MANKOP: A Knowledge Plane for Wireless Ad Hoc Networks

Daniel F. Macedo*, Aldri L. dos Santos†, Guy Pujolle*, José Marcos S. Nogueira‡
*Université Pierre et Marie Curie-Paris6, France
†Federal University of Paraná, Brazil
‡Federal University of Minas Gerais, Brazil

E-mails: Daniel.Macedo@lip6.fr, aldri@inf.ufpr.br, Guy.Pujolle@lip6.fr, jmarcos@dcc.ufmg.br

***Abstract*— In mobile wireless ad hoc networks (MANETs), layering is frequently broken to cope with changes on the medium. Layering is also violated in order to implement autonomic behavior, which depends on correlating data from various layers to identify relevant events. This paper proposes a networking plane, called *MANET KnOwledge Plane* (MANKOP), that stores information concerning all protocol layers. This plane improves network performance, as protocols may employ a broader range of inputs on their algorithms. Moreover, being an information repository, MANKOP eases the deployment of self-optimizing and self-configuring mechanisms. To showcase our solution, we use MANKOP to build transmission power control aware protocols, which increases the throughput of the network by up to 280%.**

I. Introduction

Mobile devices are used in situations where there is no preinstalled infrastructure, or the existing infrastructure cannot be used due to catastrophic events. Hence, the devices must organize themselves to form a mobile wireless ad hoc network (MANET) acting as source and data sinks, but also as data forwarders. MANETs face several challenges that are not found in wired networks. Node mobility, the dynamism of the wireless medium and limited battery and resources must be considered on protocol design.

Due to the high dynamism of the physical medium, several protocols in the literature require the collaboration among layers. They are termed *cross-layer* protocols, as they violate the layered approach. The large amount of existing cross-layer protocols suggests that the layering model should be revised. At the same time, the concept of autonomic networking demands an entity that oversees the operation of the protocol stack as a whole [1]. This entity should evaluate the state of all networking layers in order to identify significant events and suggest actions based on the current configuration of the protocols. Thus, MANETs demand a mechanism to support collaboration among layers. Since existing protocols were not created with measurement and management in mind, the required information is hidden inside each protocol. Hence, we must develop mechanisms that make this knowledge available to all protocol layers and to autonomic solutions.

A Knowledge Plane [1] is one such mechanism. The *knowledge plane* (KP) is a horizontal plane that stores information pertaining the state of the network and all the protocol stack (the *data plane*). Further, the KP provides the substrate over which policy-based systems, autonomic management planes and cross-layer protocols will be built, as the KP abstraction simplifies the cooperation among layers and the creation of autonomic solutions. This is due to the concentration of the data from the protocol stack and the inter-layer communication interfaces in one place. Thus, a KP simplifies and ameliorates autonomic management solutions and networking protocols, as those might employ algorithms using input parameters that would otherwise be hard to obtain.

This paper describes a knowledge plane for MANETs, called MANKOP, that concentrates the information concerning the operation of MANETs. MANKOP implements mechanisms to retrieve and update the stored information, thus allowing protocols to improve their performance using augmented inputs. This paper extends the preliminary model proposed on [2], presenting the implementation of MANKOP over a case study. We use MANKOP to build a set of optimized transmission power control (TPC) aware protocols. TPC-aware protocols adjust the transmission power on each link in order to decrease energy consumption and contention. They also build routes that take into account the energy consumption and the data rate on each link. Such protocols require a tight cooperation among layers, thus being a showcase for MANKOP. Results show that the MANKOP-enabled protocols significantly improve the throughput and energy consumption of the network.

The rest of this paper is organized as follows. Section II depicts the related works. MANKOP is detailed in section III. Section IV shows the case study and the simulation results. Section V presents our conclusions and future work.

II. Related Work

Since cross-layer design is common in MANETs, Kawadia and Kumar describe the pros and cons of cross-layer design. They argue that cross-layering might induce a high layer interdependency, producing less modular code [3]. Several papers propose cross-layer algorithms to optimize network performance based on augmented information. Those propositions assume information availability, while in fact they are restricted to the protocols that produce them.

Cross-layer protocols suppose the availability of the information required from each network element, but no such thing

978-1-4244-2066-7/08/$25.00 ©2008 IEEE

exists nowadays. Hence, our work is complementary to such propositions, because MANKOP will provide the substrate over which policy-based systems, autonomic planes and cross-layer protocols will be built. Other works propose a plane to gather the information produced by all protocols. However, those solutions only store information concerning the current node, limiting their applicability.

Conti et al. proposed event-based cross-layer interfaces to notify protocols of significant events on other layers, simplifying the design of cross-layer protocols while at the same time keeping the benefits of layering [4]. The authors demonstrate their interfaces on a cross-layer version of Gnutella, where the neighbors of a peer are determined by the connectivity at the routing level. This approach has several limitations. First, there is still replication of information, since protocols cannot see each others' data unless they are explicitly exported. No push-based access is provided, and the information is not available to other nodes. Razzaque et al. create a new plane that is dynamically fed by the protocol stack [5]. Protocols use *contextors* to insert and query information from the KP. However, the stored information is limited to the current node.

Winter et al. created a KP for the POEM project [6]. This plane is divided into local and global views. The local view stores information about the host node, while the global view stores the overall state of the network. As an example, the global view could indicate that the average queue utilization is 10%. Each node periodically propagates its local view to its neighbors, which feed this information into aggregation algorithms to update their global view. Although more flexible than previous propositions, this KP still lacks information from individual nodes, which is essential to solving more complex problems such as transmission power control.

III. The MANKOP Knowledge Plane

MANKOP is a distributed KP, where each node stores information concerning itself and its neighbors. Unlike other proposals in the literature [1], [6], MANKOP is essentially a dumb distributed database. We opted to separate *intelligence* and *knowledge*, in order to make a scalable and generic solution. This separation has several benefits. First, since there is no protocol- or application-specific algorithms, the KP module is the same no matter which applications or protocols are employed. Second, MANKOP relies on others to define which information it stores and when they must be updated. Thus, each piece of information can be refreshed based on a much finer granularity and only when strictly necessary, while other propositions simply send all the changes that occurred after a given period of time. Finally, since there is no control hard-wired on the KP, we are able to change how a node or the entire network store and process information.

The MANKOP module installed on each node is composed of two blocks. The first block, called Networking-level Knowledge Plane (NKP), stores information on the layers one through six of the protocol stack[1]. The second block, called Application-level Knowledge Plane (AKP), allows secure queries from applications or from external sources. Figure 1 shows the organization of MANKOP and how it interacts with the data plane.

[1]From this moment on, *protocols* are the distributed algorithms that run below the application layer.

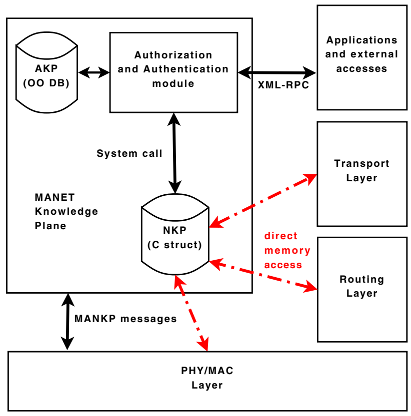

Fig. 1. The components of a MANKOP module and its interaction with the protocol stack.

As the NKP block provides information to the networking layer, it must be responsive and simple, because such information may be accessed many times per packet. Thus, we model the NKP as a C-like structure where protocols have full read and write access. This approach avoids costly operations, such as XML marshaling. MANKOP considers that layers 1-6 are trustworthy. It assumes that the OS verifies the authenticity and integrity of the code before loading it. To avoid race conditions, the access to each field or groups of fields is controlled by semaphores.

The AKP block stores the information pertaining to applications and overlay networks. It is object-oriented, allowing the storage of complex information. The AKP also acts as a proxy among the applications and the NKP, allowing read-only access to the information stored on the NKP and hiding sensitive information. All queries received by NKP are controlled by the Authorization and Authentication module. Applications are considered untrusted entities, and thus have read/write access to their own data and read-only access to data on the NKP. To allow a consistent view of application and kernel-level data, the AKP translates the information stored in the NKP into an object-oriented representation. Applications communicate with the AKP using XML-RPC queries. Using this mechanism, MANKOP allows queries from anywhere on the network and guarantees an homogeneous interface to applications.

The MANKOP module on each node periodically sends broadcast messages in order to update the information stored by neighbor nodes. Before delivering messages to the MAC layer, MANKOP requests the registered protocols to embed their own data, allowing each level of the stack to define what information must be sent. When the message arrives on the neighbors, their MANKOP module notifies the concerned layers. Those layers process the received information, extracting useful data and updating the information on MANKOP.

Figure 2 shows the classes used by the MANKOP module

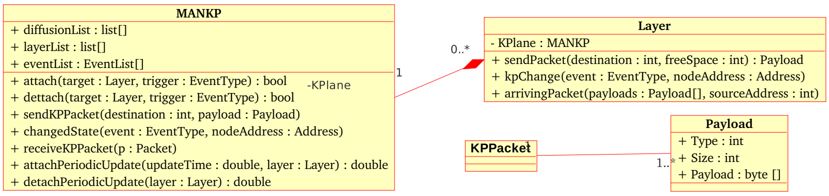

Fig. 2. Class diagram for the interaction among protocol layers and MANKOP.

stored in each node. We describe below how protocols interact with the NKP using the classes specified above. The same access patterns are used on the AKP, however they are performed using XML-RPC calls and requests must be approved by the Authentication and Authorization module. The class `MANKOP` stores network information in a public field, allowing push access and easy data update[2]. The method `kpChange` on the `Layer` class, which is a parent class for all layers, allows pull access. To activate pull queries, layers call the `attach` method, indicating which event they want to monitor. In order to update the stored information on each node, MANKOP periodically executes the `sendKPPacket` function, which calls the `sendPacket` function on all layers, allowing them to piggy-back their data. Upon the reception of a KP packet, MANKOP verifies the payload of the message, which might contain data from one or more layers. It calls the `arrivingPacket` function of the concerned layer based on the `Type` field the payload. The layer, in turn, processes the data and updates the information in the NKP. Protocols signal information updates by calling the `changedState` function. After all layers have treated their payloads, MANKOP notifies listening layers for change events.

The Authorization and Authentication module controls the access and modification of AKP data by applications and other nodes. By default, applications have read/write access to their own data, and no access to the data of other applications. If an application wishes to open the access to others, it can do so using access control lists. Those lists will allow access to machines, users or specific programs or classes or programs. NKP data must be controlled on a per-case basis. For example, link encryption keys must be hidden to most programs except to certified administrative routines, while the occupation of the link must be accessible to all programs. The access to NKP data by applications must be controlled on a per-case basis.

Each application, program or user must be authenticated to access information other than its own. This process will employ either certificates or the possession of a common secret. All connections must employ transport-level encryption to avoid eavesdropping, and the authentication expires periodically to avoid unauthorized access.

IV. Case Study: Transmission Power Control

To show the benefits of the proposed knowledge plane in concrete network situations, we chose a case where three communication layers (physical, MAC, and network) collaborate to reduce the energy consumption of the communication, using transmission power control (TPC) techniques [7]. TPC protocols aim to adjust the transmission power to minimize energy consumption. Whenever a station has data to transmit, it does so at the lowest transmission power necessary to reach the destination, consuming less power when compared to a static power configuration. Further, TPC algorithms can improve the latency of links with dynamic data rates, which are common on most PHY/MAC standards (e.g WiFi, WiMax, among others). The dynamic data rate is a consequence of the automatic selection of coding and modulation schemes, used to avoid the effects of interference and noise [8].

TPC algorithms can benefit greatly from the use of MANKOP, since they require the cooperation of the PHY, MAC and routing layers. The MANKOP-enabled TPC solution requires two changes in standard routing algorithms. First, we define the cost of a path (P_{ij} in our notation, the set of links traversed) as a tuple $[datarate, length, energy]$, where the $datarate$ is the minimum data rate of all the links involved, $lengths$ is the size in hops of the route and $energy$ is the sum of the energy consumed in all links. The parameter $energy$ is dependent on the data rate, since the energy consumed depends on the reception threshold used. Finally, in order to take the data rate and the energy consumption into account, the paths build must maximize the data rate, minimize the route length and energy consumption of the path, in this order.

The evaluation employed the NS-2 simulator extended with the TPC code of Kawadia and Kumar [3]. We implemented automatic rate adaptation techniques following the recommendations of the IEEE 802.11b standard. We compare our routing protocol against Clusterpow, one of the protocols proposed by Kawadia and Kumar. Our TPC-aware, MANKOP enabled routing protocol is a modified version of DSDV, called DSDV-KP. We chose DSDV because it is also used on Clusterpow. Results are also shown for a vanilla version of DSDV using the maximum transmission power.

The transmission powers and their respective consumption are based on a Cisco 802.11 a/b/g CardBus adapter operating on 802.11b mode [9]. One hundred nodes were randomly placed on a rectangular region, where the height is always half of the width (e.g. 2kmx1km, 4kmx2km), in order to maximise hop counts. For example, a rectangle with the dimensions of 2kmx1km will be depicted in the graphs with the value 2km^2. There were three constant bit rate 64Kbps UDP flows on the network. The sender and the receiver of each flow were placed

[2]On the AKP level, the access is based on XML-RPC `get` and `set` methods.

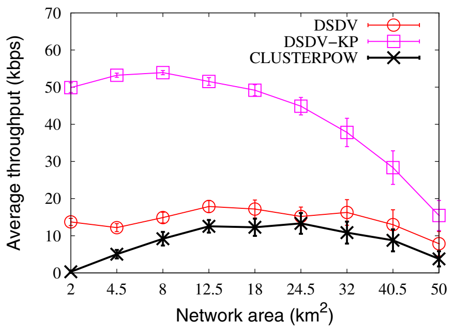

Fig. 3. Average throughput.

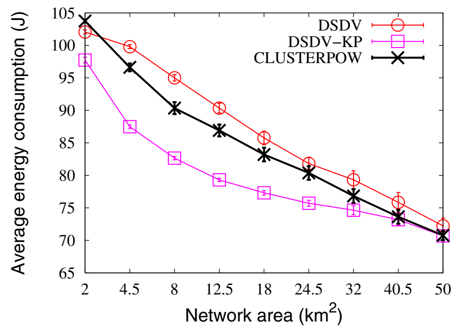

Fig. 4. Average energy consumption.

at diagonally opposite corners of the region. Nodes moved following the random way-point mobility model, with an average node speed of 2m/s. The simulations lasted for 100s, allowing all flows to reach a stationary state. Results were averaged over 60 independent simulations with a confidence interval of 99%.

Figure 3 shows the average throughput. DSDV and Clusterpow's performance resembles a parabola. This is due to the high amount of collisions on small networks, while on large networks packets are lost as no routes could be found. DSDV-KP presented the highest throughput, outperforming DSDV from 80% up to 280%. As DSDV and Clusterpow do not take the data rate into account when building their routes, both protocols employ, on average, the lowest data rate. This occurs because distant nodes, which are reachable only using a reduced data rate, are frequently present on routes. This effect occurs because route advertisement packets are broadcasted, and thus must be sent at the basic data rate. DSDV-KP, on the other hand, prefers nodes with higher data rates, thus achieving an average data rate of up to 9Mbps.

Even though Clusterpow requires up to 80% more control overhead than DSDV, the use of smaller transmission powers makes Clusterpow more energy-efficient than DSDV, as shown in Figure 4. DSDV-KP reduced even further the energy consumption, due to the fast data rate (as the radio spends less time transmitting the data packets) and the smaller overhead. Again, all protocols perform alike for sparser networks, as they build very similar routes (with approximately the same number of hops, transmission power and data rate) and the amount of collisions is insignificant.

V. Conclusions and Future Work

The *Knowledge Plane* paradigm is one of the propositions to face the challenges presented by the evolution of networks. In this work we presented a knowledge plane for ad hoc networks, called MANKOP. Each node having a MANKOP module stores information from all layers and from other nodes of the network in one centralized place, accessible to protocols and applications. Thus, protocols and applications may use algorithms that take into account information from other layers to improve their performance.

Simulations of MANKOP-aware transmission power control protocols showed that MANKOP allows a significant performance increase in terms of throughput and energy consumption. This is due to higher integration of the protocols when compared to the traditional, i.e. layered approaches.

Future work will employ the correlation of the information in MANKOP to improve the diagnosis of important network events, such as node and link failures, security attacks, among others, enabling improved auto-configuration and auto-optimization algorithms for MANETs.

Acknoweldgements

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

References

[1] D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, "A knowledge plane for the internet," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, 2003, pp. 3–10.

[2] D. F. Macedo, A. L. dos Santos, J. M. Nogueira, and G. Pujolle, "Optimizing wireless ad hoc communication resources with a knowledge plane," in *The ninth IFIP International Conference on Mobile and Wireless Communications Networks (MWCN)*, September 2007, pp. 86–90.

[3] V. Kawadia and P. R. Kumar, "Principles and protocols for power control in wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 76–88, Jan 2005.

[4] M. Conti, E. Gregori, and G. Turi, "A cross-layer optimization of gnutella for mobile ad hoc networks," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, 2005, pp. 343–354.

[5] M. Razzaque, S. Dobson, and P. Nixon, "A cross-layer architecture for autonomic communications," in *Autonomic Networking*, November 2006, pp. 25–35.

[6] R. Winter, J. H. Schiller, N. Nikaein, and C. Bonnet, "Crosstalk: Cross-layer decision support based on global knowledge," *IEEE Communications Magazine*, vol. 44, no. 1, pp. 93–99, Jan 2006.

[7] L. H. A. Correia, D. F. Macedo, A. L. dos Santos, A. A. F. Loureiro, and J. M. S. Nogueira, "Transmission power control techniques for wireless sensor networks," *Elsevier Computer Networks*, vol. 51, no. 17, pp. 4765–4779, Dec. 2007.

[8] S. H. Y. Wong, S. Lu, H. Yang, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking (MobiCom)*, 2006, pp. 146–157.

[9] "Cisco Aironet 802.11abg Cardbus Adapter," http://www.cisco.com/en/US/products/hw/wireless/ps4555/ps5818/index.html, Oct., 2007.