Designing an Implementable User-Oriented Objective Function for MANETs

Karen Zita Haigh
BBN Technologies
10 Moulton Street
Cambridge, MA 02138
khaigh@bbn.com

Olu Olofinboba
Honeywell Labs
3660 Technology Drive
Minneapolis, MN 55418
olu.a.olofinboba@honeywell.com

Choon Yik Tang
School of Electrical and Computer Engineering
University of Oklahoma
202 West Boyd Street
cytang@ou.edu

Abstract

Communication networks are usually managed with network-oriented objectives such as throughput and latency. However, in military networks, these objectives are inappropriate; instead, objectives should be oriented to the mission and network stakeholder perspectives. A satisfactory objective function must exhibit the following characteristics: (i) operationally meaningful, (ii) sufficiently flexible, (iii) easy to modify, and (iv) implementable and optimizable in practice. In this paper, we present an approach to designing an implementable, operationally meaningful objective function.

1 Introduction

Communication networks are usually managed with network-oriented objectives such as throughput and latency. However, in military networks, these objectives are inappropriate; instead, objectives should be oriented to the mission and network stakeholder perspectives. A satisfactory objective function must exhibit the following characteristics: (i) operationally meaningful, (ii) sufficiently flexible, (iii) easy to modify, and (iv) implementable and optimizable in practice.

In this paper, we describe Policy-based methods for a human expert to assert statements in a light-weight way about what should be sought or avoided, while capturing the goals of multiple stakeholders. Additionally, Policies provide a rigorous mapping to a mathematical objective function, providing the basis for accurate, mission-oriented optimization. Each request for information is associated with (a) an importance weight that depends on currently pertinent policies, and (b) a set of conformance metrics that measure how well the information request is being served. We characterize the objective function in terms of these Policy-derived components, and then describe how it can be implemented and optimized in practice.

Note that any network-level performance measure is, by definition, a global notion that cannot be computed by individual nodes based solely on local information; additional communication is required. Moreover, the delay inherent in providing this feedback is often prohibitively slow for the rapid timescales of a MANET. We present our approach to providing the necessary feedback and computing the objective function so that appropriate control decisions can be made in a timely manner.

One approach to optimizing the performance of the MANET can be found elsewhere [5].

2 Policy-based Representation of User and Operational Concerns

To improve effective performance for network stakeholders, we must incorporate network operator and user perspectives *in the initial solution*. Our solution lets stakeholders specify *policy* for incorporating user goals and operator constraints, building on successful work in airline operations and military communications [3, 4]. Policy-based systems provide a flexible, rigorous mapping from multiple users' intent to importance weightings of a mathematical objective function.

Policy is a means for a human expert to intuitively assert statements, in a mathematically useful format, about what should be sought or avoided in the domain. Policy statements are heuristics with asserted, quantitative values. The concept is central to any problem requiring precise resource allocation, such as capturing a network stakeholder's intent. Policy-based control facilitates propagation of high-level enterprise goals down to the operational level; the use of a single metric allows us to capture the needs of multiple stakeholders intuitively to the single decision-maker at the point of decision. In this way, Policy-based systems also facilitate operator awareness of automation function.

This policy approach was pioneered in Intuitive Policy Specification for Optimized Flow of Asynchronous C3I Transmission in Operations (IPSO FACTO) [4]. IPSO FACTO enables commanders to specify an information allocation policy, which informs an automated communications resources management system. In particular, relevant components include Policy Representation, Information Request, Policy Conformance Metrics, Adaptive Information Control, and Multi-User Systems.

A *Policy Representation* states a user goal and its relative importance. Each user's policy is a set of statements, or '*policy elements*.' Each statement assigns an importance value to sub-regions in multi-dimensional space. Policy elements are sequenced in order of precedence, typically from most specific to least specific.

An *Information Request* r_k is of the form (w_k, s_k, d_k, c_k) where w_k is the owner (originator of information request), s_k is the source of the information, d_k is the set of destinations, and c_k is the content characterization (application type). For each information request r_k, a policy provides an *importance value* $i_k = p(w_k, s_k, d_k, c_k)$. Policies can be based on a single dimension (e.g., requests by Soldier A [Owner, w_k] have Importance $i_k = 0.5$) or a combination of dimensions (e.g., requests by Soldier A [Owner, w_k] for Situation Awareness Maps [Content, c_k] sent by Vehicle Team B [Source, s_k] to All Other Vehicle Teams [Destinations, d_k] have Importance $i_k = 0.9$).

Policy Conformance Metrics are a computational framework for evaluating solutions against expressed policy. Each request is associated with a given commander's governing policy, and is then matched against the policy element statements, with importance determined by the first policy element match. In cases where there is no match, we use a set of Quality of Service (QoS) dimensions with predefined usefulness ratings based on attributes like freshness and reliability.

Multi-User Systems give multiple users with differing interests and scopes of control authority the ability to work collaboratively to establish a single effective policy. This approach allows us to more accurately model an operational environment where there is no single commander who makes decisions about resource usage. Each commander allocates resources according to policies of those above them in the hierarchy. New requests are matched against the governing policy set.

Adaptive Information Control allocates resources, taking into account expressed policy and world state. It treats resource allocation as an optimization problem, and attempts to maximize some measure of total information delivery value. There are several ways to estimate information delivery value in the objective function, e.g., $\sum_k i_k u_k(QoS_k)$ where i is the importance assigned to request r and u is the function that determines the utility of this QoS to the owner. Our definition of the objective function is discussed in Section 3.

A critical need for users of a dynamic MANET is the ability to view and affect 'real-time' network state, which will increase the utility, stakeholder trust and acceptance of Policy-based control. The biggest challenge in developing a real-time Graphical User Interface for network stakeholders is that users can not change policies frequently in a realistic battlefield environment. Stakeholders need the ability to modify active policies without increasing their workload. The role-based policy bundles are available at runtime, with a particular bundle's instantiation being user initiated, or potentially automatically invoked by a situation assessor.

Building on our prior experience with competing stakeholders making 'real-time' policy changes [3], we allow users to invoke predefined (or default) *policy bundles* based on specific circumstances. A policy bundle is a set of related policy statements that apply in a given situation (priorities might change given different circumstances). Initial policies and policy bundles are derived from the mission plan or created offline by network stakeholders as illustrated in Figure 1.

A commander might for instance have bundles for reconnaissance, battle, or peacekeeping operations. During reconnaissance, the commander could require that video streams have higher priority than voice communication, and all communication from a particular advance vehicle team be given the highest priority. During battle, voice communication might have the highest priority. For peacekeeping missions, the highest priority activity might be to disseminate situation awareness maps about known hazards (e.g., mine locations).

Similarly, an individual soldier might have bundles for battery conservation, covert, or critical operations. During an extended mission, the soldier could want to reduce certain data processing by their radio to conserve battery power. When in covert operations, the soldier might have policies that eliminate all voice communication but allow other kind of messages from select individuals or vehicle teams. During critical operations, the priorities on voice communication might be highest. Note that simple user interfaces can be used to instantiate policy bundles, e.g. by pressing a radio button to go into battery conservation mode.

Our approach avoids many of the pitfalls resulting from ad hoc user interface design and non-policy based techniques. We make optimization definition accessible to network stakeholders, providing a single metric for quick 'real-time' comparison of options. Our policy definition techniques provide the flexibility needed to modify the objective function to address different user goals and preferences, and different operational requirements. Just as importantly, they provide a clear mapping between the needs of multiple users and the objective function, ensuring that we are optimizing the users' current needs. Our tool allows network stakeholders to easily define new policies and their relative importance to whatever level of specificity they desire. Rather than (incorrectly) guessing user intent, we give users direct control of the system, thereby ensuring the utility, stakeholder trust and acceptance of the network.

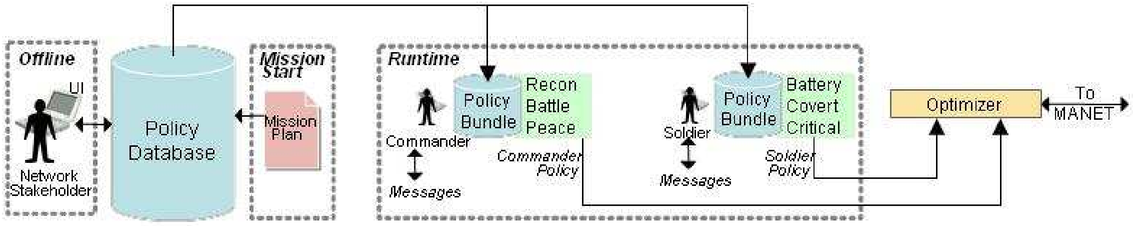

Figure 1. The Policy-based control approach enables commanders and soldiers to quickly and safely change current networking policies that drive performance goals.

3 Objective Function

Our policy-based control approach yields an operationally meaningful objective function. The next requirement is to demonstrate that it can be *implemented*. The main challenges are (1) to provide a *mathematical structure* that allows distributed nodes to collectively contribute to the global function, and (2) to *distribute the feedback* quickly and without adding overhead.

As a basis for the mathematical structure of the objective function, we have developed two scalar measures that characterize the overall MANET performance, namely, network *overhead* and *effectiveness*, treated collectively as the *objective function* to drive a network performance optimizer. These definitions of network overhead and effectiveness can be easily enhanced over time as they offer considerable flexibility to handle both planned and unplanned changes in the MANET.

Note that there are two potential optimization problem formulations:

1. Maximize Effectiveness subject to constraints on Overhead
2. Minimize Overhead subject to constraints on Effectiveness

While mathematically equivalent, these two formulations lead to different practical issues. The first formulation is most suitable when the MANET has been found to be highly wasteful and it is desirable to keep the Overhead below a smaller, known value, while providing the best possible Effectiveness. The second formulation is appropriate when there are strong indications that a mission will be successful if and only if the Effectiveness is no less than a certain value.

3.1 Overhead

Network overhead quantifies how much bandwidth the MANET *wastes* as a by-product of application-level data transfer. (Overhead is the opposite of Goodput.) Thus, the source of network overhead includes RTS/CTS/ACK packets, header of DATA packets, packets due to retransmission & forwarding, feedback for optimization purposes, etc. Moreover, multicast and network coding, in general, lead to a smaller network overhead than unicast and pure routing, respectively, since they require fewer transmissions to achieve the same amount of application-level data transfer.

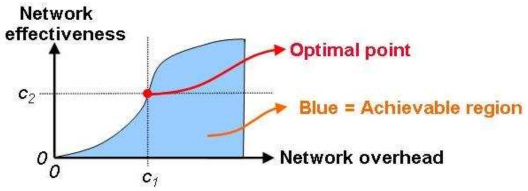

Figure 2. Effectiveness and Overhead. A goal of Overhead $\leq c_1$ and Effectiveness $\geq c_2$ determines an optimal operating point.

To mathematically capture the intended meaning of network overhead, we propose the following approach: Suppose the MANET has N nodes and suppose, during a given time interval, the PHY layer of each node i transmits m_i bits over the wireless channel, while the APP layer of each node i receives n_i bits from lower-layers. Then, the network overhead during this time interval, O, can be defined as

$$O \triangleq \frac{\sum_{i=1}^{N} m_i - \sum_{i=1}^{N} n_i}{N}, \qquad (1)$$

where the numerator represents the number of transmissions, or bandwidth, that is wasted and the denominator is a normalization factor so that the network overhead is defined on a per-node basis.

For instance, if $N = 2$ and node 1 sends a packet of 5 bits to node 2, of which 4 bits are application-level data, then the network overhead should be 1-bit-total, or 1/2-bit-per-node. Indeed, we have $m_1 = 5$, $m_2 = 0$, $n_1 = 0$, $n_2 = 4$, and $O = 1/2$ from Equation 1.

3.2 Effectiveness

Unlike network overhead, the definition of network effectiveness is more complex, since it is intended to quantify how well the MANET satisfies the following application and node requirements from mission, situational, and social standpoints. If there are K such requirements, and a set of weights w_k assigned by the Policy-based approach, then we can define Effectiveness as follows:

$$E \triangleq \frac{\sum_{k=1}^{K} w_k f_k}{\sum_{k=1}^{K} w_k} \qquad (2)$$

Each *network effectiveness component* f_k represents a measure of how well a requirement is being satisfied.

Let us examine E in more detail from the perspective of three potential requirements: (a) application-level QoS requirements of multiple heterogeneous traffic flows, (b) node-level constraints on battery lifetime, and (c) node-level constraints on LPI/LPD.

To define the network effectiveness component for (a), we use the following procedure: First, list all the T types of unicast/multicast applications, which can possibly be running on the MANET (e.g., voice, video stream, situation awareness data). Second, list all the M application-level QoS metrics, denoted as $\mathbf{q} \triangleq (q_1, q_2, \ldots, q_M)$, which can possibly be of interest (e.g., end-to-end packet loss rate, latency, jitter). Finally, for each application type $i = 1, 2, \ldots, T$, list the set of equality/inequality, soft/hard constraints on the QoS metrics $\mathbf{q}$ and define a function $f_i^a(\mathbf{q}) \in [0,1]$, which translates this set of constraints into a single number, such that $f_i^a(\mathbf{q}) = 1$ represents one extreme, where all the constraints are satisfied, and $f_i^a(\mathbf{q}) = 0$ the other, where all of them are violated or at least one hard constraint is violated. These QoS functions may be defined according to particular QoS perception profiles, or based on elementary mathematical functions. For example, if $M = 2$, q_1 is the latency, q_2 is the jitter, $q_1 < 3$ is hard, and $q_2 = 0$ is soft, then $f_i^a(q_1, q_2) = u(3 - q_1) \cdot \exp(-\alpha q_2^2)$ is a reasonable definition, where $u(\cdot)$ denotes the unit step function and $\alpha > 0$ is a parameter that characterizes how soft the jitter constraint is. Note that the above procedure is flexible in that it can easily accommodate for new application types and requirements that may emerge over time; all that is needed are the new $f_i^a(\mathbf{q})$'s.

To define the network effectiveness component for (b), we consider two cases: If node i is on battery power, let $l_i \geq 0$ denote the projected remaining battery lifetime, assuming that the battery energy will be consumed at its current rate. In addition, let $f_i^b(l_i) \in [0,1]$ be a continuous, strictly increasing function of l_i satisfying $f_i^b(0) = 0$ and $f_i^b(l_i) \to 1$ as $l_i \to \infty$, so that $f_i^b(l_i) \to 1$ implies that the projected remaining battery lifetime is very long and $f_i^b(l_i) = 0$ implies that the battery is dead. If node i is on wall power, let $f_i^b(l_i) = 1$, thus indicating that it is never concerned with energy resources.

To define the network effectiveness component for (c), we let $f_i^c \in [0,1]$ represent the inverse likelihood that RF communications among node i and its neighbors are intercepted/detected by hostile entities. Therefore, $f_i^c = 1$ implies negligible likelihood, due possibly to it being far away from enemy territory or low RF communication activities in its vicinity, and $f_i^c = 0$ implies extreme likelihood, due possibly to it being at the frontline or high RF communication activities.

The functions $f_i^a(\mathbf{q})$'s, $f_i^b(l_i)$'s, and f_i^c's defined above represent the network effectiveness components for (a), (b), and (c), respectively. Based on these components, the network effectiveness during a given time interval, E, can be defined as

$$E \triangleq \frac{\sum_{j=1}^{A} w_j^a f_{T_j}^a(\mathbf{q}) + \sum_{i=1}^{N} w_i^b f_i^b(l_i) + \sum_{i=1}^{N} w_i^c f_i^c}{\sum_{j=1}^{A} w_j^a + \sum_{i=1}^{N} w_i^b + \sum_{i=1}^{N} w_i^c} \in [0,1],$$

where A is the number of applications that are running during this time interval and $T_j \in \{1, 2, \ldots, T\}$ is the type of application j. The quantities w_j^a's, w_i^b's, and w_i^c's are non-negative weights representing the relative importance of various application-level QoS requirements and node-level constraints on battery lifetime and LPI/LPD. For instance, w_j^a is likely to be large when application j involves commanders or is time-critical for mission success; w_i^b is likely to be large when node i is strategically positioned or serving as the only routing node between two communicating subnets, thereby making its battery power more precious than others; and w_i^c is likely to be large when the discovery of node i by hostile entities would fail a mission.

To ensure that E is operationally meaningful to network stakeholders, these weights w are assigned by the policy-based control scheme, introduced in Section 2. The formulation of Equation 2 is extremely flexible in that requirements can easily be added or changed, thereby handling both planned and unplanned changes in the MANET.

4 Rapid Feedback

The fourth requirement of a satisfactory objective function is that it must be implementable and optimizable in practice. Our first challenge lies in the fact that O and E as defined in Equation 1 and Equation 2 are not readily implementable because *none* of the nodes in the MANET have sufficient knowledge to compute them. In fact, any network-level performance measure is, by definition, a global notion that cannot be computed by individual nodes based solely on local information; additional communication and feedback activities must be conducted to enable such computations.

Moreover, an additional challenge to overcome is the fact that a MANET that adapts its behavior based on feedback on O and E operates like a closed-loop control system with feedback delays. (Essentially, the nodes that can *control* the MANET are not the same nodes as those who can *observe* the effectiveness.) Feedback delays may lead to significant performance degradation, or worse, instability. Therefore, performance feedback generally needs to be as rapid as possible.

To solve this challenge, *Distributed averaging* [1, 2, 6, 7, 8] can be used to determine O and E quickly. This technique has the following property: Suppose a network has N nodes and each node i observes a number x_i. By using distributed averaging, each node i can quickly compute the average $\frac{1}{N}\sum_{i=1}^{N} x_i$, despite not knowing N or the network topology and even when x_i varies over time. Distributed averaging techniques are applicable for our MANET environment because Overhead and Effectiveness can both be cast into this form.

Although a number of distributed averaging algorithms have been proposed, they are not specifically developed for wireless networks. In this paper, we present a simple distributed averaging algorithm that exploits the fact that wireless communications are broadcast in nature, so that the rate of convergence is increased. To describe this algorithm, let $x_i(k)$ represent the number node i observes at time k and let $\hat{x}_i(k)$ represent node i's estimate of the average $\frac{1}{N}\sum_{j=1}^{N} x_j(k)$ at time k. Under the algorithm, at each time k, one of the N nodes, say, node $I(k)$ broadcast its current estimate, $\hat{x}_{I(k)}(k)$, to its one-hop neighboring nodes. Each neighboring node j, upon reception of the broadcast, computes its newest estimate $\hat{x}_j(k+1)$ according to

$$\hat{x}_j(k+1) = \alpha\frac{\hat{x}_j(k) + \hat{x}_{I(k)}(k)}{2} + (1-\alpha)x_j(k+1),$$

where $\alpha \in (0,1)$ is a weighting factor. The above equation suggests that node j's newest estimate is a weighted sum of the pairwise average of node j's current estimate and node $I(k)$'s current estimate as well as node j's newest observation. Node $I(k)$, on the other hand, computes its newest estimate $\hat{x}_{I(k)}(k+1)$ based on a first-order filter: $\hat{x}_{I(k)}(k+1) = \alpha\hat{x}_{I(k)}(k) + (1-\alpha)x_{I(k)}(k+1)$. Finally, for every node ℓ that is not a one-hop neighbor of node $I(k)$, no computation takes place, i.e., $\hat{x}_\ell(k+1) = \hat{x}_\ell(k)$. Although we have yet to perform a detailed study on the behavior of the above algorithm, it is not difficult to see why the algorithm should yield rapid convergence—every node uses all the information it receives to update its estimate; no overheard information is ever thrown away. One drawback of the algorithm, however, is that it leads to steady-state errors, the magnitudes of which depend on the value of α.

Figure 3 shows a distributed averaging result for a wireless network with 100 nodes, located on a 10-by-10 rectangular grid. The result is obtained using the above algorithm with $\alpha = 0.95$ and with each node broadcasting its latest estimate 10 times per second. Observe that although $x_i(k)$'s (the gray lines) change over time, $\hat{x}_i(k)$'s (the red lines) manage to closely track the average $\frac{1}{N}\sum_{j=1}^{N} x_j(k)$ (the black line). Also observe that a delay factor of about a second is induced. One way to decrease this delay is to either broadcast more frequently or change the value of α.

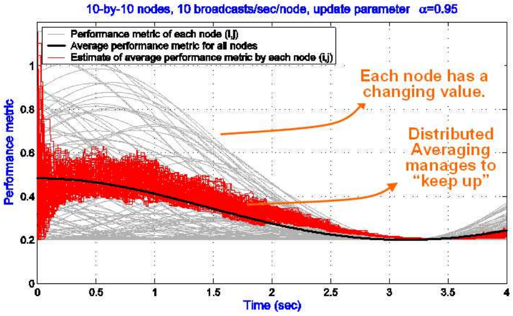

Figure 3. Distributed Averaging techniques enable nodes to rapidly share information.

Distributed averaging enables us to efficiently and rapidly distribute performance feedback among the nodes so that our objective function is both implementable and optimizable in practice.

4.1 Overhead

To cast the global network overhead O in Equation 1 into the distributed averaging form, let o_i represent the overhead induced at node i during $[t_1, t_2]$, measured in unit seconds. Let τ_i represent the amount of time the PHY layer of node i is busy transmitting during $[t_1, t_2]$. (Listening need not be penalized.) Let b_i represent the Goodput at node i. Let T_b denote the bit duration. Then the individual node overhead $o_i = \tau_i - b_i T_b$, and the global overhead is correctly formulated as $O = \frac{1}{N}\sum_{i=1}^{N} o_i$.

Figure 4 demonstrates that o_1 and o_2 can be summed to achieve the same result as in Equation 1. Figures 5, 6, and 7 show similar results for Multicast, Retransmissions, and Forwarding respectively.

4.2 Effectiveness

Effectiveness E from Equation 2 can also be cast into a form suitable for distributed averaging: $(\frac{1}{N}\sum_{i=1}^{N} e_i)/(\frac{1}{N}\sum_{i=1}^{N} a_i)$, where e_i, and a_i are known to each node i.

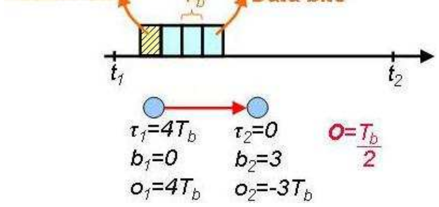

Figure 4. Unicast Overhead. From Equation 1 we expect $O = 1/2$; by decomposing O into o_1 and o_2, we get the same result.

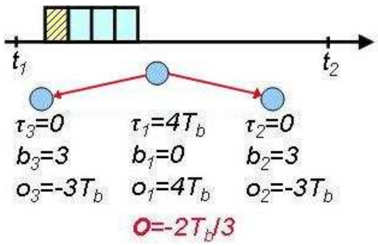

Figure 5. Multicast is rewarded (negative) because multiple nodes measure a benefit, hence outweighing the overhead.

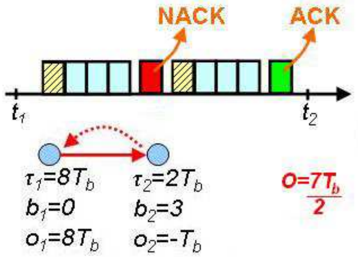

Figure 6. Retransmission is costly because only one transmission contributes to Goodput while repeat transmissions contribute only to overhead.

We have K applications, each with an associated measure f_k of how well its requirements are being met. f_k can be an arbitrarily complicated expression, provided that it is *decomposable*, i.e. each node can independently contribute to f_k without requiring information from other nodes. Intuitively, decomposability means that each node who benefits from a transmission can calculate its worth. To the best of our knowledge, decomposability holds for most application types.

Decomposability can be expressed mathematically as $f_k \triangleq \sum_{i=1}^{N} f_{ik}$, where each f_{ik} indicates the value that node i contributes to f_k; if node i does not observe or otherwise contribute to f_k, then $f_{ik} = 0$. Intuitively, if node i is the only receiver for the transmissions of a particular application, then node i is the only node that can calculate the effectiveness of those transmissions for that application. If multiple nodes benefit from the transmissions, then each contributes to f_k.

Define also A_{ik} as $A_{ik} = 1$ if node i is the source for application k, and $A_{ik} = 0$ otherwise.

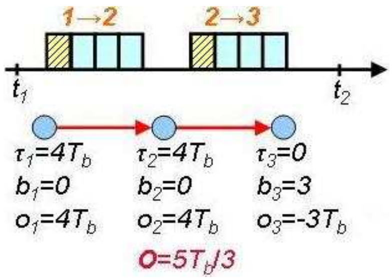

Figure 7. Forwarding is also costly.

Rewriting Equation 2, we get

$$E = \frac{\sum_{k=1}^{K} w_k (\sum_{i=1}^{N} f_{ik})}{\sum_{k=1}^{K} w_k (\sum_{i=1}^{N} A_{ik})} = \frac{\frac{1}{N}\left(\sum_{i=1}^{N} (\sum_{k=1}^{K} w_i f_{ik})\right)}{\frac{1}{N}\left(\sum_{i=1}^{N} (\sum_{k=1}^{K} w_k A_{ik})\right)}$$

Each node i can directly observe $e_i \triangleq \sum_{k=1}^{K} w_i f_{ik}$ and $a_i \triangleq \sum_{k=1}^{K} w_k A_{ik}$, demonstrating that E from Equation 2 can be cast into a form suitable for distributed averaging. Note that e_i and a_i both need to be shared to calculate the network-wide average of E.

5 Conclusions

In this paper, we have described an approach to designing and capturing a user-oriented objective function that can be optimized in practice. We describe Policy-based methods that capture the goals of multiple stakeholders in a mathematically meaningful way. We describe how to construct an objective function that captures the tradeoff between overhead and effectiveness, and can be implemented with rapid feedback for a performance optimizer. One approach to optimizing global network performance can be found elsewhere [5].

6 References

[1] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Proc. IEEE Conference on Decision and Control and European Control Conference*, Seville, Spain, December 2005.

[2] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Gossip algorithms: Design, analysis and applications. In *Proc. IEEE INFOCOM*, pages 1653–1664, Miami, FL, 2005.

[3] M. Dorneich, S. Whitlow, C. Miller, and J. Allen. A superior tool for airline operations. *Ergonomics in Design*, 12(2), 2004.

[4] H. Funk, C. Miller, J. Richardson, C. Johnson, and J. Shackleton. Applying intent-sensitive policy to automated resource allocation: command, communication and most importantly, control. In *Proceedings of the 5th International Conference on Human Interaction with Complex Systems*, 30 April - 2 May 2000.

[5] K. Z. Haigh, S. Varadarajan, and C. Y. Tang. Automatic learning-based manet cross-layer parameter configuration. In *Workshop on Wireless Ad hoc and Sensor Networks (WWASN2006)*, Lisbon, Portugal, 2006. To appear.

[6] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proc. IEEE Symposium on Foundations of Computer Science*, page 482, Washington, DC, 2003.

[7] C. C. Moallemi and B. V. Roy. Consensus propagation. In *Proc. Neural Information Processing Systems*, Vancouver, Canada, December 2005.

[8] J. N. Tsitsiklis. *Problems in Decentralized Decision-Making and Computation*. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, 1984.

Acknowledgments: This work was conducted at the Honeywell Labs under IR&D funding.