# A Policy-Driven Network Management System for the Dynamic Configuration of Military Networks

Wei Koong Chai[1], Kin-Hon Ho[2], Marinos Charalambides[1], and George Pavlou[1]

[1] Networks and Services Research Lab, Department of Electronic and Electrical Engineering, University College London, Torrington Place, London, WC1E 7JE, UK
[2] Department of Computer Science, City University of Hong Kong, Kowloon Tong, Kowloon, Hong Kong
{w.chai,m.charalambides,g.pavlou}@ee.ucl.ac.uk, kinhonho@cityu.edu.hk

**Abstract.** Military networks constantly evolve to accommodate state-of-the-art technological developments across both military and commercial systems. Operating and maintaining such complex networks is no longer a trivial task. This paper presents a policy-based network management system for military networks which allows non-technical personnel (e.g. the military commander) to guide the network to behave towards specific objectives. Through policies, the system can optimize an IP-based multi-class military network with different or combinations of objectives, as requested by the decision makers. We show how the system can dynamically produce the required network configurations given specific requirements and illustrate its practicality using case studies.

**Keywords:** Policy-driven management, traffic engineering, military networks.

## 1 Introduction

In this information age, communication and information services have become vital components in security and defense agencies. Military networks are continually being enhanced with latest communication technologies to improve their flexibility, efficiency, resilience and security. Furthermore, it is highly desirable to provide service differentiation and quality of service (QoS) to ensure timely and safe delivery of mission critical information while maintaining acceptable performance to other less critical traffic. For example, during a battle, some parts of the military network may be vulnerable to attacks, reducing thus the usable available bandwidth. In this case, the network has to be re-engineered in order to sustain acceptable QoS for high-priority traffic. Failing to deliver the committed QoS may adversely affect the communications between military sites. These sophistications have made the management of the network a task requiring specific networking knowledge which decision makers (e.g. military commanders) may not possess. Although they may not have deep understanding on networking techniques, their non-technical requirements can be mapped to network-level policies, which when enforced, can achieve the high-level military objectives.

R. Sadre and A. Pras (Eds.): AIMS 2009, LNCS 5637, pp. 108–121, 2009.

An effective way to optimize the usage of network resources is to control traffic routing and subsequently support QoS. Traffic engineering (TE) is the process of specifying the manner on how traffic within a given network should be routed in order to optimize its performance [1] by balancing the load distribution or minimizing the bandwidth consumption in the network. In the context of military networks, a commander makes decisions following complex thought processes that take into account the criticality of the mission as well as current and predicted operation situations. The policies to be applied will capture the intent of the commander in terms of how the network management system should behave. Network configuration parameters are derived based on these policies and applied to the network in response to the commander's decisions.

In this paper, we concentrate on IP-based networks whereby each network link is assigned a link weight and traffic flows are routed along shortest paths to destinations. The shortest path is defined as the route between two nodes with the least total sum of link weights. Traffic routing can be controlled by setting appropriate link weights in the network by taking into account the overall traffic demand, so as to satisfy TE objectives such as improving load balancing while achieving acceptable QoS. The network is also assumed to be supporting Differentiated Services (*DiffServ*) [2] whereby several traffic classes may exist and each requires different treatment. Within this framework, critical information can be prioritized and treated differently.

In general, it is hard to find an efficient algorithm for achieving an optimal solution for the link weight setting problem. We therefore design and implement a heuristic algorithm based on the Tabu search method for solving the problem. The algorithm virtualizes the network into separate network planes so that traffic of different priorities can be forwarded with different QoS and produces a set of link weights for each virtual network plane.

In this paper, we propose a policy-based network management system that applies policy-driven TE techniques to optimize the military network such that top-level military objectives are achieved. This work has been part of a collaborative project for military networks. The paper is organized as follows: Section 2 presents the overall architecture of the proposed policy-based network management system and the inter-relationships among the components of the system. Section 3 details the implementation of the system including the specification of the link weight optimization algorithm. We evaluate our system in section 4. Case studies are presented to illustrate how our system can be used in realistic military scenarios. We provide the related work in section 5. Finally, we summarize our contributions and conclude the paper in section 6.

## 2 Decomposition of the Network Management System

### 2.1 Overall System Architecture

In this section, we present an overview of our proposed policy-driven network management system. We illustrate in Fig. 1 a decomposition of the system which is composed of two subsystems: *Policy-based Management (PBM)* and *Network Dimensioning (ND)*.

Network-level policies achieving military objectives are entered in the PBM subsystem, stored in a repository and subsequently enforced, influencing the functionality of the ND subsystem in order to address the continuously changing high-level objectives of the decision makers. Based on the requirements specified in the policies, the ND subsystem then optimizes the network by executing the optimization algorithm (detailed in section 3.1).

A Java-based Graphical User Interface (GUI) is also implemented to provide a clear and controllable representation of the outcome produced by the ND subsystem. Users can view how each traffic class is being routed across any source/destination pair and the load of each link before and after the optimization. Statistical information given in plots can be instantly generated to facilitate users in understanding the effect of the optimization. The GUI also allows users to manipulate the network topology (e.g. zooming and re-arranging the nodes) for a clearer view of the targeted network.

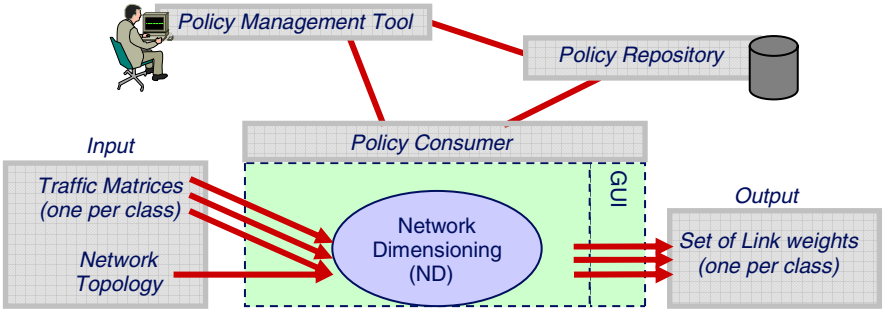


**Fig. 1.** Overview of the policy-driven network management system

## 2.2 Design of Traffic Engineering Components

The ND subsystem is developed with both existing and future TE constraints in mind. The core of the ND subsystem is the link weight optimization algorithm which incorporates the multi-topology (MT) concept for handling multi-traffic class scenario. It provides a means to configure class-based routing for different types of traffic. The physical military network is virtualized as separate network planes (or virtual topologies). Each traffic class uses a specific routing table for that virtual topology. In our algorithm, MT is used to isolate the routing of each *DiffServ* Per-Hop Behavior (PHB) by providing different link weight settings for individual PHBs. Hence, packets of different PHBs can be routed independently from one another. This advanced feature is supported by configuring multi-topology protocols such as M-OSPF [3]. Coupled with the link weight optimization algorithm, an optimal solution can thus be computed for the multi-QoS class scenario.

Fig. 2 illustrates the concept of MT-TE and serves as an example of how MT provides intra-domain path diversity across three virtual topologies between a single source/destination pair. With default link weight as 1 for all links, all traffic flows are routed via the middle path of the topology, causing congestion. However, using MT-TE, our ND subsystem can compute a set of dedicated link weights for each traffic

class. Based on individual link weight settings, traffic flows of each class may follow a different path. Congestion in the middle path is thus alleviated.

The behavior of the ND algorithm can be directly influenced by the policy directives generated by the PBM subsystem. For instance, the optimization algorithm may use a different optimization objective for achieving the desired performance represented by different policies. Besides the input from the PBM subsystem, the algorithm also requires two other inputs: the network topology (including link capacities) and the expected traffic demand (in the form of estimated traffic matrices for each traffic class).

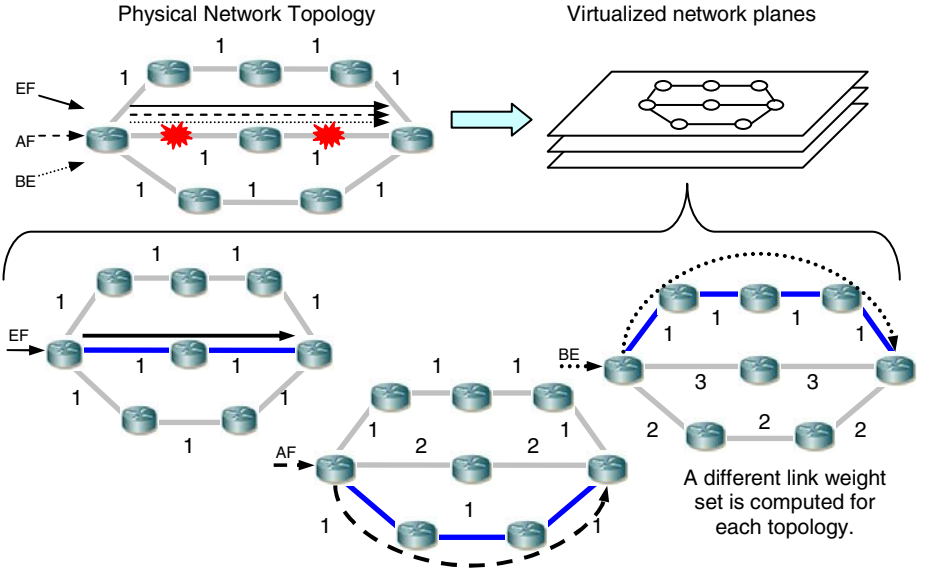


**Fig. 2.** MT-TE: The physical network virtualized as separate logical topologies with each having a different link weight setting. Congestion at the shortest path (middle route) is avoided.

### 2.3 Design of Policy Components

Policy-based management [4] provides the ability to (re-)configure networks so that desired QoS goals are achieved. The approach facilitates flexibility and adaptability as policies can be dynamically changed without modifying the underlying implementation. This is particularly useful in military scenarios where the high-level objectives of a commander can be encoded into policies and used to derive new configurations on demand. The key components of Fig. 1 are briefly described below.

- *Policy Management Tool (PMT)*: The PMT [5] provides the policy creation environment through which a network administrator can enter new policies. The latter are of the form *if* <*condition*> *then* <*action*>, where the conditional part can be a compound expression encapsulating network state and events. The <*action*> expression can be a set of actions that specify the way in which the optimization

algorithm should run to achieve the high-level objectives. The tool allows the user to store newly created policies, or view policies already stored in the repository.

- *Policy Repository (PR)*: The PR is a centralized component based on an Light-weight Directory Access Protocol (LDAP) implementation that stores policies after they have been translated into object-oriented representation. Once a new policy is stored, activation information is passed to the Policy Consumer in order to retrieve and enforce it when the relevant conditions are met.
- *Policy Consumer (PC)*: The PC [6] is the most critical component of the policy architecture and is responsible for enforcing policies on the fly while the network is operating. We integrate the functionality of both the Policy Decision Point (PDP) and the Policy Enforcement Point (PEP) of the IETF policy framework [7] into the PC. Before enforcing a policy, the consumer communicates with the PR and downloads the relevant policy objects. These are subsequently used to generate a script that implements the policy, which is interpreted into management operations when the policy is enforced. The latter can be achieved either statically through the PMT, or dynamically based on network events, and in both occasions management operations involve setting optimization attributes of the ND algorithm.

# 3 Policy-Driven IP Traffic Engineering

## 3.1 IP Traffic Engineering

### 3.1.1 Problem Formulation

We formally define the IP traffic engineering (i.e. link weight optimization) problem. A network is modeled as a directed graph $G = (V, A)$ where $V$ and $A$ represent the set of nodes and links respectively. Each link $a \in A$ has a capacity denoted by $c(a)$. We have a traffic matrix $D$ that for each pair $(s,t) \in V \times V$ represents the demand $D(s,t)$ in traffic flow between source node $s$ and destination node $t$. With each pair of $(s,t)$ and each link $a$, we associate a variable $f_a^{(s,t)}$ telling how much of the traffic flow from $s$ to $t$ goes over $a$. Variable $l(a)$ represents the total load on link $a$, i.e. the sum of the flows going over $a$. Furthermore, we denote the utilization of link $a$ by $u(a) = l(a)/c(a)$.

With the above notation, the IP TE problem can be formulated as below:

$$Minimize \quad \Phi \tag{1}$$

subject to

$$\sum_{x:(x,y)\in A} f_{(x,y)}^{(s,t)} - \sum_{x:(y,z)\in A} f_{(y,z)}^{(s,t)} = \begin{cases} -D(s,t) & \text{if } y = s, \\ D(s,t) & \text{if } y = t, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

$$f_a^{(s,t)} \geq 0 \qquad \text{a} \in \text{A; s,t} \in \text{V} \tag{3}$$

Constraints (2) are flow conservation constraints that ensure the desired traffic flow is routed from $s$ to $t$.

In our policy-based network management system, we implemented three different ND optimization objectives for policies to be mapped to.

*Optimization Objective 1 (OBJ-1): Minimize the maximum link utilization:*

$$\Phi = \underset{\forall a \in A}{\text{Max}} \ \text{u(a)} \tag{4}$$

Maximum link utilization is defined as the highest utilization among all the links in the network. Minimizing this value ensures that traffic is moved away from congested to less utilized links and is balanced over the links.

*Optimization Objective 2 (OBJ-2): Minimize the average link utilization:*

$$\Phi = \frac{\sum_{\forall a \in A} u(a)}{\mid A \mid} \tag{5}$$

The goal of this optimization objective is to minimize the overall network link utilization. It tries to minimize the total bandwidth consumption in the network by shortening the routes to be used for traffic delivery.

*Optimization Objective 3 (OBJ-3): Minimize the weighted link utilization:*

$$\Phi = w \cdot \underset{\forall a \in A}{\text{Max}} \ \text{u(a)} \ + (1 - w) \frac{\sum_{\forall a \in A} u(a)}{\mid A \mid} \tag{6}$$

This optimization objective minimizes the weighted sum of the maximum and average link utilization. It allows policy users to specify the value of $w$ in order to adjust the importance and balance of the two optimization objectives.

### 3.1.2 A Heuristic Multi-QoS Class Link Weight Optimization Algorithm

We propose a Neighborhood Search Algorithm (NSA) based on the Tabu Search technique for solving the problem efficiently with respect to performance and computation complexity. The NSA is an important tool to solve hard combinatorial optimization problems efficiently. The basic steps of NSA can be summarized as follows. Consider a starting solution $x$. NSA explores the solution space by identifying the neighborhood of $x$, $N(x)$. The neighbors of $x$ are solutions that can be obtained by applying a single local transformation (or a move) on $x$. The best solution in the neighborhood is selected as the new current solution. This neighborhood searching iterates until the stopping criterion is satisfied. The algorithm returns the best visited solution.

During neighborhood search, NSA can move the current solution to the best neighbor that either improves or worsens the quality of the solution. To avoid cycling, a special memory list is used to store previously visited solutions for a certain number of iterations. A neighbor solution is rejected if it is already in the list. To increase

effectiveness, an intensification or diversification technique is used to force the algorithm to explore parts of the solution space that have not been searched yet.

Our NSA incorporates following aspects:

*1) Neighborhood search:* the best move in the neighborhood is identified as follows:

- ***Step 1.*** Identify two sets of links – those whose utilizations are within a small percentage of the maximum link utilization (i.e. heavily utilized links) and those whose utilizations are within a small percentage of the minimum link utilization (i.e. lightly utilized links). Consider the most utilized link in the first set.
- ***Step 2.*** Increase the weight of the link by a random value in an attempt to move traffic away from that link and reduce its load. Randomly select a link from the lightly utilized link set and decrease its weight by a random value in an attempt to attract more traffic over this link from the highly utilized links.
- ***Step 3.*** Run Dijkstra's shortest path first algorithm for the current link weights to re-calculate the routes for the traffic. Then re-calculate the objective.
- ***Step 4***. Select the next most utilized link and repeat steps 2 to 5 until all the links in the heavily utilized link set have been considered.
- ***Step 5***. Among all feasible solutions, choose the one with the minimum maximum link utilization and consider it as the current solution.

*2) Tabu list*: The tabu list memorizes the most recent moves, operating as a first-in-first-out queue. Its size depends on the size and characteristics of the problem. In our problem, the tabu list consists of the links whose weights have been recently changed and the amount of increase/decrease applied to the corresponding link weight.

*3) Diversification*: The goal of diversification is to prevent the searching procedure from indefinitely exploring a region of the solution space that consists of only poor quality solutions. It is applied when there is no obvious performance improvement after a certain number of iterations. In other words, we ensure that the algorithm is not restricted to a local optimum but will explore further to find a global optimum solution. For a diversification, several links are picked up from each of the lightly and heavily utilized link sets. The weights of the selected links from the former set are decreased while the weights of the selected links from the latter set are increased. Note that any solution produced by the diversification is acceptable if it is feasible.

*4) Stopping Criterion*: the search procedure stops if either the pre-defined maximum number of iterations is reached or there is no pre-defined performance improvement for the objective function after a certain number of consecutive diversifications.

### 3.1.3 Optimization for Multiple Traffic Classes

The ND subsystem allows the optimization of multiple traffic classes. We implemented this feature in the following way. First, the traffic classes are prioritized according to their importance. In *DiffServ*, the priority of traffic classes follows the order of *EF*, *AF* and *BE*. Next, the ND algorithm is executed for the highest-priority traffic class. The outcome (i.e. the set of link weights) is assigned to that traffic class over the corresponding virtual network plane. The residual bandwidth in the network, which has not been used by any traffic class, is recorded. The ND algorithm repeats the procedures above for optimizing the next highest-priority traffic class while taking into account the residual bandwidth. It stops when all the traffic classes have been processed.

### 3.2 Policies for Optimizing Military Networks

To effectively utilize the ND algorithm and adapt the network configuration according to some objectives, we have defined a set of military policies that can influence the output of the algorithm and demonstrate its capabilities. These policies are derived from sample high-level military objectives that can be requested by a commander as follows:

*Commander Objective 1 (CO-1):*
Under normal conditions setup network to either:

a. avoid service quality degradation
b. accommodate as many services as possible

*Commander Objective 2 (CO-2):*
During rescue missions sustain quality of associated services

*Commander Objective 3 (CO-3):*
During battle readiness level accommodate as many services as possible without compromising quality.

Each of the above objectives is mapped to the policies of Table 1 which when triggered, instruct the ND subsystem to execute the algorithm with the necessary parameters and produce a new network configuration. It should be noted that we distinguish between off-line (e.g. P1) and on-line policies (e.g. P2, P3). The former are evaluated statically before the network is deployed, whereas the latter are evaluated dynamically while the network is operating.

**Table 1.** Network optimization policies

| Policy ID | Policy Condition | Policy Action |
|---|---|---|
| P1 | event(initNetwork) | optimize(*objective*) |
| P2 | event(congestion) && event(rescueOp) | optimize(maxLinkLoad) |
| P3 | event(battleReadiness) && maxLinkLoad < 90% | decrWeight(20%) |

Policy P1 is used to generate a link weight setting for the initial deployment of the network. It is an off-line policy that executes the algorithm according to the ND optimization objective, which is specified as a parameter in the policy action. Minimizing the maximum link load spreads the traffic throughout the network thus achieving CO-1a, whereas minimizing the average link load forces traffic to follow the shortest path thus achieving CO-1b.

Policy P2 is a special case of P1 but is instead triggered by the run-time event of network congestion during a rescue operation. Since priority in such missions should be given to supporting services (e.g. video feed from the rescue location), the policy will result to a congestion-resolving configuration, despite the fact that other lower-priority services may suffer longer delays.

The last policy (P3) is used to achieve CO-3 and, instead of optimizing individual ND objectives, it takes the balanced approach described in the previous section,

where a weight $w$ acts as the bias between the two objectives. This policy is triggered when the network is required to switch to battle readiness mode. The additional condition concerning the maximum link load forces the dimensioning algorithm to iteratively decrease $w$ (initially set to 1) by 20% until the resulting configuration does not overload any network link. The results of enforcing these policies are described in the section 4.3, demonstrating that their dynamic nature can provide the network with self-optimization capabilities.

## 4 System Evaluation and Analysis

To evaluate the capability of the proposed policy-based network management system, we carry out a systematic evaluation study of two scenarios. In the first scenario, we evaluate the effectiveness of our ND algorithm on synthetic network topologies and traffic matrices. This aims to ensure that the algorithm performs as expected. After that, we provide three case studies to illustrate how our policy-based network management system can be applied to practical military situations.

### 4.1 Evaluation Study Setup

As mentioned, there are two inputs to the ND algorithm: the network topology and the traffic matrix. In our evaluation study on the synthetic network, we generate the traffic matrices based on the gravity model [8]. We manipulate the level of traffic demand via the traffic intensity multiplier *k* in the model. The larger the value of *k*, the higher the traffic load to the network. The network topology is generated based on the exponential random graph model [9]. The tunable variables in this model are the network connectivity multiplier, $\alpha$ and the maximum distance between any pair of network nodes, *L*. In this model, the probability of a link connecting a pair of nodes increases linearly with $\alpha$ and decreases exponentially with the distance between them.

### 4.2 Evaluation on a Synthetic Network

We first present our evaluation results for the ND algorithm in finding the optimal link weight set for a network topology with a range of traffic load conditions so that the traffic flows are routed to avoid overloading of network links. A 30-node network is generated with $\alpha = 0.5$ and $L = 50$ unit distance. All links have the same capacity of 8 Mbps. In this section, we configure the network to support only one PHB. The default weights for all links are set to 100. We varied the traffic load by tuning the variable *k* in the Gravity model. We optimized the network based on *OBJ-1* for this evaluation (i.e. minimizing the maximum link utilization).

We observe from Fig. 3 a general increasing trend of the highest link load when *k* is increased. This is because the network load is directly proportional to *k*. In all cases, our algorithm manages to find a set of link weight that reduces the highest link load. When $k > 25$, the highest link load before optimization is over 100%, implying severe network congestion. For cases where $30 \leq k \leq 40$, the ND subsystem manages to find a link weight setting that avoid link overloading in the network (i.e. reduce highest link load to below 100%). However, it fails to find a feasible solution that avoids link overloading when $k > 40$. This is caused by the existence of a critical links which traffic flows to some destinations cannot avoid. In such cases, simply

optimizing traffic routing may not solve the congestion problem. Additional physical resources (e.g. new links or increased capacity to the critical link) are necessary in order to accommodate the traffic load.

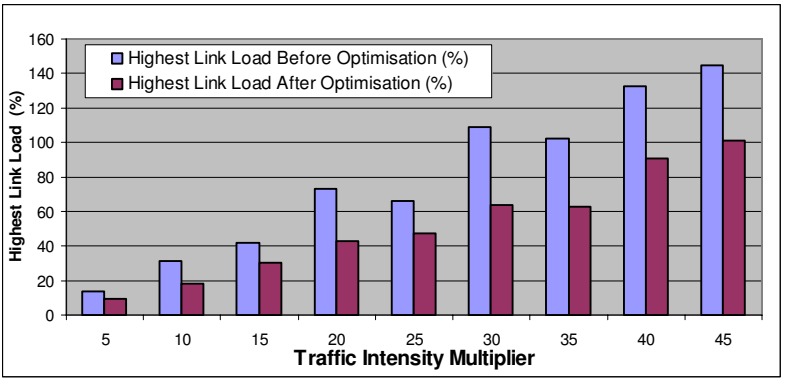


**Fig. 3.** Highest link load for different traffic load is always reduced after optimization

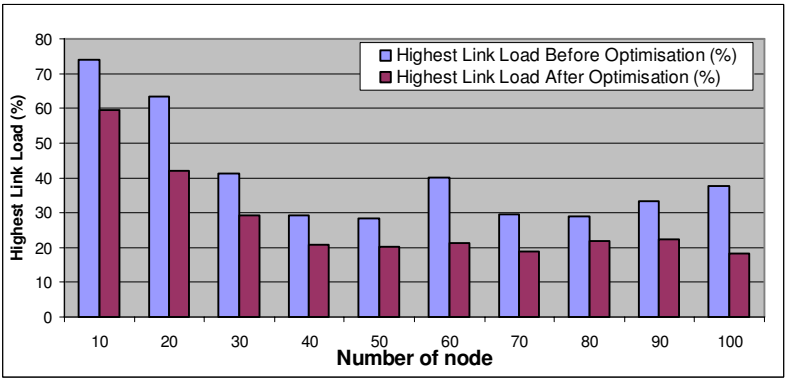


**Fig. 4.** Highest link load for different network sizes is always reduced after optimization

Next, we evaluate the effects of different network sizes with $\alpha = 1.0$, $L = 100$, link capacity = 8 Mbps and $k = 30$. Fig. 4 shows the optimization results. In all cases, the highest link load is decreased after the optimization.

### 4.3 Case Study

In section 4.2, we have shown the effectiveness of the ND subsystem in optimizing network resource utilization. Now, we illustrate how the proposed policy-driven network management system can be applied to practical military networks. For the case studies presented here, we take a sample military network topology of 14 nodes and estimated traffic matrices. For confidentiality reasons, we appropriately scale the traffic matrices and do not reveal the topology details in the paper.

### 4.3.1 Case Study 1: Initialization of the Military Network

A military network with *DiffServ* capabilities has just been deployed and a daily communication load is forecasted for each PHB. The network must be dimensioned to spread traffic across the network in a balanced manner to ensure acceptable service to all PHBs (i.e. CO-1a). We achieve this by applying policy P1 that instructs the ND subsystem to optimize the traffic routing based on *optimization objective 1*.

***if*** *event(initNetwork)* ***then*** *optimize(OBJ-1)*

We show the effect of applying this policy to the network in Fig. 5. We observe that the highest link load is reduced from 151.25% to 78.75%. Before the optimization, there are seven links that are being utilized and link 3-4 and 6-10 are overloaded. After optimization, more links are being used to carry the traffic flows (i.e. 12 links) and all the link loads are well below 100% with the highest reaching approximately 80% (i.e. link 8-9). This implies that some traffic flows are being routed via longer paths traversing some links which are otherwise unused so that no link(s) are being overloaded. In other words, the new link weight setting manages to spread the load in a more evenly manner. The downside of this is that the overall link load is increased from 22.765% to 24.948%. Nevertheless, this increment is negligible.

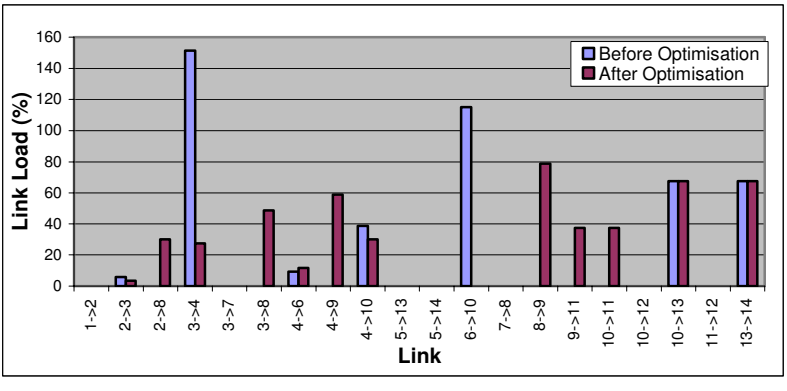


**Fig. 5.** The link load before and after optimization

### 4.3.2 Case Study 2: Rescue Mission

A military helicopter carrying an important person crashes at a politically sensitive site. The exact coordinates of the site are unknown. A rescue mission to recover the personnel is initiated. An unmanned surveillance vehicle that provides a video feed back to the headquarters is deployed to determine the crash site. Meanwhile, a link within the network fails causing the quality of the video to deteriorate significantly because of congestion. The quality of the video feed must be immediately restored. Policy P2 is applied to the network with the new information of the failed link.

***if*** *event(rescueOp) && event(congestion)* ***then*** *optimize(OBJ-1)*

We show in Fig. 6 screenshots from our system. Note that the failed link (i.e. link 4-10) has already been removed from the network topology. Before applying policy

P2, we can clearly see that link 3-4 and 6-10 are overloaded (i.e. traffic flowing through these links suffers QoS degradation). After the optimization, these links are no longer overloaded while several links carry increased traffic (e.g. link 3-8 and 10-11). Some traffic flows have been diverted to other routes to alleviate congestion.

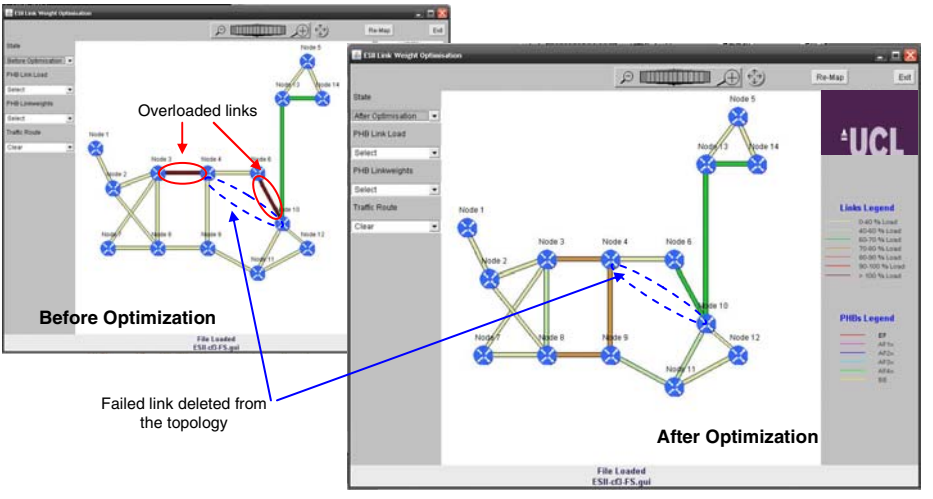


**Fig. 6.** Re-optimizing the network to sustain QoS after link failure

### 4.3.3 Case Study 3: Dynamic Configuration of the Military Network

New intelligence gathered indicates possible attacks. The network is to be set to battle readiness mode (i.e. preparing to accommodate new missions and thus having increased network load). This requires the network to have minimal average load without causing QoS degradation to in-progress services. With the weight $w$ of *optimization objective 3* initially set to 1.0, policy P3 is applied to the network as follows:

***if*** *event(battleReadiness)* ***&&*** *maxLinkLoad < 90%* ***then*** *decrWeight(20%)*

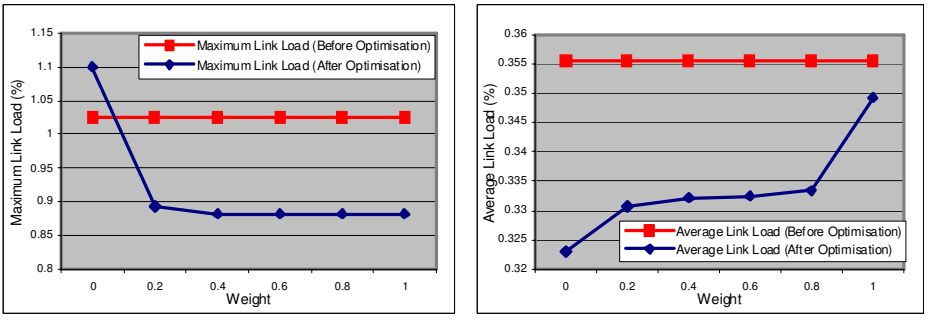


**Fig. 7.** Optimization with different weights; (left) maximum link load, (right) average link load

We present the results of the optimization via *OBJ-3* by varying the weight from 0.0 to 1.0 in Fig. 7. The plots show how the weight can tune the optimization results from one objective to another. When the weight is 0.0, we are essentially optimizing the network via *OBJ-2*. Conversely, when the weight to 1.0, we are actually using *OBJ-1* as the ND optimization objective.

In this case study, the objective is to prepare the network for new incoming mission traffic. At the same time, in-progress services are not to be disrupted by congestion. Hence, the policy runs the ND algorithm iteratively with decreasing weight until the maximum link utilization exceeds 90%. From our results, we can see that the optimal weight setting is 0.2 where the average network is decreased without having any link higher than 90% utilization.

## 5 Related Work

Military networks are by nature highly complex and dynamic. Compared to commercial networks, the response times for military networks are often much shorter and critical. Such networks also require high level of automation. As such, policies, especially dynamic ones, are much sought-after for efficient enforcement of network configurations. In fact, [10] has illustrated the potential of a policy-based management system for military networks. Previous work on policy-based network management system (e.g. [5] and [11]) mostly targeted regular civilian or commercial networks, disregarding the specific characteristics of military networks. We built upon past work and developed a full policy-based management system focusing on the requirements of military networks.

Routing management system is being used by network and service providers to effectively manage routing in their networks. The objective is to optimize the operational IP performance such as minimizing bandwidth consumption or improving load balancing. There are several such systems in literature. [12] has proposed a system that optimizes routing in pure IP networks by appropriately adjusting the link weights of the OSPF protocol. [13] proposed a similar system for IP/MPLS networks. Instead of optimizing link weights, explicit paths between each pair of nodes are determined. However, these systems required the participation of network operators to manually configure their optimization behaviours and have not taken into account system autonomy and self-management which are two features much needed in military networks.

## 6 Conclusions

In this paper, we presented the results of our work from a collaborative research project regarding military networks. A policy-based network management system with user-friendly graphical interface targeting military requirements and scenarios was described. The system is driven by policies to optimize the network for different objectives. It is also capable of dealing with conflicting optimization objectives via a weight acting as a tuning parameter. We also developed and validated a link weight optimization algorithm employing the Tabu Search method to influence the dynamics of routing traffic within the network. We incorporated the concept of multiple virtual

topologies in our algorithm for handling multi-QoS class scenarios. Our results suggest that the pure IP-based *DiffServ* network can be optimized for various TE objectives through the intelligent assignment of link weights. With the exception of heavily loaded conditions, there will generally be multiple feasible link weight solutions. We should also mention that this optimization and subsequent configuration is fairly robust as the traffic matrices in those networks can be fairly accurately predicted, unlike in operational ISP networks in which end-user behaviour can vary dramatically. Finally, we demonstrated our system via three realistic military scenarios applying policies to achieve specific high level objectives. We illustrated the application of both static and dynamic policies to the military network.

## References


1. Awduche, D., et al.: Overview and Principles of Internet Traffic Engineering, IETF RFC 3272 (May 2002)
2. Blake, S., et al.: An architecture for Differentiated Service, RFC 2475 (December 1998)
3. Psenak, P., et al.: Multi-Topology (MT) Routing in OSPF, IETF RFC 4915 (June 2007)
4. Verma, D.C.: Policy-Based Networking, pp. 155–165. New Riders Publishing (2000) ISBN 1-57870-226-7
5. Flegkas, P., et al.: Design and Implementation of a Policy-based Resource Management Architecture. In: Proc. IEEE/IFIP Integrated Management Symposium (March 2003)
6. Flegkas, P., Trimintzios, P., Pavlou, G.: A Policy-based Quality of Service Management Architecture for IP DiffServ Networks. IEEE Network Magazine Special Issue on Policy Based Networking 16(2), 50–56 (2002)
7. Yavatkar, R., Pendarakis, D., Guerin, R.: A Framework for Policy Based Admission Control, Informational RFC 2753 (January 2000)
8. Zhang, Y., et al.: Fast Accurate Computation of Large-Scale IP Traffic Matrices from Link Loads. In: Proc. ACM SIGMETRICS (2003)
9. Zegura, E., Calvert, K., Bhattacharjee, S.: How to Model an Internetwork. In: Proc. IEEE INFOCOM (April 1996)
10. Chadha, R.: Beyond the Hype: Policies for Military Network Operations. In: Proc. Int'l. Conf. on Systems and Network Communications (ICSNC) (October 2006)
11. Trimintzios, P., et al.: Service-driven Traffic Engineering for Intradomain Quality of Service Management. IEEE Networks Magazine, 29–36 (May/June 2003)
12. Fortz, B., Rexford, J., Thorup, M.: Traffic Engineering with Traditional IP Routing Protocols. IEEE Communications Magazine 40(10), 118–124 (2002)
13. Xiao, X., Hannan, A., Bailey, B., Ni, L.M.: Traffic Engineering with MPLS in the Internet. IEEE Network Magazine 14(2), 28–33 (2000)