute a special set of policies that combines a-priori knowledge of localized events (e.g. scheduled sport event) with dynamic real-time context information (e.g. processing load or free memory of each PDP). Different placement algorithms can be integrated in the implementation of policy actions, resulting in a customizable deployment of the DPR overlay. The relevant policies for DPR management are outlined in Table I, while in Section IV.A. more details are provided.

The coordination of distributed PDPs in a wireless environment is quite hard and remains an open research topic [13]. In the proposed solution, we transform this problem to the maintenance and deployment of the DPR by exploiting standardized LDAP operations and replication features. In this way, the DPR glues together the distributed PDPs and offers a logically uniform view of network management objectives through policies. Each DPR replica controls a configurable number of PDPs and each PDP is responsible to discover a replica for retrieving of policies and updates. The adopted pull-based approach relieves the DPR replicas from tracking PDPs and their operation is not affected from the intermittence of connections or the fluctuating number of PDPs. The proposed policy-based deployment is shown in Fig.2, in contrast with traditional design. A dashed horizontal line separates devices from operators' network, while thin dashed lines depict backup components.

TABLE I. DPR MANAGEMENT POLICIES

P	*Event*	*if {Conditions} then {Actions}*
a	chkDPR	if{$t_{Weekend}$}^{countPDPs($area_1$)/countDPRs($area_1$)>thr_1} then {locatePDPs($area_1$)}, {selectDPRhost(algorithm $_a$, $context_1$)}, {deployDPR(all)}
b	>>	if{$t_{Kickoff}$ - 2h}^{countPDPs($venue_1$)/countDPRs($venue_1$)>thr_2} ^ {countUsers($venue_1$)/countPDPs($venue_1$)>thr_3} then {locatePDPs($venue_1$)}, {selectDPRhost(algorithm $_b$,$context_2$)}, {deployDPR(service1,service2)}

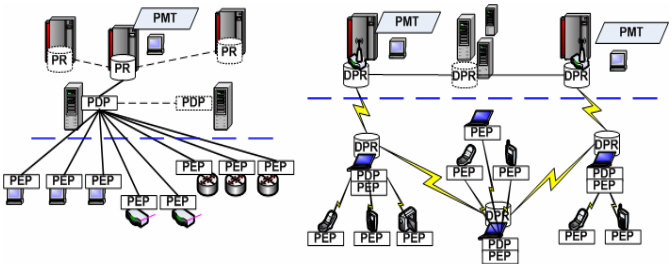

Figure 2. Traditional (left) Vs Proposed (right) PBM deployment

One of the innovative features of the proposed DPR design is the ability to deploy and maintain special purpose partial replicas of the Policy Repository. These replicas provide a partial view of network policies and can relate to a specific service or location. Accordingly, attached PDPs are responsible only for the enforcement of a policy subset and can be dynamically deployed to provision time-based events or local conditions. This feature can be employed when there is a need for localized control in areas with dense user population, such as a conference site or a stadium. In such cases, while node population (i.e. users) increases, the management system can deploy special-purpose DPR replicas and accordingly more PDPs that will be responsible for the distributed enforcement of specific management tasks. Special policies (Table I, P.b) guide the deployment of partial DPR copies, based on a-priori knowledge of localized events (e.g. sport event) and real-time context information (e.g. memory).

B. Hybrid Ad hoc networks self-management

To fully exploit the benefits of policy distribution in wireless networks, we select the case study of hybrid ad hoc networks for experimentation. We design the policies and algorithms necessary for the deployment of such networks and in the next section evaluate their performance and applicability through testbed implementation. The deployment of ad hoc networks and their coexistence with managed WLANs has not received enough research interest, since in most cases the assumptions are that an interference-free area is available and all ad hoc stations communicate using the same channel. We argue that by facilitating a predictable and controlled ad hoc network deployment, the performance of both managed WLAN and ad hoc networks can be significantly improved. Our solution can be deployed on top of existing and future access networks using a technology-independent PBM layer. The approach spans among different architecture layers of the protocol stack, exploiting cross layer principles but at the same time preserving the layers' modularity. This paradigm was deemed necessary, since the applicability domain of hybrid ad hoc networks is based on a majority of off-the-shelf end-user devices and only a few special purpose devices, e.g. mesh routers. In addition standards conformance is important for any to solution to be applicable.

Inter-layer communication is used between MAC and Application layers, aiming to make the PBM system aware of the wireless channel conditions and provide a feedback mechanism for policies. Based on specified application events (e.g. reduced throughput), the triggered policies can initiate relevant procedures that with the inspection of MAC layer headers provide feedback to the system and possibly trigger further policies to correct the problem. A "closed control loop" management system is thus formed, adding a degree of autonomy. There are two important advantages with the adoption of this approach. First, by using a policy-based design, the system is highly extensible and easily configurable. Policies can change dynamically and independently of the underlying technology. And second, by implementing decision logic at the Application layer, modularity is preserved without modifying the MAC protocol. Policies and extracted inter-layer parameters relieve the MAC layer from additional computations since inter-layer communication is only used when needed. The alternative solution of implementing the decision logic within the MAC layer would produce a new proprietary MAC layer. In this case, the benefit of increased speed is offset by the lack of flexibility and adaptability. Our implementation and testbed measurements have indicated tangible benefits from the adopted design.

As mentioned already, today the deployment of ad hoc networks is becoming a popular solution for spontaneous networking and quick network setup [3]. Unfortunately, user experiences have been disappointing, mostly because of difficulties in setup and poor performance. We have identified two potential obstacles that need to be overcome in order to make the deployment of ad hoc networks easy, efficient and safe: (1) interference between newly created ad hoc networks and existing WLANs, and (2) regulatory conformance of ad hoc networks' deployment. End-users have no need to be aware of channels and regulations, as long as they are connecting to infrastructure-based WLANs, regardless of their geographic area. The problems described are bound to ad hoc networks, since it is up to the initiating device to select a channel for deployment. In addition, it would be useful to ensure that roaming users are conforming to regional regulations with minimal inconvenience. We attempt to propose solutions to the above problems based on the designed policies for a PBM system, shown in Table II. Further explanation and parameters setup are provided in the following Evaluation Section

1) Interference between ad hoc and WLAN networks: Interference between deployed ad hoc networks and existing infrastructure-based WLANs, as well as interference with already deployed ad hoc networks in the same area is the main reason for the disappointing performance of ad hoc networks and it can lead to severe problems in the throughput and coverage of collocated infrastructure-based WLANs. Devices operating in ISM bands can arbitrarily use any of the defined channels and should be able to cope with interference from devices competing to access the same unlicensed bands. The MAC layer can be fairly tolerant to interference and noise at the cost of speed and performance. Choosing a random channel is likely to have a detrimental effect on the ad hoc network performance. The above problem has been verified by testbed measurements. To tackle this problem, we design policies P1 to P8 (Table II) that exploit MAC layer information for the initial configuration as well as the dynamic adaptation of the occupied wireless channel.

TABLE II. HYBRID AD HOC NETWORKS SELF-MANAGEMENT POLICIES

P#	*Event*	*if {Conditions} then {Actions}*
1	Init_new_adhoc	if {ready} then {scanChannels()},{generateScanComplete(results)}
2	ScanComplete(results)	if{otherWLANdetected=true}^{FC:=freeChannels(results),FC=true}^{PC:=preffered(FC,*ch_list*),PC=true} then{optimizeChannel(PC,$algorithm_1$($criteria_1$))}
3	>>	if{otherWLANdetected=true}^{FC:=freeChannels(results),FC=true}^{PC:= preffered(FC,*ch_list*),PC=false} then{optimizeChannel(FC,$algorithm_2$(criteria$_2$))}
4	>>	if{otherWLANdetected=true}^{FC:=freeChannels(results),FC=false} then{optimizeChannel(all, $algorithm_3$ ($criteria_3$))}
5	NewWLANdetected	if {*dyn_adapt*=true} then {generateStartAdapt(newWLANinfo)}
6	LinkQualityCheck	if{LinkQuality<thr_a}^{*dyn_adapt*=true} then {generateStartAdapt(cachedWLANinfo)}
7	StartAdapt(WLANinfo)	if {channel_distance(WLANinfo,current) < *dist*}^{*app_specific_metric* < thr_b) then{scanChannels()},{generateAdaptChannel(results)}
8	AdaptChannel(results)	if {results_evaluation()=true} then {channel_switch(all,$algorithm_4$($criteria_4$))},{verify_switch()}
9	SystemBoot	if {region=FCC} then set_criteria(approvedChannels[$list_1$])
10	>>	if {region=EU} then set_criteria(approvedChannels[$list_2$])

2) Regulatory conformance of ad hoc networks deployment: Although this issue is rarely addressed, it is indirectly affecting the popularity and usability of ad hoc networks. Users attempting to deploy ad hoc networks may be breaking the law, especially if their devices have been configured with the default settings of a different geographic area than their current. For example, the regulatory domain of Japan allows the use of all 14 defined channels of the 802.11b/g standards for the deployment of WLANs. For most devices used in this region, the default channel for ad hoc deployment is channel 14. However, the rest of the regulatory domains, e.g. Europe or Americas, explicitly forbid the use of channel 14 by WLANs. In the Americas, channels 12 and 13 are also forbidden, adding to the confusion of ad hoc network users. To prevent such problems, additional policies (Table II: P9,10) can be introduced by the regional network managers, which in turn influence the criteria for the policy-based channel selection described above (Table II: P2,3,4,8). For example, for P9:*list1* = 1..11 and for P10:*list2* = 1..13.

IV. EVALUATION - IMPLEMENTATION

In this Section we attempt to realize some critical features of the designed framework, based on the described case study for the configuration and optimization of hybrid ad hoc networks. First we present some implementation details and considerations regarding DPR and then we explain detailed measurements from the case study implementation and deployment on an experimental wireless testbed.

A. Distributed Policy Repository

For the implementation of the Distributed Policy Repository we have used OpenLDAP Server [19]. This selection was made because it is an open source implementation of a very fast and reliable LDAP v3 Directory Server for Linux. In addition, the minimum specifications required for running this server allow an extensive range of devices, including low-spec laptops to efficiently host a directory replica. The DPR consists of one or more Master read-write directories and several read-only directory replicas (shadow copies). Master directories are hosted and controlled by the managing network entities, i.e. Network Operator and/or Service Providers.

To achieve the above, we exploit OpenLDAP's replication engine to enable the policy-based distribution of replicated read-only directories (shadow copies) among the user devices, as well as partial copies for specific purposes (e.g. policies for multimedia services). OpenLDAP implements a Sync replication engine (syncrepl), based on the Content Synchronization Operation (RFC4533 [20]). Syncrepl engine offers client-side (consumer) initiation for replication of all policies or for a custom policy selection, relieving the providing directory (provider) from tracking and updating replicas. This functionality is very useful since the operation of a directory provider is not disrupted by the presence of consumers and can operate even when they are temporary disconnected because of wireless link intermittence. Upon reconnection, the directory consumers compare their current content with their provider's and retrieve any updates. Elaborating on the defined policies in Table I, a periodic chkDPR event causes the evaluation of conditions to determine if the current ratio of existing PDPs per DPRs or Users per PDPs in specified areas ($area_n$,$venue_n$) has exceeded the defined thresholds (thr_n). Additional time period constraints ensure triggering of policies when needed, e.g. on weekends or two hours before kickoff ($t_{Weekend}$, $t_{Kickoff}$ - 2h). The generic methods locatePDPs and selectDPRhost can use distributed algorithms for locating participating PDPs and for the best possible placement of replicated directories. The optimal placement solution is a computationally intensive task, hindered by the distributed nature of wireless systems and is out of the scope of this paper. In [4,18] we have described and evaluated a distributed algorithm based on context-aware heuristics to form a dominating set of nodes that share management responsibilities. The same approach is adopted for the implementation of algorithms for policies in Table I. The context parameter affects the used heuristics, by modifying the

weights of metrics used in the algorithm. Method deployDPR() is used to configure a replicated directory, part of the Distributed Policy Repository. First the directory configuration file (slapd.conf) is modified to define the replication provider and once the replica is initiated, it automatically connects and retrieves policies from its provider. Method parameters (all, service1, service2) define policy groups that will be replicated.

B. Adaptive hybrid ad hoc networks deployment

To illustrate our proposed solution we investigated wireless networks based on IEEE 802.11 standards, since it is the most widely deployed technology for WLANs. Let us assume that a user initiates an ad hoc network using a device supporting 802.11b/g. The device is set in IBSS mode (Independent Basic Service Set or ad hoc/peer-to-peer mode) and device-dependent software and hardware configure the transmission parameters. The device assumes the role of the wireless Access Point and begins to emit beacon messages advertising the existence of an ad hoc network on the statically defined channel. Other parameters are also advertised, such as the beaconing interval and any encryption methods used, thus enabling nearby devices to join the ad hoc network in a peer-to-peer manner. If we realistically assume deployment in a populated area, such deployment would imply the coexistence of various WLANs (either ad hoc or infrastructure-based) and possibly their interference. Choosing the default channel or even a random channel is likely to have a detrimental effect on the ad hoc network performance. The problems arise from the access to the wireless medium and three cases can be identified during the deployment of an ad hoc network on a channel: (a) the channel is already in use by other WLANs (b) adjacent or nearby channels are in use by other WLANs and (c) no nearby channels in use by other WLANs. In practice, cases (b) and (c) are difficult to be separated since co-channel interference depends on unpredictable environmental factors and is also technology dependent.

The above cases were examined on an experimental testbed and measurements were obtained. We have deployed our policy-based solution that aims to dynamically assign the best available channel and autonomously adapt to changes in the wireless environment. To prevent the detrimental effects of interference, we used context information extracted from the headers of Layer 2 frames. One device used a wireless interface to passively monitor all packets it can hear (rf-monitor) and forwarded them to the monitoring policies for processing of the 802.11 MAC headers as well as the 802.3 LL headers. The drawback of this method is that the monitoring interface cannot be used for communication. However, the solution is still applicable if all devices have single interfaces. In promiscuous mode the device can still detect interference while actively communicating, though not as effectively as when using a second interface.

In order to assess the performance of our policy-based approach we used a wireless testbed to evaluate the prototype's performance. In addition, we used the testbed to measure the effects of interference between devices using the same channel or devices with varying channel distance. Experiments were performed in a confined indoor space, matching the typical conditions of the described case studies. Our experimental testbed (Fig. 3 and Table III) consists of 10 nodes: 2 laptops, 4 PDAs and 4 Internet Tablets. All devices are equipped with internal 802.11b wireless interfaces, while the two laptops have an additional PCMCIA external wireless card. For the configuration of the wireless interfaces, Linux scripts were used with wireless-tools. For monitoring the wireless channel we have modified the source code of airodump-ng, a popular open source 802.11 packet sniffer, part of the aircrack-ng suite (http://www.aircrack-ng.org). The modifications allowed us to view and dynamically use the captured information within the policy-based interface.

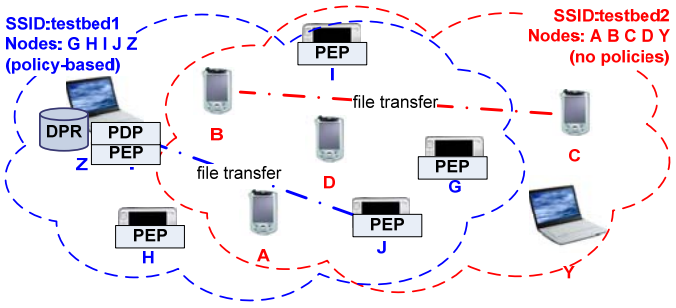

Figure 3. Ad hoc networks testbed deployment

For the purpose of our experiments, the devices were organized in two independent clusters of five nodes as seen in fig. 3. The clusters were setup using different SSID (Service Set Identifiers) in IBSS (ad hoc) mode. The manufacturers default channel for ad hoc networks creation was found to be Channel 1 (2412Mhz). The network speed (rate) was set to 11Mbps, to allow comparable results among nodes. One of the clusters had integrated PBM support and the cluster head deployed a PDP for the needs of its cluster. After the PDP had retrieved policies 1 – 8 (Table II) from the nearest DPR, it had accordingly instantiated policy objects (PO) for the monitoring and enforcement of decisions among cluster nodes. For evaluation purposes the PBM support was selectively used to measure its effect on the network performance.

Before demonstrating the implemented PBM system, we elaborate on the algorithms integrated in policies P2,3,4,8. Triggered actions *optimizeChannel* and *channel_switch* are called using as parameters the monitored measurements of a channel set (e.g. FC: free channels, PC: preferred channels) and the algorithm to be used for channel selection. For the purpose of our case study we have implemented an algorithm based on the weighted average (WA) of a channel metric (1).

$$WA(x)= \left[\sum_{i=1}^{n} (w_i x_i)\right]\left[\sum_{i=1}^{n} (w_i)\right]^{-1} \quad (1)$$

The criteria parameter of each policy specifies the channel metric (x_i) and weights (w_i) to use for the calculation of the WA, for each candidate channel. The flexibility of a PBM design is evident since different algorithms and criteria can be used to achieve the desired management objectives. For example, P8 used the monitored average packet/sec metric and

TABLE III. WIRELESS TESTBED CONFIGURATION

	Operat.System (Linux Kernel)	*Processor (MHz -family)*	*Ram (MB)*	*Wifi support*
Sony Vaio Z1XMP	Debian R4.0 (2.6.18)	1500 - Intel	512	802.11bg
HP iPAQ H5550	Familiar v0.8.4 (2.4.19)	400 - ARM	128	802.11b
Nokia N800	IT OS2007 (2.6.18)	330 - ARM	128	802.11bg

TABLE IV. POLICY-BASED CHANNEL ASSIGNMENT MEASUREMENTS

testbed1, 2 (channel)	**Goodput *testbed1*(Mbps)**	**Goodput decrease (%)**	**Downl.Time increase (%)**
1,1	3.48	-20.38	+20.00
2,1	2.92	-33.27	+46.67
4,1	4.26	-2.68	0.00
6,1	4.38	---	---

calculated the WA for all allowed channels, in order to select the one with the minimum value. As shown by testbed measurements, the described algorithm and parameters have identified a better channel to avoid interference.

1) Self-Configuration for Static Channel Assignment

Initially, we performed static measurements of the channel performance in the presence of multiple ad hoc networks with varying channel distance. According to this scenario, the two clusters would simultaneously attempt to initiate file transfer among peers of the same cluster. First, the two ad hoc networks were formed on the same default channel (Channel 1). This was possible by using different network names (SSIDs), namely "*testbed1*" and "*testbed2*". Afterwards, the same networks were deployed in different channels, with varying channel distances. The results of the average data download throughput (goodput) for each channel combination are shown in Table IV. What is worth noticing is that the goodput performance of ad hoc deployment in consecutive channels is even worse than deployment on the same channel by 13%. This can be explained by considering the MAC layer functionality: while on the same channel, all devices listen for Request To Sent (RTS) frames and back-off from using the channel and thus can avoid collisions. On the contrary, when nearby channels are used, frames from different channels are perceived as interference and increase channel noise, causing the MAC layer to retransmit lost frames and possibly reduce transmission rate to avoid excessive BER. As recorded by our measurements this effect is reduced the furthest apart the channels are, although is still noticeable even when "non-overlapping" channels are used (e.g. 6,1). This can be explained because of the devices' proximity which results in the near-far effect.

By enabling the PBM support for *testbed1*, the cluster head (node Z) ensures that policies 1-4 are applied during the initial phase of ad hoc deployment. After P1 scanned channels, P2 detects the presence of *testbed2* on channel 1 and the scan results indicated channels 2-10 as free (FC=true,PC=true). Since channel 6 of the preferred (non-overlapping) channels list was free, method *assignChannel* initiates the ad hoc network on the selected channel and the rest of the cluster nodes join using SSID *testbed1* on the same frequency. Effectively, the cluster is self-configuring the initial ad hoc deployment and this results in a 20.4% increase of average goodput when compared to using default channels and up to 33.3% increase for random channel assignment. File download duration is accordingly improved.

2) Self- Optimization for Dynamic Channel Switch

The second implemented scenario investigates the dynamic adaptation of hybrid ad hoc networks to anticipate interference and throughput degradation. Based on the topology of Fig. 3, we formed two separate ad hoc networks on the same channel (*testbed1* and *testbed2* on channel 1). Initially, no traffic transfers were performed between nodes. The scenario execution had two phases: (phase 1) ad hoc network *testbed1* initiates a file transfer between nodes, with cluster node J downloading media from cluster head Z and (phase 2) ad hoc network *testbed2* initiates another file transfer between nodes A and D. To evaluate our solution, two sets of the described scenario experiments were executed, one set with our PBM solution enabled and enforcing policies 5-8 and another set without any PBM functionality. The results (Fig.4) demonstrate a significant improvement in network performance when the proposed PBM solution is used.

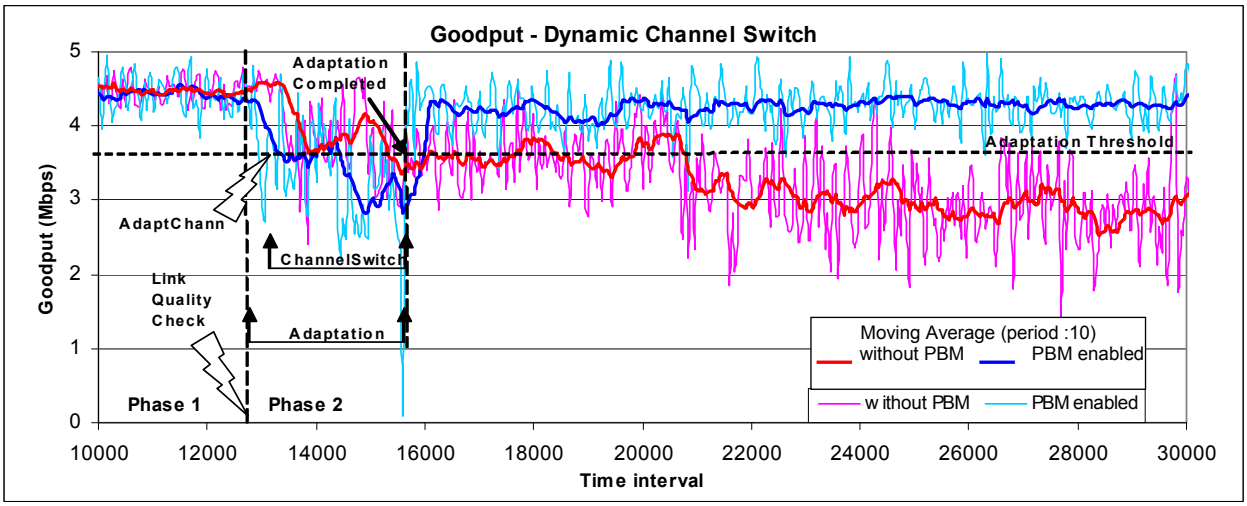

Figure 4. Testbed measurements of goodput using dynamic channel switch

The ad hoc cluster *testbed1* is self-optimizing by monitoring events and conditions, resulting in reconfiguration of the transmission channel to avoid interfering WLAN. When the competing ad hoc network (*testbed2*) initiates a file transfer, this results in increased collisions and missed frames for both clusters, which is reflected in reduced Link Quality reported by the wireless interface at node Z. Policy 6, triggered by LinkQualityCheck event, evaluates the moving average of LinkQuality as less than 50% (thr_a) and executes action generateStartAdapt to initiate the adaptation process for channel optimization. This triggers policy 7 that monitors the specified application's metric (Fig.4), in this case the moving average of goodput measurement for the file download between nodes Z and J (*app_specific_metric)*. The use of a moving average smoothes goodput fluctuations and prevents false triggering of adaptation policies. Once policy 7 detects the reduction of goodput below 3.67Mbps (thr_b), it acts by scanning the wireless channel, triggering policy 8 and passing scan results (event AdaptChannel). Policy 8 acts by executing *channel_switch* method using the weighted average algorithm ($algoritm_4$) with specified weights ($criteria_4$). The method indicates that a better channel is available and initiates dynamic switch of ad hoc network *testbed1* to channel 6. A channel switch period takes place, causing temporary disconnection of nodes from their cluster head. The measurements show that L2 disconnection and connectivity loss occur, however the effect on the ongoing file transfer between J and Z was temporary goodput reduction with a quick recovery to significantly higher goodput. In fact, when compared to the execution without PBM support, the described self-optimization resulted in an average goodput increase from 413.54 KB/s to 518.79KB/s (20.3% increase), reaching a peak increase of 33.5%. Also, average download time for a 46MB file dropped from 116sec to 50sec.

V. CONCLUSIONS

We have adopted a pragmatic view towards the management of ubiquitous networks, by adopting the hybrid ad hoc network paradigm. Using a policy-based design we have implemented on a testbed a system with self-management capabilities. The hybrid ad hoc system demonstrated self-configuration and self-optimization behavior, significantly improving its performance by dynamically switching channel to avoid interfering WLAN.

The designed policies were stored in a Distributed Policy Repository, in an effort to coordinate Policy Decision Points. We propose the distribution of management tasks among dispersed PDPs hosted on user devices and based on the policy guidelines stored in the DPR. The solution was demonstrated on a testbed based on 802.11b/g; the system however is extensible to cover more WLAN technologies, like 802.11a and 802.11n. High-level policies need not be changed since their low level enforcement can be implemented independently.

By facilitating easy and efficient deployment, we believe hybrid ad hoc networks can be the building blocks of ubiquitous systems. Based on real deployments, we plan to further investigate policies to accommodate the needs of mesh networks and future WLAN technologies. Assuming that policies do not change frequently, the design is applicable to stand-alone ad hoc networks as well; provided at least one DPR replica can be instantiated. As policy design becomes more sophisticated, robust conflict detection and resolution algorithms should be investigated in parallel. We believe a policy-based system can be a future-proof solution, where business objectives and user preferences will be encapsulated in policies. Eventually policies will vanish into systems, allowing users to enjoy ubiquitous networking.

REFERENCES

[1] M. Conti, S. Giordano, "Multihop Ad Hoc Networking: The Reality", IEEE Communications Magazine, vol.45(4) pp.88 - 95, 2007

[2] P. Gupta,P. R. Kumar, "The capacity of wireless networks," IEEE Transactions on Information Theory, vol. 46, pp. 388-404, 2000

[3] A.Gostev, R.Schouwenberg "War-driving in Germany - CeBIT2006" accessed Oct.2007, http://www.viruslist.com/analysis?pubid=182068392

[4] A.M.Hadjiantonis, A.Malatras, G. Pavlou, "A context-aware, policy-based framework for the management of MANETs", 7th IEEE Intl. Work. on Policies for Distributed Systems and Networks (Policy 2006)

[5] M.Sloman, E. Lupu, "Policy Specification for Programmable Networks", Proceedings of First International Working Conference on Active Networks (IWAN'99), Berlin, June 1999

[6] D.C. Verma, "Simplifying network administration using policy-based management" , IEEE Network, Vol.16,Iss.2, Mar-Apr.2002

[7] A.M.Hadjiantonis, M.Charalambides, G.Pavlou, "A policy-based approach for managing ubiquitous networks in urban spaces", IEEE Intl. Conf. on Communications 2007, Glasgow (ICC2007)

[8] R. Boutaba, S. Omari, A. Virk, "SELFCON: An Architecture for Self-Configuration of Networks", Journal of Communications and Networks,Vol.3. No.4, pp.317-323, 2001

[9] R. Chadha et al, "Policy Based Mobile Ad hoc Network Management" 5th IEEE Intl. Work. on Policies for Distributed Systems and Networks

[10] J. O. Kephart, D. M. Chess, "The Vision of Autonomic Computing", IEEE Computer, Vol. 36, No. 1, pp. 41-50, Jan. 2003

[11] M.Burgess, G.Canright, "Scalability of peer configuration management in logically ad hoc networks", eTransactions on Network and Service Management, Volume 1/ No.1 Second Quarter 2004

[12] IBM Corp., "Policy Management for Autonomic Computing, V1.2", accessed on Sep.2007, http://www.alphaworks.ibm.com/tech/pmac

[13] D.Chadwick et al, "Coordination between distributed PDPs", 7th IEEE Intl. Work. on Policies for Distributed Systems and Networks,2006

[14] J.Sermersheim, "Lightweight Directory Access Protocol (LDAP):The Protocol", RFC4511, Standards Track, Jun.2006

[15] IEEE 802.11 WLAN Working Group , accessed Sep.2007 http://grouper.ieee.org/groups/802/11/

[16] A. Jayasuriya, et al, "Hidden vs. Exposed Terminal Problem in Ad hoc Networks", Proc. of Australian Telec. Net. and App. Conf. , Dec 2004

[17] Cisco Systems, "Channel Deployment Issues for 2.4-GHz 802.11 WLANs", accessed on Sep.2007, http://www.cisco.com

[18] A. Malatras, A.M. Hadjiantonis, G. Pavlou, "Exploiting Context-awareness for the Autonomic Management of Mobile Ad Hoc Networks", Journal of Network and System Management, Special Issue on Autonomic Pervasive and Context-aware Systems, v.15(1), Mar.2007

[19] OpenLDAP Foundation, OpenLDAP 2.3 Directory Services, www.openldap.org

[20] K. Zeilenga, J.H.Choi, "The Lightweight Directory Access Protocol Content Synchronization Operation", RFC4533, Experimental, Jun.2006