A Knowledge Plane for Autonomic Context-Aware Wireless Mobile Ad Hoc Networks

Daniel F. Macedo[1], Aldri L. dos Santos[2], José Marcos S. Nogueira[3], and Guy Pujolle[1]

[1] Université Pierre et Marie Curie-Paris6, France
[2] Federal University of Paraná, Curitiba-PR, Brazil
[3] Federal University of Minas Gerais, Belo Horizonte-MG, Brazil
{Daniel.Macedo,Guy.Pujolle}@lip6.fr, aldri@inf.ufpr.br, jmarcos@dcc.ufmg.br

Abstract. Due to the emergence of multimedia context-rich applications and services over wireless networks, networking protocols and services are becoming more and more integrated, thus relying on context and application information to operate. Further, wireless protocols and services have employed information from several network layers and the environment, breaking the layering paradigm. In order to cope with this increasing reliance on knowledge, we propose MANKOP, a middleware for MANETs that instantiates a new networking plane. The *Knowledge Plane* (KP) is a distributed entity that stores and disseminates information concerning the network, its services and the environment, orchestrating the collaboration among cross-layer protocols, autonomic management solutions and context-aware services. We use MANKOP to support the autonomic reconfiguration of a P2P network over MANETs. Simulation results show that the MANKOP-enabled solution is applicable to more scenarios than the classic approaches, as the network adapts its query dissemination strategy to match the current conditions of the peers.

1 Introduction

With the evolution of networking and electronics, communication capabilities are nowadays implemented in a huge set of devices, such as notebooks, PDAs, cell phones or even household appliances. Those devices can provide multimedia and interactive content, such as pictures, audio, video or online gaming, requiring strict QoS guarantees from the underlying networking infrastructure. Further, although some of those devices face several connection challenges due to mobility, users expect a responsive and efficient experience at all times.

Those severe performance requirements have led to the creation of cross-layer protocols, which aim at optimizing the operation of the network by breaking the encapsulation and isolation principles, sharing information with several layers at the same time [1]. Further, ubiquitous mobile devices and networks should seamlessly blend with their surroundings, thus requiring that networking solutions and services adapt automatically to changes in their environment and in user needs (their *context*) [2]. Finally, the complexity and dynamics of wireless networks has spurred the research on autonomic network management solutions, which also require a view of the network as a whole [3].

The urgent need of information driven by *cross-layer protocols*, *context-aware services* and *autonomic networking* was hence the driving force for the conception of middlewares and mechanisms that promote information sharing. However, existing solutions tend to be specific to one of the three problems above, thus requiring multiple independent information services. Due to the use of disparate systems, information is not entirely shared, requiring replicated effort to maintain the same data on several systems.

In order to solve this problem, we advocate the use of a single distributed information service that stores any knowledge required to the operation of the network. This service can be seen as a new networking plane, called *Knowledge Plane* (KP), that deals with knowledge sharing. The KP lies between networking protocols and services[1], serving the information needs of both. Thus, all information concerning a service or protocol would be stored on this plane, spurring the free flow of information among services. The KP would replace the information repositories of autonomic management solutions and context-aware systems, storing context, knowledge, policies and past network behavior. It would also work as a cross-layer framework, exposing the interdependency among protocols and allowing the sharing of information using rich information models.

The use of a KP is natural in MANETs, since the resource restrictions of such networks impose the development of integrated, multi-layer context-aware solutions. For example, it is impossible to think of a routing or transport protocol that does not take link quality and battery life into account [4, 5]. Further, node movement and node and link failures demand that each protocol autonomically reconfigures some of its parameters. Since such networks are highly dynamic, due to the unpredictability of the hardware and of wireless links, existing information services for wired networks are not applicable because of their high overhead. The low bandwidth and high latency of the links make data synchronization too costly, thus services deployed over MANETs should rely on a partial knowledge of the state of the network [6, 7].

This paper proposes a middleware that instantiates a KP over MANETs. This KP, called MANKOP (MANet KnOwledge Plane), is adapted to the characteristics of the wireless medium and the resource restrictions of mobile nodes. It provides secure access to context and knowledge, as well as mechanisms for automatic information dissemination among MANET nodes. Due to the availability of more knowledge for the optimization of network protocols, services and autonomic management solutions, MANKOP allows a smoother operation of demanding applications such as multimedia and cooperative services. In order to show the benefits of MANKOP, we evaluated a MANKOP-aware Peer-to-Peer (P2P) network that automatically reconfigures the way queries are disseminated based on information stored on the KP. Simulation results show that the

[1] In this text we also consider management protocols and overlay networks such as P2P networks as services.

MANKOP-enabled solution is applicable to more scenarios than the classic approaches, as it presents a performance comparable to that of the classic query strategy more suitable for each scenario.

This paper is organized as follows. Section 2 describes the related work. Section 3 details the MANKOP architecture. Section 4 shows the case study and simulation results. Finally, Section 5 shows our conclusions and future work.

2 Related Work

In order to adapt the autonomic concepts to the networking world, Clark et al. added a plane to the OSI model, called knowledge plane [8]. Clark's knowledge plane manages the networking protocols (the *data plane*) using artificial intelligence algorithms that automatically react to changes in the network, in user requirements and in the environment[2]. Clark's as well as other autonomic management architectures [8, 9, 10] use context and application awareness to base their decisions upon rich models of the services and the environment. Those works usually assume that there is an information base from where knowledge can be stored and fetched in a scalable way. However, such information systems are still an open research issue [10, 11]. MANKOP is an instance of such a knowledge base, and thus could be support the operation of those architectures on MANETs.

Due to the scarce resources and the harshness of the wireless medium, *cross-layer design* is frequent in MANETs [12]. Cross-layering might induce a high layer interdependency, producing less modular code. To reduce those effects, several works proposed middlewares for MANETs providing standardized mechanisms for information sharing [13, 14, 15]. Usually, however, those middlewares are restricted to the information stored in each node, and employ simple information models or no model at all. Our work allows the representation of information using elaborate models and provides means to fetch information from any node in the network, thus improving the state of the art.

Conti et al. proposed event-based cross-layer interfaces [13]. Razzaque et al. created a middleware that is dynamically fed by the protocol stack, using *contextors* [14]. Each protocol has its contextor, which inserts and queries information from the middleware. In both approaches the stored information is limited to the current node. Further, there is still replication of information, since protocols cannot see each other's data unless they are explicitly exported. Winter et al. created a cross-layering middleware that divides information into local and global views [15]. The local view concerns the host node, while the global view represents the aggregated state of the network. The lack of information pertaining individual nodes precludes the use of those solutions for tasks such as management.

Other middlewares for information dissemination on MANETs support multi-agent systems. They limit the amount of information available to each agent

[2] Throughout this paper we refer to a knowledge plane as the plane responsible only for the dissemination and management of knowledge.

to reduce their resource usage. In the spatial programming paradigm, the data available to each node varies according to its location [16]. This model is handy for applications where nodes in a region tend to perform similar computations (e.g. for target tracking, detection of natural events), being simple to implement and highly scalable. The problem is that certain applications, such as Peer-to-Peer overlays, require information from nodes that are far away. TOTA (Tuples Over The Air) is an information dissemination middleware for multi-agent systems [17]. It employs the concept of stigmergy, that is, individual agents communicate by changing their local environment. Although simple and scalable, protocols and services must be modeled using stigmergic patterns, which may be hard for tasks such as distributed management. Also, this middleware may not scale well with complex dissemination and maintenance rules.

Haggle is an autonomic middleware designed for opportunistic MANETs [18], which are networks where nodes face frequent disconnections for extended periods of time. Nodes access remote information as if it were local, in a secure and simple way. This approach also has its drawbacks, such as the lack of trap-based access, which is extensively used in network management.

3 The MANKOP Middleware

The MANKOP middleware constructs a distributed information base, called Knowledge Plane (KP). The KP stores information and knowledge pertaining all the protocols and services of the network. Further, the KP acts as a collaboration layer, encouraging the adoption of context-aware, cross-layer protocols and services by providing interfaces for information dissemination and sharing. This plane lies between the service plane, composed by the services and management solutions, and the data plane, composed of devices and networking protocols, as shown in Figure 1. This organization allows services and autonomic managers to incorporate context-awareness or use the KP as a database for services and capabilities of the network, in order to compose more complex services. Further, protocols could use MANKOP to optimize their operation through cross-layering and context awareness.

In order to improve scalability, each node stores information concerning itself and also replicates data from local neighbors or other nodes with which it communicates. Information from distant nodes can be either queried directly from the nodes that produced it, or cached locally. However, the middleware does not guarantee that the cached copies are synchronized with the original data. The middleware also supports event notification services.

Data stored in MANKOP can be accessed using push/pull commands like GETS and TRAPS. Even though MANKOP does not specify a standalone data model, it is advisable to use one that supports rich information models. For instance, XML-RPC has been shown to be feasible in MANETs composed of modest PDAs [19]. Further, the content stored in MANKOP is updated in one of two ways. Either the services that produced the information specify an information dissemination policy, or they disseminate the information themselves. In

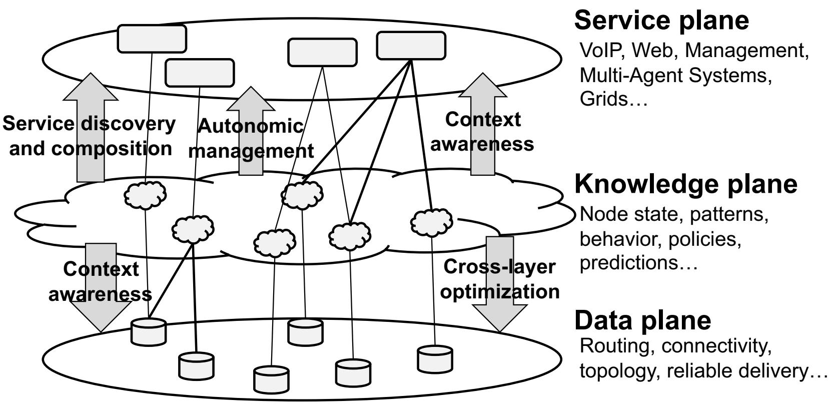
Service plane
VoIP, Web, Management, Multi-Agent Systems, Grids…
Service discovery and composition
Autonomic management
Context awareness
Knowledge plane
Node state, patterns, behavior, policies, predictions…
Context awareness
Cross-layer optimization
Data plane
Routing, connectivity, topology, reliable delivery…

Fig. 1. The three planes in autonomic networks

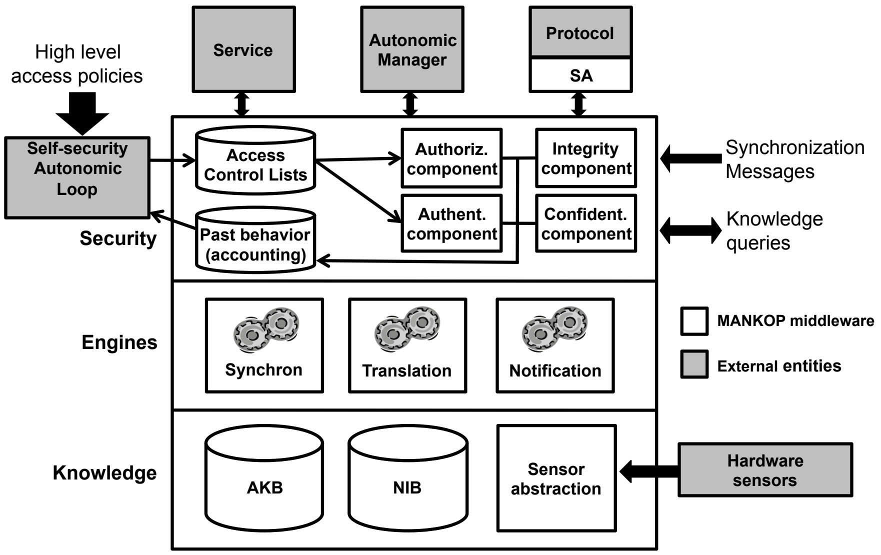

Fig. 2. The MANKOP module and its interaction with protocols and services

both cases, MANKOP provides interfaces that encapsulate the sending of messages, allowing piggybacking several pieces of information in a single packet. This reduces the control overhead of the network, which reduces energy consumption.

The organization of the MANKOP middleware is depicted in Figure 2. It is composed of three tiers: the *knowledge tier*, which stores the knowledge produced by the network, the *engine tier*, which performs several tasks over the stored knowledge, and a *security tier*, which secures the access to information.

3.1 The Knowledge Tier

MANKOP's knowledge tier has three components, which store knowledge produced by the application and provide abstractions for physical sensors. MANKOP should model information following a standardized information model, such as DEN-ng [20], DMTF's CIM [21] or RDF schemas.

The *Sensor Abstraction Component* deals with access to hardware sensors. In this paper we differentiate among hardware sensors and virtual sensors, as it is usual in the literature of context-aware and multi-agent systems [2]. Hardware sensors monitor physical devices, e.g. a GPS, a battery indicator, an accelerator. Virtual sensors, on the other hand, monitor software, providing readings such as the occupation of a packet queue, the number of TCP flows or the version of a program. The sensor abstraction layer provides a standardized interface for the access to hardware sensors, hiding device-specific interfaces and commands from other MANKOP components. For example, location data from diverse devices, such as a GPS or infrared-based systems, would be translated into a standardized representation for location, e.g. coordinates in degrees with their associated imprecision. The sensor abstraction component does not deal with virtual sensor data, which would be stored either at the AKB or at the NIB.

The *AKB* and *NIB Components* store knowledge. The Application Knowledge Base (AKB) stores knowledge coming from applications and high-level services, whereas the Network Information Base (NIB) stores information from networking protocols. This separation is due to performance reasons. Protocols have stronger real-time requirements and must respond very fast to events. Further, they usually require simple data, which can be easily represented using simple data structures (e.g. a floating-point number would suffice to represent queue occupation, signal-to-noise ratio and link reliability). Autonomic services, on the other hand, rely on complex information describing services, policies and products (a book, a VoIP connection). Hence, such services demand object-oriented, feature-rich representations of information. However, this capacity comes with a certain performance penalty. In order to allow fast access to information to lower layers, and at the same time provide support to rich information models to the applications and services, the knowledge tier stores the information in two different components.

The difference between AKB and NIB lies on the data models. While AKB should use object-oriented data models (e.g. a OO-DBMS), the NIB should use simple data models (e.g. ASN.1 or SMI). Lower level services would still access data on the AKB, providing that they accept a higher response time. The two data models are bridged using the translation engine described later. We chose not to define standard data models to allow each developer to pick one that fits the restrictions of the network.

In order to allow a higher degree of interoperability, the information stored in MANKOP should follow clearly defined information models. Using a standard representation, applications and/or protocols implemented by different vendors would be able to communicate (e.g. a free source FTP client could use information stored by a commercial client). Furthermore, a standardized representation reduces the effort to develop MANKOP compatible solutions, since developers are already familiar with such a representation.

3.2 The Engine Tier

The Engine tier of MANKOP offers engines for event notification, data synchronization and translation among different data representations.

The *Translation Engine* converts data among the different data models used in AKB and NIB to the format used for queries coming from protocols and services. Suppose, for example, that queries to information stored on MANKOP could be performed using XML-RPC. The translation engine would interpret the queries, identify where the information is stored (AKB or NIB), fetch it and marshal the data following a defined XML schema. Also, when receiving a SET command, MANKOP would employ the translation engine to translate the incoming information into its AKB and NIB representations. The translation engine should support two or more representations to data queries. For example, it could use XML-RPC for applications and services and ASN.1 for protocols. This would allow a faster response time for protocols that require a timely response (e.g. MAC and PHY level algorithms).

The *Notification Engine* watches the state of the local copy of data stored in the MANKOP middleware to create alarms whenever a certain condition happens. Applications can subscribe to events happening at the local node, or at other nodes. As an example, events could indicate low link reliability or an alarm for insufficient battery.

The *Synchronization Engine* automatically disseminates information based on simple rules. Those rules are defined by the application in terms of a distance in hops, a timeout and a periodicity of updates. Automatic synchronization is useful for services and protocols where the information required by each node is clearly divided in regions, for example as in routing and clustering [7]. The synchronization rules also allow nodes joining the network to easily populate their local AKB and NIB bases by issuing a synchronization request to its neighbors. Those nodes, in turn, will check the stored dissemination rules to determine which data will be sent to the arriving node. The engine uses broadcast messages, profiting from the broadcast nature of the wireless medium.

The synchronization of each piece of information is determined by a tuple (D, R, T, S). D defines the dissemination policy, that is, when the information should be disseminated. The possible values are *on every change*, *periodical* and *do not disseminate*. Next, the reach of the dissemination is defined by R, which defines the range in hops. Thus, if node X has data marked with a range of 5, its direct neighbors will cache it with a value of 4, and the neighbors of the neighbors will be marked with a value of 3. Each tuple also defines an Access Control List (ACL), stored in S, and a time to live, the value T. The time to live is reset on every update. In order to minimize the amount of packets sent, and thus reduce energy consumption, the synchronization engine should aggregate information updates as much as possible. For instance, defining a minimum interval among updates rather than disseminating changes as soon as they happen.

As mentioned before, some applications do not adopt a range-based neighborhood. In P2P applications, for example, neighbor peers may be several hops away, or even at the other side of the network. Thus, such applications would define themselves their propagation policy. Also, for applications where the propagated data is highly dynamic, each application would define what and when to send. For this scenario, MANKOP allows applications to either piggy-back their own data on MANKOP synchronization messages or to send their own MANKOP messages. In order to do so, the developer would implement a *Synchronization Agent* (SA) to select which and how information must be updated. This agent would autonomously manage the synchronization of all data concerning the service.

3.3 The Security Tier

Security is essential to MANKOP knowledge exchanges, once attacks to MANKOP could disrupt the operation of the entire network. The security tier assumes that a running PKI system exists on the network.

The *Confidentiality* and *Integrity* components perform encryption, decryption and signing of MANKOP messages. The *Authentication* component deals with the identification of services and protocols. Whenever a request arrives, the source must authenticate within MANKOP. Since in MANETs we should not rely on a central authority, authentication should be performed either by tickets created by a group of nodes [22] or by a PGP-like trust model. Finally, the *Authorization* component uses access control lists (ACLs) to define in a per-object granularity the access privileges of nodes, users and services. The definition of ACLs is not in the scope of MANKOP, which only enforces them. All the components produce traces (or past behavior), which are used for accounting as well as the automatic adaptation of the security tier by means of an external self-security component.

3.4 Interoperability with Regular Nodes

Due to the number of mobile devices already deployed, MANKOP-based nodes will have to interact with nodes that do not employ a KP, called *regular nodes.* Thus, MANKOP-aware protocols will provide both KP-based and traditional information dissemination methods. If no regular node is in the neighborhood of a MANKOP-aware node, it will employ the MANKOP approach. If a MANKOP-aware node receives traditional update messages (e.g. a routing message), it should respond using regular messages, based on the information stored on MANKOP. Only MANKOP-aware nodes in contact with regular nodes would operate on compatibility mode, as the others may communicate using MANKOP messages. Regular nodes, on the other hand, will discard MANKOP messages.

4 Case Study: Self-configuration of P2P Networks

In this case study we implemented a self-configuration management module for unstructured peer-to-peer (P2P) networks, since our previous work [23] showed that such networks perform better than structured ones on MANETs. Unstructured networks can be divided into *flooding-based* and *random walk-based*. In flooding-based networks, nodes forward queries to all their neighbors, bringing fault tolerance to protocols, however at a high energy and load cost. Random walk-based networks disseminate a fixed amount of queries, called *walkers*, which wander randomly around the network [24]. Usually flooding-based techniques have a higher hit rate (the amount of successful queries) due to the high number of messages sent. However, random walks perform better than flooding on high load networks.

In this case study we use MANKOP to support an autonomic manager that automatically selects the best query technique according to the conditions of the MANET. As in TCP, where the data rate is determined by the congestion of the network, the autonomic manager uses flooding when the network allows, switching to random walk (a more economic strategy) when the charge reaches a critical level. This adaptation indirectly optimizes the number of successful P2P file queries (the hit rate), the response time perceived by the user and the energy consumption by reducing the amount of collisions in the MAC layer. The manager requires information from different layers (the application and the MAC layers) and from several nodes, which justify the use of a middleware such as MANKOP.

A *MANKOP-aware Autonomic Manager* (MAM) is installed in all nodes. It divides the network in clusters, where the cluster-heads (CH) coordinate the monitoring of the network condition and also decide if the nodes on the cluster should change the employed query strategy. In order to create clusters, the MAM sets a periodic timer on all nodes. Upon its timeout, nodes choose if they will become CHs with a probability of 10%. Those nodes will then advertise themselves, so non-CH nodes can choose which cluster they will join. Next, the CH watches two events on all member nodes, as described below.

The decision to change the query strategy is based on the queue occupation (the amount of the packet queue that is being used) of a k-hop neighborhood, which is calculated using the information stored in MANKOP. Each node inserts in the KP its queue occupation as well as the aggregated queue occupation of its one-hop neighborhood (*ohocc*), using the formula $ohocc = max(max(N.occ), occ)$, where N is the set of all the one-hop neighbors of a node, and *occ* is the node's occupation. Next, each node calculates the two-hop occupation $thocc = max(N.ohocc)$ in order to obtain the maximum occupation in two hops. We used two hops since empirical results showed that this configuration performed better than zero (local) and one-hop knowledge. If *thocc* is below a threshold, the network does not seem to be congested, hence flooding should be employed. In this case, nodes produce a `FLOOD_THRESHOLD_EVT` MANKOP event. However, if the two-hop estimate is above a second pre-defined threshold, the network is saturated, and thus random walk should be used. To signal this to the CH, nodes produce a `RW_THRESHOLD_EVT` event. An operator defines the values of those thresholds.

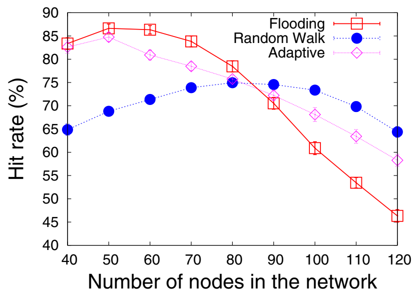

Fig. 3. Average hit rate

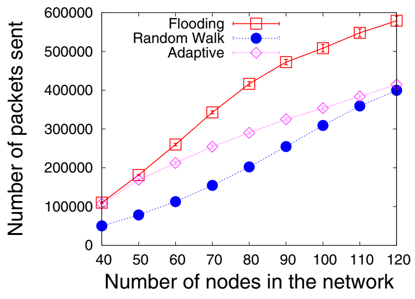

Fig. 4. Average number of query packets sent

The evaluation was performed on NS-2 over an IEEE 802.11b network with nodes having a Cisco Aironet radio [25]. The radio model supports adaptive modulation and coding, thus the transmission range varies from 355 to 1061 meters. We use a proactive routing protocol (DSDV). For flooding-based queries, we employ the Gnutella protocol, while for random walk we employ the model presented in [24]. Random walks use four walkers with a TTL equal to 25% of the number of nodes. For Flooding, the TTL was set to four, following the results in [23]. The `FLOOD_THRESHOLD_EVT` and `RW_THRESHOLD_EVT` events are triggered when the occupation is below 10% and over 50%, respectively. Likewise, the MAM changes from flooding to random walk if 40% of the nodes signal a `FLOOD_THRESHOLD_EVT` event within a clustering interval (set as 30s), and changes from random walk to flooding if 90% of the nodes signal the corresponding event. Those values were defined using empirical experiments. Results are averaged over 35 independent simulations and are plotted with a confidence interval of 99%.

Figure 3 presents the hit rate. The figure shows that flooding performs better on smaller networks, however on networks with more than 90 nodes, random walk performs better, as expected. This is due to the number of packets required for queries, shown in Figure 4. Flooding sends more than the double of packets than Random Walk, thus the network becomes congested faster. Hence, it would be recommended to use flooding for networks up to 80 nodes, and random walk for bigger networks. This is the idea of the Adaptive protocol, which employs the MAM described previously. The adaptive method had a performance between flooding and random walk techniques, performing up to 5% worse than Flooding on networks of up to 80 nodes, and performing from 1% to 6% worse than Random Walk on bigger networks. This behavior is due to the addition of the MAM, which reduces the amount of query packets in order to reduce collisions (Figure 5).

Figure 6 shows the percentage of nodes using each query strategy. When the load increases, the autonomic manager gradually reconfigures the nodes. Thus, for lighter networks most nodes use flooding, while nodes on busy networks usually resort to random walk. The adaptive solution has an intermediary energy

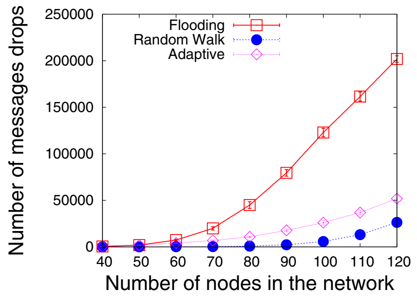

Fig. 5. Average number of packet drops

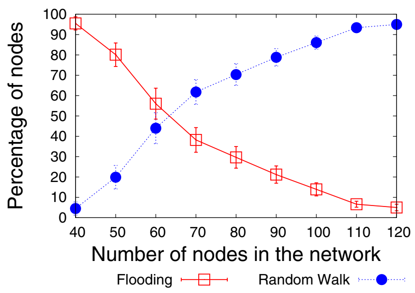

Fig. 6. Average percentage of nodes using each query strategy

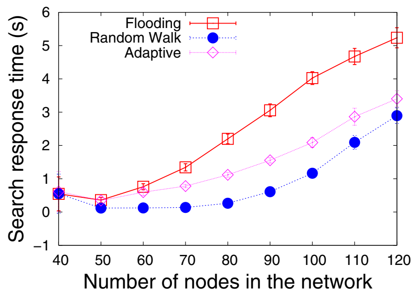

Fig. 7. Average query response time

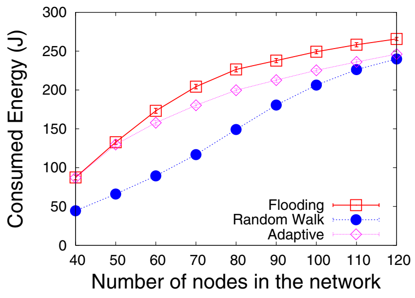

Fig. 8. Average energy consumption

consumption and response time, as shown in Figures 7 and 8. The response time and energy consumption of the Adaptive solution are not as low as that of Random Walk for two reasons. First, the operation of the MAM requires additional control traffic. Second, large networks still have a small amount of nodes running Gnutella, which increase the energy consumption and the response time. This mix of query strategies reduces the hit rate of the adaptive method when compared to pure methods. However, this degradation is acceptable once the autonomic solution has a more predictable response time and energy consumption.

5 Conclusions

This work presented a knowledge plane for ad hoc networks called MANKOP in order to face the challenges raised by multimedia, context-aware and autonomic services. Each node having a MANKOP module stores information from all layers and from other nodes of the network in one single abstraction, accessible to protocols and applications. Thus, protocols and services may use algorithms that take into account context, service and network information to improve their operation.

We showcased the benefits of MANKOP with a self-configuration manager that automatically adapts the query strategy on unstructured P2P networks. The insertion of a MANKOP manager has produced a protocol that is more adapted to a higher range of scenarios. The autonomic solution presents a hit rate, response time and energy consumption comparable to the best P2P protocol on each scenario. However, since the manager imposes a small control overhead, there is a small but acceptable performance degradation.

As future work, we will define the network-level information stored at the control plane and investigate the correlation of this information to improve the treatment of events such as node and link failures, security attacks, among others.

Acknowledgements

We would like to thank CNPq, an organization from the Science and Technology Ministry of Brazil, for partially funding this research.

References

1. Srivastava, V., Motani, M.: Cross-layer design: A survey and the road ahead. IEEE Comm. Mag. 43(12), 112–119 (2005)
2. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. International Journal of Ad Hoc and Ubiquitous Computing 2(4), 263–277 (2007)
3. Dobson, S., Denazis, S., Fernández, A., Gaïti, D., Gelenbe, E., Massacci, F., Nixon, P., Saffre, F., Schmidt, N., Zambonelli, F.: A survey of autonomic communications. ACM Trans. on Autonomic and Adaptive Systems 1(2), 223–259 (2006)
4. Akkaya, K., Younis, M.: A survey of routing protocols in wireless sensor networks. Elsevier Ad Hoc Network Journal 3(3), 325–349 (2005)
5. Katabi, D., Handley, M., Rohrs, C.: Congestion control for high bandwidth-delay product networks. In: ACM SIGCOMM, pp. 89–102 (2002)
6. Hollos, D., Karl, H., Wolisz, A.: Regionalizing global optimization algorithms to improve the operation of large ad hoc networks. In: IEEE Wireless Communications and Networking Conference, pp. 819–824 (2004)
7. Biskupski, B., Dowling, J., Sacha, J.: Properties and mechanisms of self-organizing MANET and P2P systems. ACM Trans. on Autonomic and Adaptive Systems 2(1-34), 1 (2007)
8. Clark, D., Partridge, C., Ramming, J.C., Wroclawski, J.T.: A knowledge plane for the internet. In: ACM SIGCOMM, pp. 3–10 (2003)
9. Niebert, N., Abramowicz, H., Malmgren, G., Sachs, J., Horn, U., Prehofer, C., Karl, H.: Ambient networks: an architecture for communication networks beyond 3G. IEEE Wireless Communications 11(2), 14–22 (2004)
10. Malatras, A., Hadjiantonis, A.M., Pavlou, G.: Exploiting context-awareness for the autonomic management of mobile ad hoc networks. Journal of Network and System Management 15(1) (March 2007)
11. Giaffreda, R., Pentikousis, K., Hepworth, E., Agüero, R., Galis, A.: An information service infrastructure for ambient networks. In: IASTED Multi-Conference on Parallel and Distributed Computing and Networks, pp. 21–27 (2007)
12. Kawadia, V., Kumar, P.R.: A cautionary perspective on cross layer design. IEEE Wireless Communications 12(1), 3–11 (2005)

13. Conti, M., Gregori, E., Turi, G.: A cross-layer optimization of gnutella for mobile ad hoc networks. In: ACM MobiHoc, pp. 343–354 (2005)
14. Razzaque, M., Dobson, S., Nixon, P.: A cross-layer architecture for autonomic communications. In: Autonomic Networking, pp. 25–35 (November 2006)
15. Winter, R., Schiller, J.H., Nikaein, N., Bonnet, C.: Crosstalk: Cross-layer decision support based on global knowledge. IEEE Comm. Mag. 44(1), 93–99 (2006)
16. Borcea, C., Intanagonwiwat, C., Kang, P., Kremer, U., Iftode, L.: Spatial programming using smart messages: Design and implementation. In: International Conference on Distributed Computing Systems, pp. 690–699 (2004)
17. Mamei, M., Zambonelli, F.: Programming stigmergic coordination with the TOTA middleware. In: International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 415–422 (2005)
18. Scott, J., Hui, P., Crowcroft, J., Diot, C.: Haggle: A networking architecture designed around mobile users. In: IFIP Conference on Wireless On-demand Network Systems and Services (January 2006)
19. Pavlou, G., Flegkas, P., Gouveris, S., Liotta, A.: On management technologies and the potential of web services. IEEE Comm. Mag. 42(7), 58–66 (2004)
20. Strassner, J.: DEN-ng: achieving business-driven network management. In: IEEE/IFIP Network Operations and Management Symposium, pp. 753–766 (2002)
21. DMTF, Inc.: DMTF Common Information Model (CIM) (March 2008), http://www.dmtf.org/standards/cim/
22. Luo, H., Kong, J., Zerfos, P., Lu, S., Zhang, L.: Ursa: ubiquitous and robust access control for mobile ad hoc networks. IEEE/ACM Trans. on Networking 12(6), 1049–1063 (2004)
23. Oliveira, L.B., Siqueira, I., Macedo, D.F., Loureiro, A.A., Nogueira, J.M.: Evaluation of peer-to-peer network content discovery techniques over mobile ad hoc networks. In: IEEE WoWMoM, pp. 51–56 (June 2005)
24. Gkantsidis, C., Mihail, M., Saberi, A.: Random walks in peer-to-peer networks: Algorithms and evaluation. Performance Evaluation 63(3), 241–263 (2006)
25. Cisco Systems: Cisco aironet 350 series client adapters (March 2008), http://www.cisco.com/en/US/products/hw/wireless/ps4555/products_data_sheet09186a0080088828.html