

# TOWARDS AUTOMATION OF MANAGEMENT AND PLANNING FOR FUTURE MILITARY TACTICAL NETWORKS

Cho-Yu Jason Chiang and Ritu Chadha  
Telcordia Technologies, Piscataway, NJ

Scott Newman and Richard Lo  
U.S. Army CERDEC, Fort Monmouth, NJ

## ABSTRACT

*Mobile Ad hoc NETWORKS (MANETs) have been adopted as the building blocks of future military tactical networks. We believe that the dynamic nature of such networks makes the automation of both network management and network planning a necessity rather than just a desirable feature. The creation of DRAMA, a policy-based network management system designed for managing dynamic MANETs, is the first step of this automation effort. Since the current practice requires that policies be specified manually, which we have found to be a challenging task, it is highly desirable to automate the generation of network management policies by network planning. In this paper, we present our vision of automating the management and planning of future tactical networks, describe what has been accomplished and ongoing activities, and discuss future work towards achieving this vision.*

## INTRODUCTION

To accommodate the agility required by the military force, future tactical networks such as the FCS [1] networks will depend heavily on the use of Mobile Ad hoc NETWORKS (MANETs) [8]. Such networks are characterized by (i) absence of any infrastructure; (ii) use of radios rather than fibers/cables, i.e. wireless communication; (iii) participation of every node in packet forwarding (i.e. every node is a mobile router); and (iv) dynamic network topologies. Past research on MANETs has been focused mostly on the enabling technologies such as radios, media access, routing, etc. With the coming of age of these networks, a grand challenge has emerged: *how do we automate both network management and network planning to assure that networks will function effectively in dynamic and possibly bandwidth-constrained environments?*

Today's commercial network management and network planning tools were mostly designed for networks with abundant bandwidth and stable topologies. These tools are ill-suited for military MANETs due to the differences in characteristics between MANETs and conventional networks. MANET management tools require self-healing and self-adapting capabilities because MANET conditions may change significantly. MANETs must limit network management traffic because the limited bandwidth

available should be used predominantly by application traffic; besides, they also need to cope with unpredictable link quality and network connectivity. To address these needs, the U.S. Army CERDEC launched the DRAMA<sup>1</sup> (Dynamic ReAddressing and Management for the Army) Science and Technologies Objective (STO) to develop a tool suitable for managing MANETs. It resulted in the creation of a network management tool [2][3][4] that is distributed, agent-based, and policy-enabled in order to provide the necessary self-healing and self-adaptive functionalities required for managing MANETs. The enforcement of policies by distributed intelligent agents allows the behavior of this management tool to autonomously adapt to dynamic network condition changes.

DRAMA technologies represent the state-of-the-art in automating MANET management, and they are currently being transitioned to the FCS network management system. By using a policy-based management tool like DRAMA, a network can autonomously adjust its behavior by implementing policies, thus greatly reducing the dependency on human administration and providing much better network resilience and reliability. Although the successful crafting of the DRAMA management tool shows that a big stride has been made in automating network management, policy-based management systems need policies as input. The state of the art is that policies are specified by network administrators. Given that military missions each have differing communications requirements and therefore require different management policies for the networks to function effectively, the following question arises: *in the network planning process, can we automate the process of generating network management policies suitable for different types of military missions having their own communications requirements?*

The complete automation of both network management and network planning based on specific mission needs is a

---

<sup>1</sup> The research reported in this document/presentation was performed in connection with contract number DAAD19-01-C-0062 with the U.S. Army Research Laboratory. The views and conclusions contained in this document are those of the authors and should not be interpreted as presenting the official policies or position, either expressed or implied, of the U.S. Army Research Laboratory, or the U.S. Government unless so designated by other authorized documents. Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

significant challenge. The main purpose of this paper is to draw the attention of the military network operations community to the importance of automating network management and network planning, and to stimulate interest in addressing the various issues that arise in this automation effort. The design and development of the DRAMA management system can be considered as the first step towards automating network operations. By using DRAMA, we have demonstrated that the distributed, policy-based management paradigm is suitable for managing MANETs as they scale to 500 nodes [7]. In addition, our experience in using DRAMA gave us an opportunity to recognize that there was another cornerstone challenge to conquer: it is necessary to automate the specifications of appropriate management policies for different types of military missions. Our position is that automating the generation of network management policies in the network planning process is a necessity rather than a desirable feature. The specification of a suitable set of policies to manage dynamic MANETs is not trivial; besides, it is very difficult to validate the correctness and consistency of a large set of policies. Therefore, in addition to automating network management with policy control, we must also automate the generation of network management policies during the network planning process by taking into account the communications needs of the target mission.

The rest of this paper is organized as follows. First we give an overview of the design of the DRAMA management tool, and explain how it automates MANET management by implementing policies in a distributed manner. Next, we describe our seedling work on automating MANET planning in terms of generating policies for different types of missions with different communications needs. Finally we discuss future work towards achieving the complete automation of network management and network planning.

## NETWORK MANAGEMENT AUTOMATION

As mentioned earlier, our work was motivated by the need for a tool for managing MANETs, which pose the following unique challenges:

- Network management functions have to be tunable in a flexible manner, per administration needs;
- Network management traffic overhead must be regulated appropriately;
- A network management system must maintain its functionality autonomously in the event of network topology changes.

These challenges prompted us to further explore policy-based computing [9], which promised the benefits of automating the network management tasks and enabling flexible configuration/reconfiguration. A policy-based

management system works as follows. It allows network operators to enter network objectives as policies into the system, and ensures automatic enforcement of these policies so that no further manual action is required of the network operators. Once such policies are defined, they are automatically enforced by the management system. These capabilities provide network operators with very powerful tools to configure and control their network, and to re-configure their network in response to ever-changing network conditions, with the highest possible level of automation. Some examples of added functionality that would be enabled by policy-based management systems include:

- Dynamically changing the role of a node to act as a server (e.g. DNS server) based on relevant capabilities such as computing power, battery power, radio signal strength, mobility pattern, etc. Capabilities that are important in a certain environment may be irrelevant in others (e.g. battery power may be important at night but not during the daytime for solar-powered nodes); such constraints are captured as policies.
- Dynamically changing the QoS configuration parameters depending on the application needs and available bandwidth in the network.

The most important benefits of using policies to manage a network is automating network management, as evidenced by the numerous industry efforts in this area dealing with diverse networking domains, e.g., configuration management, quality of service control, traffic engineering, security, etc. A policy implies a pre-determined action pattern that is repeated by an entity whenever certain system conditions appear. We use Figure 1 to show the major benefits of policy-based management.

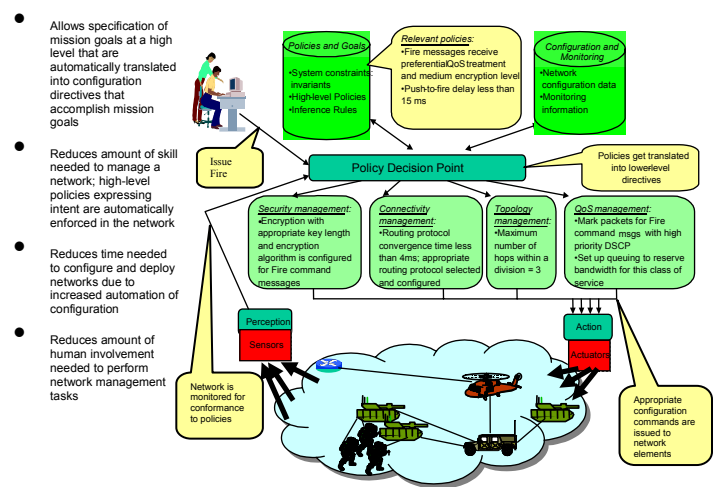


Figure 1. Benefits of policy-based management

However, before the creation of DRAMA, most of today's policy-based systems were designed to manage the Internet. These systems lack the important properties that a MANET management system must possess.

- Generic and Customizable. The management system should not be specifically designed for a particular military mission or a specific application. It needs to embed in itself a generic design allowing it to be used for network monitoring, network configuration, data collection, fault detection and resolution, distributed control and collaboration, etc. The system should not make any assumption about the underlying networking platform, which could be the Internet, a wireless mobile network, a cellular network, an ad hoc network, or even a hybrid network composed of all the above.
- Bandwidth efficient. Bandwidth is scarce resource in MANETs. Research has shown that cross-network, multi-hops traffic significantly affect the amount of available bandwidth in MANETs. Since the purpose of a network management system is to manage networks appropriately to support network applications, network management traffic overhead therefore must be localized and reduced.
- Fully distributed. The centralized management system architecture is not suitable for MANETs due to dynamicity in such networks. Besides, centralized network management systems such as those based on SNMP are known for not using bandwidth efficiently. Therefore, management system should adopt a distributed architecture to overcome the shortcomings of the centralized architecture.
- Autonomous. A MANET management system must be capable of reacting to changes in its operating environment. It must be self-healing and self-organizing so that problems can be corrected and system can be reconfigured by itself without human intervention.
- Adaptive. A MANET management system needs to adjust its behavior based on the conditions of its operating environment. The goal of the adaptation is to enhance the efficiency and effectiveness of the managed network.
- Robust and resilient. A MANET management system must consider the most stringent MANET conditions into its design to ensure that it can always perform management functions regardless network conditions such as network partition, prolonged network delay, excessive packet corruption and packet loss, and so on.

### THE DRAMA APPROACH

We have designed and developed a suite of technologies that was used as the foundation for building a policy-based

management software framework for managing MANETs. The major design features of this software framework include (i) pluggable policy-enabled software architecture with full extensibility; (ii) scalable, multi-agent framework that enables the building of robust and resilient distributed software agents for MANETs; and (iii) adaptive communications middleware moderating the limitations of the TCP/IP stack API to support distributed software over MANETs. These design features endow the DRAMA management system with all the aforementioned properties needed for managing MANETs. Below we briefly describe and discuss each of the major design features.

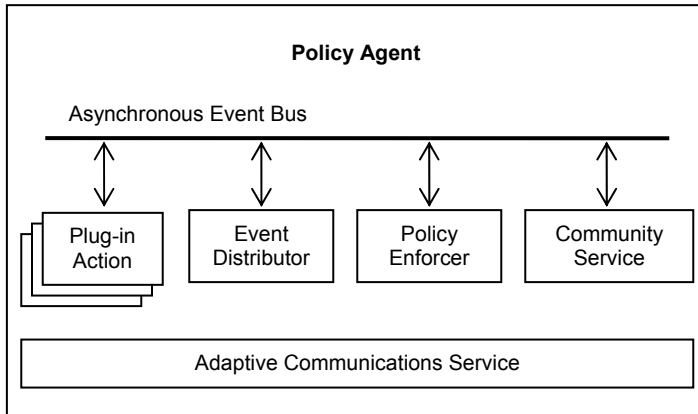
#### Pluggable Policies Architecture

Policies provide a means of specifying the desired behavior of a network at a high level. In the DRAMA system they are enforced by *policy agents*. The design of policies involves translating the specification of desired network behavior into appropriate monitoring, evaluation, filtering, aggregation, configuration and reporting actions that establish and maintain the desired network behavior. DRAMA policies include the following four components:

- **Event**: The event component of a policy specifies a list of events that will trigger the execution of this policy. Whenever one of the listed events occurs, the condition component of this policy is evaluated.
- **Condition**: A condition is a boolean expression that is evaluated by a policy agent to determine if the associated action needs to be executed.
- **Action**: An action represents a task to be executed. Actions may need additional parameters to define the actual operations to be carried out. Thus an action component includes two sub-components: action module and action parameters.
- **Scope**: The scope of a policy specifies the applicability of this policy, i.e. on which nodes it should be enforced. Only applicable policies will be enforced by a node.

Once a policy is introduced to the DRAMA policy system it could be in one of the following states—*created*, *activated*, *deactivated*, and *deleted*. A *created* policy means that the policy system has received the policy from and has stored this policy in its policy repository. A policy in created state won't take effect until it is activated. An *activated* policy is enforced by all the policy agents within the scope of the policy. Policy agents will monitor the occurrences of events on behalf of all activated policies and evaluate the conditions of policies if their events are published. If a condition evaluates to true, the associated action will be executed. If an activated policy is not needed anymore, it can be deactivated. A *deactivated* policy will be kept in the policy repositories of the system for possible reinstatement. A policy can be eventually

purged from the policy repositories and stored for historical reference purposes after it is *deleted*, which can happen when the policy will no longer be used.



**Figure 2. The policy agent framework in the DRAMA system**

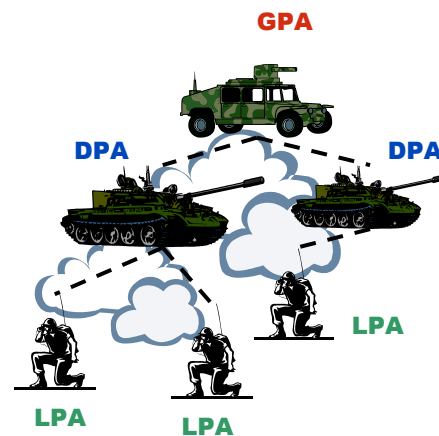
The *Policy Agent Framework (PAF)*, as shown in Figure 2, features the integration of policy control with the agent component architecture [3]. The design of this policy agent framework has been transitioned to the U.S. Army FCS program as the building foundation for its network management system. This agent framework endows its agents with the following major functionalities: policy enforcement, community service, asynchronous event transport, and adaptive communications service. In the following, we first give an overview of these components and describe how policy control is exercised within agents of this framework. We then elaborate on each of the major functionalities in detail.

The behavior of a policy agent is determined by a set of plug-in actions that are associated with policies. Plug-in actions are Java programs inheriting a set of characteristics from their base *action* class, which implements a policy control interface and provides the base implementation of this interface. The set of policies ready for enforcement by a policy agent is dynamically configurable. Therefore, the behavior of an agent can be adjusted according to the consolidation of the goals of the agents and the requirements from their network environment. Policies that are in *activated* state are ready for enforcement by the *Policy Enforcer*, a component of a policy agent, which coordinates and controls the execution of plug-in actions associated with the enforced policies. During the life cycle of a plug-in action, it may publish events based on monitoring, aggregation, or computation. The events are published to an asynchronous event bus where an event distributor will notify the appropriate subscribers that have registered to receive the events of interest to them. The primary function of the *Community Service* is to determine a *role* for an agent in the community that it joins. The community service runs in the background, maintains the

role of an agent dynamically by exchanging messages with peers in the same community, and publishes an event to the *Asynchronous Event Bus* whenever a role change event occurs. All communications by the *Plug-in Actions*, *Community Service*, and any other component of an agent use an *Adaptive Communications Service* as their only transport vehicle, which can also be controlled by policies. The purpose is to let the system communications behavior be adjusted dynamically, so that communications performance can be enhanced and the situational requirements can be observed.

### **Scalable Multi-agent Framework**

The high-level architecture of a DRAMA management system is shown in Figure 3. As shown, a collection of *Policy Agents* with different roles are used to manage a MANET. At the highest level, the *Global Policy Agent*, or *GPA*, manages multiple *Domain Policy Agents*, or *DPAs*. A DPA can manage multiple DPAs or *Local Policy Agents (LPAs)*. An LPA manages a node, and can also manage co-located devices. LPAs perform local policy-controlled configuration, monitoring, filtering, aggregation, and reporting, thus reducing management bandwidth overhead. Policies are disseminated from the GPA to DPAs, and then from DPAs to other DPAs or LPAs. Policy agents react to network status changes on various levels (globally, locally, domain-wide) by automatically reconfiguring the network as needed to deal with fault and performance issues. In this architecture, any node can dynamically take over the role of another node, which ensures system resilience. The flexible policy agent framework allows adjustments of management functionalities and behaviors through policy changes.

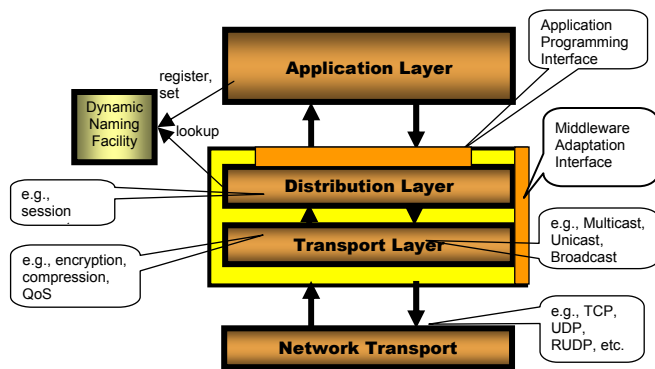


**Figure 3. The Dynamic Hierarchy of the DRAMA System**

Given that a multi-tier hierarchy is essential for scaling a distributed management system, it is critical for this hierarchy to systematically and autonomously self-form, especially when the system will be used to manage MANETs. MANET nodes may lose network connectivity

temporarily or permanently. As a result, the management hierarchy will require adjustments at times. To address this issue, the DRAMA system enables nodes to form “clusters” and then link clusters into a tree-like hierarchy. A cluster includes a single cluster leader and a set of cluster associates. Leaders of several clusters can form another cluster, and this process can be repeated to form a hierarchy comprising a cluster of recursively-formed clusters. The DRAMA system allows a hierarchy to have an arbitrary depth, since limiting the depth of the hierarchy will affect the scalability of the system; it also can adjust the hierarchy to the size of the network by expanding or shrinking the number of tiers in the hierarchy.

### Adaptive Communications Middleware



**Figure 4. Adaptive communications Middleware**

To be able to adapt communications behavior to dynamic MANET conditions, we have designed an adaptive communications middleware [2] that responds to condition changes. This allows automatic tuning of communications performance. The high-level architecture of this adaptive communications middleware is illustrated in Figure 4. This middleware consists of two layers: distribution layer and transport layer. The rationale behind having two layers in this middleware architecture is to keep the components interfacing with the DRAMA components and those interfacing with the network transport protocols completely separate. This design approach offers the following advantages:

- Since there is an interface between the distribution layer and the transport layer, it allows the middleware system to handle on-the-fly different distribution requirements with the most suitable network transport technologies. For example, if a new transport protocol can provide reliable transport similar to that provided by TCP, the middleware will use this new transport protocol instead of TCP under certain situations, as long as application communications requirements can still be satisfied.

- The separation of distribution and transport functionalities also realizes the concept of *delayed binding* [10]. This approach simplifies the application logic because the application delegates the communication identifier resolution responsibility to the middleware system. As a result, the distribution layer is responsible for resolving communication identifiers of the communication endpoints to their DNS representations, and the transport layer in turn maps the DNS representations to the network identifiers, which could be bundle identifiers in the context of Delay/Disruption Tolerant Networks, a multicast group ID if there exists underlying multicast support, unicast and broadcast IP addresses, or a mix of all the above.

### Major Outstanding Challenges

Although DRAMA has encompassed a suite of advanced technologies for automating network management, there are outstanding issues that have not been fully addressed from the perspective of automation. The major issues are listed below.

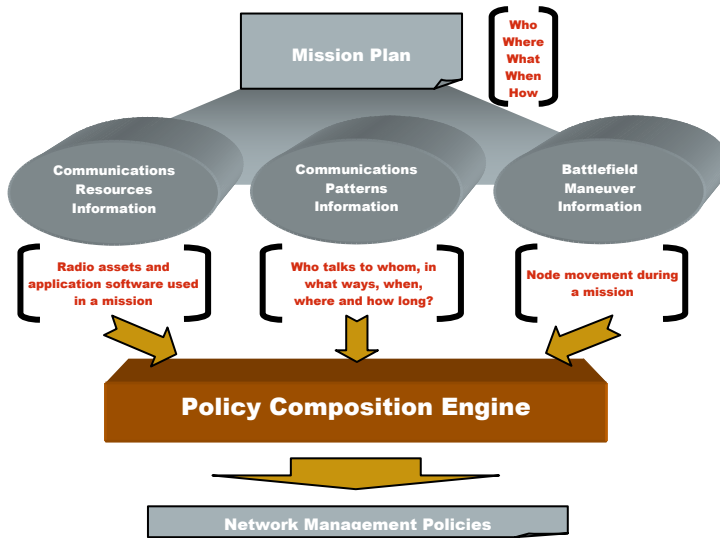
- Event distribution over MANETs. Currently DRAMA system supports the delivery of messages following its management hierarchy without any reliability assurance. More management features could be supported if a light-weight but robust event distribution mechanism is available. Existing commercial solutions have high bandwidth requirements and thus are not suitable for supporting event distribution over MANETs.
- Policy conflict detection and resolution. Policies specified by human administrators may have conflicts in them. It is crucial to detect and resolve policy conflicts before policy generation can be fully automated.

### **NETWORK PLANNING AUTOMATION**

Since policy-based management has been adopted by the FCS program as the network management paradigm, network setup and operations can be further automated if network planning tools can generate policies when given mission specifications. Automating the generation of network management policies is very difficult because the generated policies must ensure that a network will function effectively under all circumstances in highly dynamic tactical environments. With guidance and support from the FCS BCT Technologies, U.S. Army CERDEC and Telcordia are collaborating to investigate the issues involved in automating the generation of network management policies. This new endeavor, currently codenamed ‘DYMINiON’ (*Dynamic Network Managers Integrating On-the-move Networks*), aims at identifying a promising direction towards automating the generation of

network management policies. By streamlining network planning with network management, we will further automate military network operations.

The ultimate goal of the DYMINiON effort is as follows: Given an arbitrary military mission, automatically generate policies as input to a policy-based network management system such that the system can implement the policies to ensure effective functioning of the network under all circumstances. To accomplish this goal, we are taking the following steps. First, we need to precisely formulate the problem to solve. Regardless of the approach to use, it is essential to define and closely model the network for the mission, which depends on the following information: the nodes constituting the network and their configurations, the likely mobility pattern of these nodes, the physical environment that the network will operate within, and the information exchange requirements of the communications. The above concept is illustrated in Figure 5.



**Figure 5. Mission to policies generation architecture**

It is necessary to come up with appropriate data models representing the above information and define a mission communications specification language. Our view is that given a mission specified in this language, it should be possible to derive all the communications-related information from the specification to populate the data models. Second, once the mission network has been constructed, we need an approach to generating network management policies. In essence, the behavior of a network depends on its configuration settings. For example, a network supporting DiffServ QoS can be configured to use different packet scheduling algorithms (priority queueing, weighted-fair queueing, etc.) with different bandwidth allocation parameters (bucket sizes, queue weights, etc.). The performance of the network (as measured by metrics such as packet loss and transmission

delay) for a given application traffic mix can vary drastically with different configuration settings. By observation, a network servicing a given application traffic mix can be regarded as a mathematical function: it takes a network configuration setting as input, and produces a performance measurement as output. In this manner, we have formulated an optimization problem to solve. The goal of optimization is to identify a configuration setting resulting in the highest performance measurement. Once such a configuration setting is identified, it can be converted to network management policies.

The first main challenge to solving the aforementioned optimization problem is that there could be a large number of configuration parameters and each parameter could have many values to choose from, hence the search space would be enormous. In addition, the network dynamicity must also be taken into account regardless of the chosen approach. The second main challenge is measuring the performance of the network in a succinct fashion. To overcome these challenges, we have come up with an innovative approach that applies optimization heuristics in conjunction with modeling and simulation to this problem. *Simulated Annealing (SA)* was adopted as the base optimization heuristic to explore because of the reported success of applying *SA* to intrinsically intractable problems. At a high level our approach works as follows. We start by picking a configuration setting in the search space as input and find its utility (network effectiveness measure) as output. To take into account the network dynamicity, we derive the output not by evaluating a closed-form mathematical function, but by simulating communications in a dynamic network throughout a mission and then deriving a single utility value representing the aggregate performance of the communications. Then we examine the next configuration setting by following the *SA* process. The configuration setting generating the best utility value that has been obtained is recorded. The above procedure is repeated until the *SA* process determines that the result has converged. At that point we will have identified a network configuration setting producing an excellent, if not optimal, utility from the perspective of communications performance. Based on this configuration setting we can generate policies to effectively manage the network for the target mission.

Note that there are multiple dimensions of communications parameters to configure for a given mission, such as connectivity, robustness, Quality of Service (QoS), security, etc. Since QoS is a significant concern for military network operations and planning of QoS parameters is known to be difficult, automatic generation of QoS management policies has been chosen as a milestone challenge for DYMINiON. The QoS solution is based on DiffServ, following the FCS model.

We are building the above solution to show that our approach will successfully generate DiffServ configuration policies that provide effective QoS for the communications of a given mission.

### FUTURE DIRECTIONS

We regard our DYMINiON work as the first crack at a very hard problem in network planning automation. The preliminary results have shown that our approach is promising. On the other hand, we also have identified issues to address. For example, there is a tradeoff issue between the effectiveness of a plan and the computation time taken to automatically generate the plan. We therefore have redirected our attention from investigating possible alternatives at several module/algorithm decision points to reducing the computation time needed to generate a plan at the desired effectiveness level.

We also need to tie our framework with other currently available planning tools. We think there is a need to come up with an overarching architectural framework such that tools geared towards network planning purposes can be integrated seamlessly.

In addition to furthering the research on network planning, we also plan to investigate the automation of network re-planning. Network re-planning is fundamentally different from network planning—network planning plans a network from scratch while network re-planning focuses on addressing outstanding issues in the network. The goal of network planning is to take the assumed mission scenarios into account and come up with a network plan. Chances are the reality may deviate from the assumed scenarios for many reasons; in some situations, a network must be re-planned. Therefore, network re-planning is to deal with the issues that were not considered by the original plan. The modified plan should cause minimal impact to the ongoing communications and the re-planning process should be able to address issues in real time or near real-time.

### SUMMARY

In this paper, we presented our vision of automating both network management and network planning. We described previous accomplishments, ongoing research, and future directions. We envision that in the future a tool suite can be made available to streamline the automation of the process of network planning and network management. This will result in great benefits to future military network operations.

### ACKNOWLEDGEMENT

We would like to thank PM FCS BCT Technologies for their guidance and support of our DYMINiON research.

### REFERENCES

- [1] US Army, “Future Combat Systems”, <http://www.army.mil/fcs/>
- [2] C. Chiang, R. Chadha, G. Levin, S. Li, and Y-H Cheng, “AMS: An Adaptive Middleware System for Ad hoc Networks”, Proceedings of the Military Communications Conference (MILCOM 2005), Atlantic City, NJ, Oct. 17-20, 2005.
- [3] C. Chiang, R. Chadha, Y-H Cheng, S. Li, G. Levin, and A. Poylisher, “A Novel Software Agent Framework with Embedded Policy Control”, Proceedings of the Military Communications Conference (MILCOM 2005), Atlantic City, NJ, Oct. 17-20, 2005.
- [4] R. Chadha, Y-H Cheng, C. Chiang, S. Li, G. Levin, and A. Poylisher, “DRAMA: A Distributed Policy-Based Mobile Ad Hoc Network Management System”, Proceedings of the Military Communications Conference (MILCOM 2005), Atlantic City, NJ, Oct. 17-20, 2005.
- [5] R. Chadha, Y.-H. Cheng, C.-Y. J. Chiang, G. Levin, S. Li, and A. Poylisher, “DRAMA: A Distributed Policy-based Management System”, The Third International Conference on Mobile Systems, Applications, and Services, June 6-8, 2005, Seattle, WA.
- [6] R. Chadha et al., “DRAMA Performance and Scalability Analysis Report”, a deliverable to U.S. Army CERDEC, December 2005.
- [7] C. Chiang et al., “Performance analysis of drama: A distributed policy-based system for manet management”, submitted to Milcom 2006 for publication.
- [8] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris, “Capacity of Ad Hoc Wireless Networks”, MobiCom, 2001.
- [9] R. Bhatia et al., “Policy Evaluation for Network Management”, INFOCOM 2000.
- [10] J. Vetter and K. Schwan, “Techniques for delayed binding of monitoring mechanisms to application-specific instrumentation points,” Proceedings of the International Conference on Parallel Processing, 1998