Maximizing Availability of Content in Disruptive Environments by Cross-Layer Optimization

Minyoung Kim
Computer Science Laboratory
SRI International
Menlo Park, CA 94025, USA
mkim@csl.sri.com

Je-Min Kim
Computer Systems Laboratory
Sungkyunkwan University
Suwon 440-746, South Korea
jmkim@csl.skku.edu

Mark-Oliver Stehr
Computer Science Laboratory
SRI International
Menlo Park, CA 94025, USA
stehr@csl.sri.com

Ashish Gehani
Computer Science Laboratory
SRI International
Menlo Park, CA 94025, USA
gehani@csl.sri.com

Dawood Tariq
Computer Science Laboratory
SRI International
Menlo Park, CA 94025, USA
dawood.tariq@sri.com

Jin-soo Kim
Computer Systems Laboratory
Sungkyunkwan University
Suwon 440-746, South Korea
jinsoo@csl.skku.edu

ABSTRACT

Emerging applications such as search-and-rescue operations, CNS (communication, navigation, surveillance), smart spaces, vehicular networks, mission-critical infrastructure, and disaster control require reliable content distribution under harsh network conditions and all kinds of component failures. In such scenarios, potentially heterogeneous networked components — where the networks lack reliable connections — need to be managed to improve scalability, performance, and availability of the overall system. Inspired by delay- and disruption-tolerant networking, this paper presents a distributed cross-layer monitoring and optimization method for secure content delivery as a first step toward decentralized content-based mobile ad hoc networking. In particular, we address the availability maximization problem by embedding monitoring and optimization within an existing content-distribution framework. The implications of policies at security, caching, and hardware layers that control in-network storage and hop-by-hop dissemination of content then are analyzed to maximize the content availability in disruptive environments. Additional benefits can be obtained by optimizing the control based on continuously observing the response to anomalies caused by cyber-attacks. For example, if excessive (potentially fraudulent) content is injected, the content distribution system can adapt without significantly compromising the availability.

Categories and Subject Descriptors

C.2.3 [**Computer-communication Networks**]: Network Operations—*Network Management*

General Terms

Design, Performance, Reliability, Security

SAC'13 March 18–22, 2013, Coimbra, Portugal

Keywords

Content-based networking, distributed cross-layer monitoring and optimization, MANETs

1. INTRODUCTION

Content dissemination and service delivery in a disruptive environment requires adaptive decentralized control to avoid slowing down the network and making critical services unavailable when they are needed. Information access within a MANET (Mobile Ad-hoc Network) for search-and-rescue operations by first responders who carry mobile devices can be an operational demonstration scenario of content-based mobile ad hoc networking where short latency and high availability are crucial for a successful mission. In this situation, wireless networks have severe bandwidth constraints, unreliable point-to-point communications, and very limited backhaul capability. Energy-constrained devices require efficient utilization of resources. Under such circumstances, the naive flooding or gossiping of content delivery limits scalability due to its high overhead. Strategies for active management of content and resources require a decentralized content-based networking solution.

Content inserted into the network is stored and forwarded by cooperating nodes. Metadata and queries are also inserted to represent essential attributes of content and to retrieve appropriate content from the network. Routing and caching perform in-network matching between metadata and queries. Content and metadata/queries must be protected by a decentralized security framework to enable access control of content. Optimization of the content management strategy under constraints can be seen like many other problems in networking as a utility maximization problem. Generally, optimizations at each layer require situation- and resource-aware cross-layer adaptation that is cognizant of features, limitations, and dynamicity at each layer to maintain content accessibility with reasonable tradeoffs between availability and bandwidth/energy efficiency. For instance, the degree of redundancy for caching of content in a cluster of nodes should take into account the cluster density and stability (lower layer), and at the same time the type and importance of the content (higher layer).

We propose a lightweight monitoring and optimization

framework to maximize the availability of content in a cross-layer manner. We extend an existing content-based networking framework, Haggle [18], which is a suitable basis for rapid prototyping and testing. Distributed monitoring is a core service to enable caching algorithms to make informed context-aware decisions about the state of the network without requiring a global view. Cross-layer optimization uses a notion of fitness or utility maintained for all nodes in a distributed fashion. Reactive and proactive caching policies should be guided by a measure of their utility to the nodes, which among other factors depends on the spatial distribution of content over different storage sites.

A typical content dissemination framework consists of application, security, routing/caching, and hardware layers. For instance, the application aims to share content (e.g., pictures) that matches interests (e.g., location) of users and nodes. In the security layer, parameterized control [6] can provide a tradeoff space for information access control while varying its characteristics (e.g., reliability of access grant and revoke) at the granularity of individual content. In the routing/caching layer, utility-based dissemination [24, 2, 19] can move content closer to its destination by employing a resource allocation approach to determine the optimal content to replicate or discard. Resources need to be managed to prolong the lifetime of devices and to maximize the delivery of useful content within the given budget. In [10], we presented preliminary results on a cross-layer security mechanism in the context of group communication where eavesdropping risk and data confidentiality were traded against energy consumption. In this paper, we extend the cross-layer solution to be part of a distributed content-based networking architecture and focus our effort on the optimization of availability.

We present ideas for distributed monitoring and cross-layer optimization with dynamic adaptation in a layerless architecture.[1] We embedded a monitoring/optimization manager into the Haggle framework, since it allows us to incorporate existing algorithms, instead of mandating a specific algorithm for each component. Parameterized security, utility-based replication, and energy policy are implemented as part of the security, forwarding, and resource managers, respectively. Our monitoring and optimization algorithm collects and aggregates security, performance, and resource-related information and improves network performance and content availability by localized but coordinated distributed cross-layer control. We explore different policies and evaluate their compositions on the CORE network simulator [1] with individual nodes represented by lightweight Linux containers. The results indicate that lightweight monitoring and cross-layer optimization can improve content availability in disruptive environments.

2. CONTENT AVAILABILITY IN DISRUPTIVE ENVIRONMENTS

In disruptive environments, the probability of a source being able to establish a simultaneous end-to-end path to a destination is very low. The use of opportunistic and possibly delayed contacts between the source and the destination enables multihop communication with intermediate nodes acting as forwarders. The forwarding needs to be delay- and disruption-tolerant in the sense that the intermediate nodes store the messages (i.e., content) until forwarding opportunities will occur. This "store, carry, and forward" paradigm in opportunistic networks does not necessarily require any a priori knowledge on the network topology or link status at a global scale. Routing protocols and caching strategies now are very closely intertwined, because routes must be discovered hop by hop as each piece of content is being delivered toward its destination while the intermediate nodes evaluate the utility of local caching and forwarding decisions.

To avoid causing congestion for other critical communications, content is cached only opportunistically rather than being flooded. However, there are situations or types of content for which proactive replication is more efficient and can increase the availability of content. A replication algorithm should be not ad hoc, but inherently content-based, and intuitive concepts such as utility of replication need to be expressed for efficient and timely content delivery. The utility should also take into account the probability of content delivery to the destination (or a set of destinations). Security and resource provision are also important facets of content availability. To make best use of resources, content needs to be protected by adaptive decentralized encryption with parameterized control for the key sharing and revocation. Resources such as energy, bandwidth, and storage for caching need to be traded off against each other to optimize their usage.

2.1 Motivating Scenario

In a search-and-rescue operation under challenging situations, participating nodes carried by the search-and-rescue team are typically resource-constrained and often highly mobile (e.g., VANETs). Team members are interested in particular types of content such as maps of the area and photos of the person to be rescued. Content is dynamically inserted to the network and tagged with appropriate metadata such as timestamp and location. The success of a mission depends on reliable and efficient content delivery to the interested nodes in a secure and timely manner with no assumption on stable connectivities and resources. Consider situations where mobile nodes may pretend to be valid entities and, therefore, fraudulent information can be injected to severely compromise the *availability* of content. Nodes generating unnecessarily high volumes of content (due to all kinds of reasons such as transient failures) impose excess load on the system, which can slow down the information flow.

This threat model requires dynamic configuration of individual (seemingly independent) techniques to compose appropriate protections against attack situations while also making optimal use of resources. A content-sharing system must have mechanisms for resource management that optimize collective utility gained from the sharing system such as content availability. For example, it is not recommended to use nodes that suffer from resource depletion (e.g., low battery, memory, bandwidth) to cache contents for future forwarding. Strategies for securing content rather than hosts are parameterized to dynamically control the access of content.

2.2 Rapid Prototyping in Haggle Framework

Haggle [18] leverages the idea of *in-network resolution* to offer content dissemination by matching content to interest.

[1]Haggle is layerless architecture with its event-based paradigm. The layers in this paper are logical concepts rather than the implementation perspective.

Both metadata and interest are uniformly represented as attribute/value pairs. With weighted attributes and ranking, Haggle can limit the scope of local matching and prioritize results according to relative relevance. Applications register their interest for a certain content and also add content to the network with metadata. The core system is an event-based framework consisting of a single event queue to coordinate the interactions among a set of managers with the Haggle *kernel*. *Managers* in Haggle are responsible for specific tasks such as managing communication interfaces, encapsulating a set of protocols, signing/verifying content, and sending content, etc. Managers use *modules* to implement specific algorithms such as different forwarding algorithms or protocols. The Haggle kernel and managers access a central repository, the *Datastore*, for content and information about nodes and their interfaces.

For our experiments, we modified three managers:

1. Resource manager — Nodes need to adjust their behavior when resources are scarce. The resource manager issues resource policies, based on measurements of residual battery, disk space, bandwidth, and so on. Under resource constraints, this may include limiting the scope of dissemination, managing the level of security, and controlling power consumption in transmitting data. For distributed monitoring and optimization, the resource manager may also disseminate its policy to neighbors. Cross-layer optimization within each node decides local resource policies based on distributed monitoring and guides corresponding managers. The details of monitoring and optimization are explained in Section 4.3.
2. Forwarding manager — When nodes are connected to others, the forwarding manager is determined how content is disseminated to its neighbors. A forwarding module that is plugged into the forwarding manager implements a specific forwarding algorithm. An important aspect of the forwarding architecture in Haggle are delegates, i.e., nodes that can relay and carry content in which they have no interest. We extended the existing forwarding manager to work with a new module for proactive replication of the content. The forwarding manager tunes the dissemination to fit the resource policy (e.g., degree of replication).
3. Security manager — Providing primitives for signing and verifying content, authenticating neighbors, encrypting/decrypting the content, and performing integrity checks on content would be responsibilities of a security manager in Haggle. To make current practice decentralized and content-specific, access control with multiple consumers of content needs to be a policy-based operation. Dynamic policy on access request and revocation needs to be controlled by parameter selection to allow applications to trade off between security characteristics and performance. We added a simple analytic model in Section 3.1 to the original security manager of Haggle to measure the impact of different access controls.

3. UTILITY-BASED OPTIMIZATION

With heterogeneous nodes that are severely limited in terms of their capabilities and resources in disruptive environments, secure delivery of the content requires careful investigation of the utility of both content and nodes. To maximize the availability of content, content with higher content utility may have to be prioritized by using relays with higher node utility. The content utility should be defined by the nature/amount of interest and the degree of matching. Access control for the content also plays a role in content utility, as we explain in Section 3.1. In subsequent subsections, our definition of the node utility will be introduced.

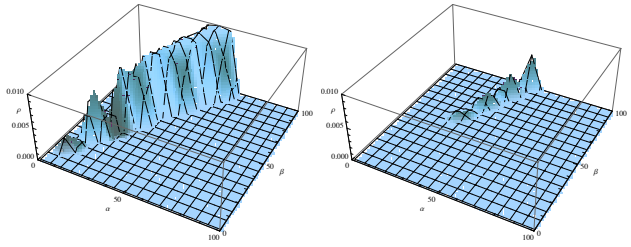

Figure 1: Reliability of both request and revocation operations as α and β are varied from 0 to 100 each, with $\rho_c = 0.9$ and (a) $\mu = 0.3$, (b) $\mu = 0.4$.

3.1 Security Layer — Parameterized Access Control

We use parameterized access control (PAC) operations by splitting each permission into β fragments, of which α are required for access to an object proposed in [6]. As long as $(\beta - \alpha + 1)$ can be removed, revocation succeeds. Reducing α or increasing β improves the rate of finding sufficient fragments for an access request to complete. Decreasing $(\beta - \alpha)$ increases the probability of successfully revoking a right. By tuning α and β, the efficiency of granting, revoking, and requesting a right can be traded. Nodes may be unreachable because of a network partition, simply be powered off or disconnected, or actively refusing to cooperate. We model the collected behavior by assuming that each node operates correctly with probability $1 - \mu$. Selecting α and β so that the ratio $\frac{\alpha}{\beta}$ exceeds μ ensures that access control operations are effected with high reliability [6].

We define the utility of PAC in terms of the benefit of the reliability of the request and revocation operations, and the cost of the computational, storage, and networking resources used by the PAC subsystem. Once we set parameters α and β, the reliability with which an access control request will complete can be modeled with $\rho_{request}(\alpha, \beta) = \sum_{i=0}^{\beta-\alpha} \binom{\beta}{i} (1-\mu)^{\beta-i} \mu^i$. Similarly, the reliability of revocation can be modeled with $\rho_{revoke}(\alpha, \beta) = \sum_{i=0}^{\alpha-1} \binom{\beta}{i} (1-\mu)^{\beta-i} \mu^i$. Therefore, we examined the combined benefit of

$$(\rho_{request} - \rho_c) \times (\rho_{revoke} - \rho_c)$$

with reliability constraint ρ_c, as shown in Figure 1.

A capability can be split into β fragments, of which α are needed to reconstruct it, using Shamir's *secret sharing* scheme [21] or a derivative. Such schemes are implemented by evaluating an $(\alpha - 1)$-degree polynomial at β points, which requires computing a total of $(\alpha - 1)\beta$ terms. However, the latency introduced by this operation is negligible in comparison to the storage and networking costs. Since copies of β fragments must be retained at nodes in the system, the storage cost is proportional to β. The networking cost comprises components for transmitting β fragments to

remote nodes during a grant operation, retrieving fragments from at least α remote nodes during a permission request, and contacting at least $(\beta - \alpha)$ remote nodes during a revocation operation.

If the insertion and retrieval of each fragment was carried out serially, the complexity would be lower bounded below by α for fragment retrieval, and by $(\beta - \alpha)$ for effecting a revocation. In practice, all three networking operations have complexity β since all fragments are inserted or requested in parallel to minimize latency. From these observations, we simplify the cost factor to be proportional to β. The utility consists of the reliability benefit normalized by this simplified operational cost (i.e., β).

3.2 Routing/Caching Layer — Replication with Opportunistic Forwarding

In [24], the authors investigated the notion of *fitness* or *utility* in opportunistic networking. Rather than storing and forwarding a copy of content to the first nodes encountered, they maintain utility function at all nodes in a distributed fashion and replicate portions of available copies according to utility. In particular, the authors of [24] control parameter L, the number of copies to be replicated with an epidemic forwarding mechanism. A node with L replicas may forward its copy L more times. Discovering the *better* relays is based on the utility function. Utility can be defined in several forms: destination-dependent (e.g., last-seen-first) or independent (e.g., most-mobile-first, most-social-first).

We combine the utility-based replication approach with Haggle's delegation forwarding that allows replication since the delegators may not be interested in the content. The Prophet [16] forwarding module implements a probabilistic routing protocol to delegate forwarding based on the statistics of node encounters and transitivity. The so-called predictability in the Prophet routing metric tries to approximate the idea that the relay will more likely deliver a message to the destination. Therefore, more replicas are appropriate, and the definition of utility needs to incorporate this aspect.

Here we simply define the replication utility u^r of node i for destination j as

$$u^r(i,j) = p(i,j)$$

where the probabilistic metric $p(i,j)$ is computed as delivery predictability from every node i for each known destination j in Prophet [16]. For our experiments, we use the following sample utility-based replication policy. If a node i_1 carrying content for a destination j encounters node i_2, it forwards $\frac{u^r(i_2,j)}{u^r(i_1,j)+u^r(i_2,j)}$ of its copies of that content to node i_2 according to the delivery predictability to node j.

3.3 Hardware Layer — Resource Provision

Transmission power control can be an effective way to minimize the eavesdropping risk in an ad hoc wireless network as proposed in [8], where the w-th eavesdropping risk is defined as the maximum probability of packets being eavesdropped with w adversarial nodes. Using a simple model, the first order eavesdropping risk is bounded below by $\frac{1}{3}r$, where r is the normalized transmission radius with nodes in an arbitrary random network. We define the destination-independent utility of node i as

$$u^p(i) = \frac{e(i)}{r(i)}$$

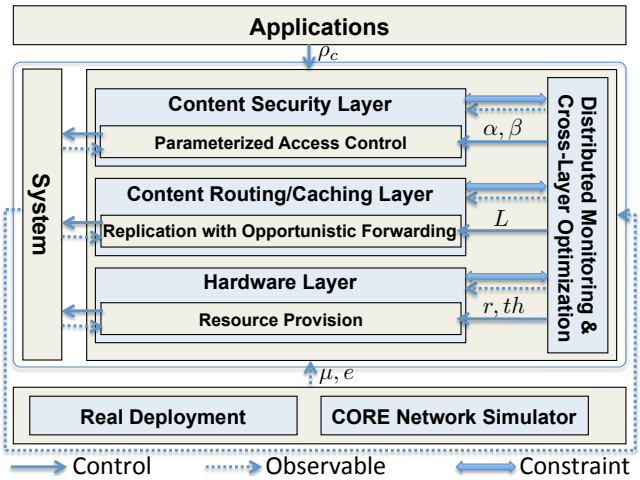

Figure 2: Distributed monitoring and cross-layer optimization framework.

where $e(i)$ and $r(i)$ represent residual energy and eavesdropping risk of node i, respectively. When node i_1 encounters node i_2, it avoids using i_2 as a delegator node if $u^p(i_1) \times th > u^p(i_2)$, where th indicates the threshold that controls the load balancing based on energy resources.

4. CROSS-LAYER OPTIMIZATION

Figure 2 illustrates our system architecture for distributed monitoring and cross-layer optimization. We adapt individual layers' utility-based optimization techniques as in Section 3. We aim to maximize the content *availability* by optimizing an objective function based on observables such as latency, energy, reliability, and user-defined *soft* and *hard* requirements.[2] The situation that the system behavior resides within the soft requirements is most desirable. When the system is observed in between soft and hard requirements, however, the optimizer needs to tune parameters such as α, β. A hard requirement indicates an upper limit, above which a user cannot tolerate the quality degradation. We define the cost function of the observables $X = (x_l, x_e, x_b, x_r)$ as

$$cost(x) = \begin{cases} \infty & \text{if } x \geq h \\ \frac{1}{h-x} - \frac{1}{h-s} & \text{if } h > x \geq s \\ 0 & \text{otherwise} \end{cases}$$

where h and s represent user-defined *hard* and *soft* requirements, respectively. The observables x_l, x_e, x_b, x_r concern latency, energy consumption, bandwidth, and (un-)reliability in PAC, respectively. The objective function is defined as

$$obj(X) = \frac{1}{\sum_i (weight_i \times cost(x_i))}$$

4.1 Optimization Procedures

Given the objective function above, we adapt a simulated annealing (SA) approach with a neighborhood operation [17] as optimization procedure. In [17], the authors implemented the concept of *exploitation* and *exploration* in SA for continuous parameter optimization from the observation that

[2]We are not deriving content *availability* from u^s, u^r, u^p. Our focus is to use existing techniques at different layers rather to devise a new utility-based optimization. More generic approaches that can take utility as an optimization criterion in a uniform way will be future work.

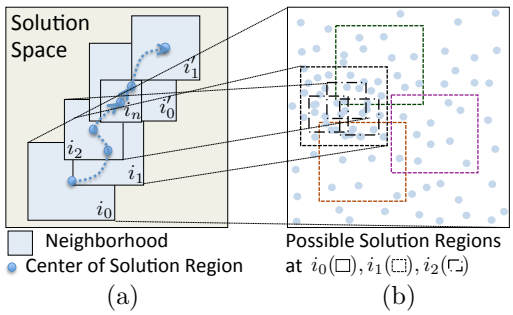

Figure 3: Optimization procedures: (a) exploitation vs. exploration with neighborhood, (b) solution region refinement by sampling.

with a vast search space it is impractical to choose direct neighbors of the current solution as new candidate solutions. Instead, new candidate solutions can be chosen from some distance of the current solution (i.e., neighborhood). There exists a tradeoff between exploitation (related to accuracy) and exploration (related to completeness) in selecting an appropriate size of the neighborhood. If the neighborhood is too small, SA presents good exploitation capabilities but the probability of reaching the global optimum is reduced. If the neighborhood is too large, SA has good exploration capabilities to find the region of the search space containing the global optimum but is less likely to exploit the optimum within the given region.

Figure 3(a) shows a sample execution of SA based on an empirically designed scaling function defining a neighborhood reduction phase $i_0 \rightarrow i_1 \rightarrow i_2 \rightarrow ... \rightarrow i_n$. Our online adaptive SA repeats the process of exploration and exploitation to balance between accuracy and completeness. For continuous optimization, sometimes the neighborhood needs to be expanded between phases (e.g., $i_n \rightarrow i_0'$) to explore further possibilities or to take into account significant changes in operational conditions (e.g., perturbation in environment or anomaly has been detected). However, we have not implemented neighborhood expansion in this study. For exploitation, we use a fixed number of iterations (i.e., constant n) to perform experiments in Section 5.

We extend SA to improve robustness and composability by representing the current solution as a region [13], instead of a single best parameter setting that gives a maximum objective value. As shown in Figure 3(b), we obtain observables by iterative sampling over the current region represented by the Cartesian product of intervals for each of the parameters. Given the parameter space $\mathbb{P}$, a region $P \in \mathcal{R}(\mathbb{P})$ is a closed convex set $P \subseteq \mathbb{P}$, i.e., if $x, z \in P$ and $x < y < z$, then $y \in P$ and P is finitely representable (e.g., interval-based). Each layer's region has the form $P_{layer} = [p_1^{min}, p_1^{max}] \times ... \times [p_k^{min}, p_k^{max}]$, where $[p_k^{min}, p_k^{max}]$ represents the interval for parameter p_k.

We subsequently refine the region to achieve a given goal (e.g., maximizing the average objective values for performance, maximizing the minimum objective values for robustness). SA generates new candidate solutions within a neighborhood (i.e., current region). Based on the samples available and the given refinement ratio τ $(0.0 < \tau < 1.0)$, P' is a possible refinement of P if $size(P') = size(P) \cdot \tau$ and P' has an available sample in the center (see Figure 3(a)). The refinement that maximizes the objective based on its enclosed samples becomes the new region for the next iteration. We define our compositional optimization based on this representation in the following section.

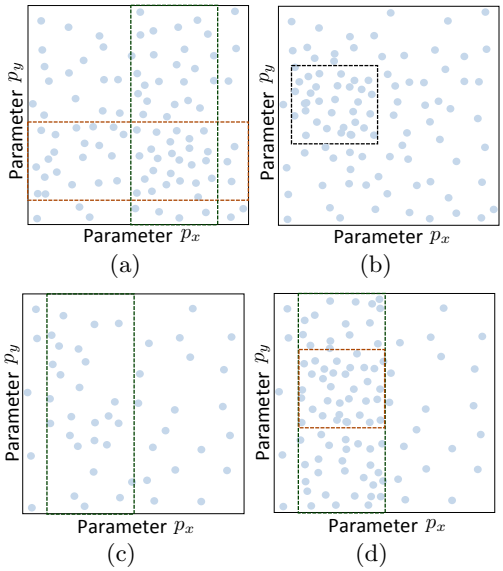

Figure 4: Solution refinement of (a) local vs. (b) global vs. (c)(d) compositional optimization.

4.2 Cross-Layer Composition

To support online cross-layer optimization that computes the refined parameter settings, a constraint refinement approach [13] allows encapsulation of detailed system optimization information. In Figure 2, the key idea underlying the compositional optimization is to exchange the local optimizers' solutions for a more informed parameter selection. More specifically, each local optimizer uses the other optimizer's refinement results as its constraints. As an example, if the security layer optimizer refines the PAC parameter α to [20, 60] and β to [15, 45], then the caching layer optimizer refines its parameter L to [5, 10], taking the security layer parameter ranges as constraints. The caching layer results are transmitted to the other layers' optimizers for further refinement. Thus, constraints can be used as the generic interface among different local optimizers, leading to improvement of solution quality at low complexity.

Figure 4 shows a simple example of solution refinement of local vs. global vs. compositional optimization, where each layer optimizes p_x and p_y, respectively. Solution refinement is a sequence of exploited sub-regions (of the parameter space) that satisfy a given goal using the interval-based representation. The input (P) and output (P') of each refinement step are regions. With local optimization, depicted in Figure 4(a), the refinement proceeds without considering other layers, as illustrated with dashed boxes. The intersection of refined regions is the set of admissible parameter settings at termination. Global optimization, depicted in Figure 4(b), samples over the entire parameter space to find an optimal solution. Our compositional approach lifts the level of abstraction by treating P as *constraints* when we restrict the resampling space to find P'. With the same number of samples generated by SA, compositional optimization shown in Figure 4(c) optimizes p_x and restricts the sampling space of other layers as shown in Figure 4(d).

Note that compositional optimization through constraint refinement enables a controller to coordinate existing opti-

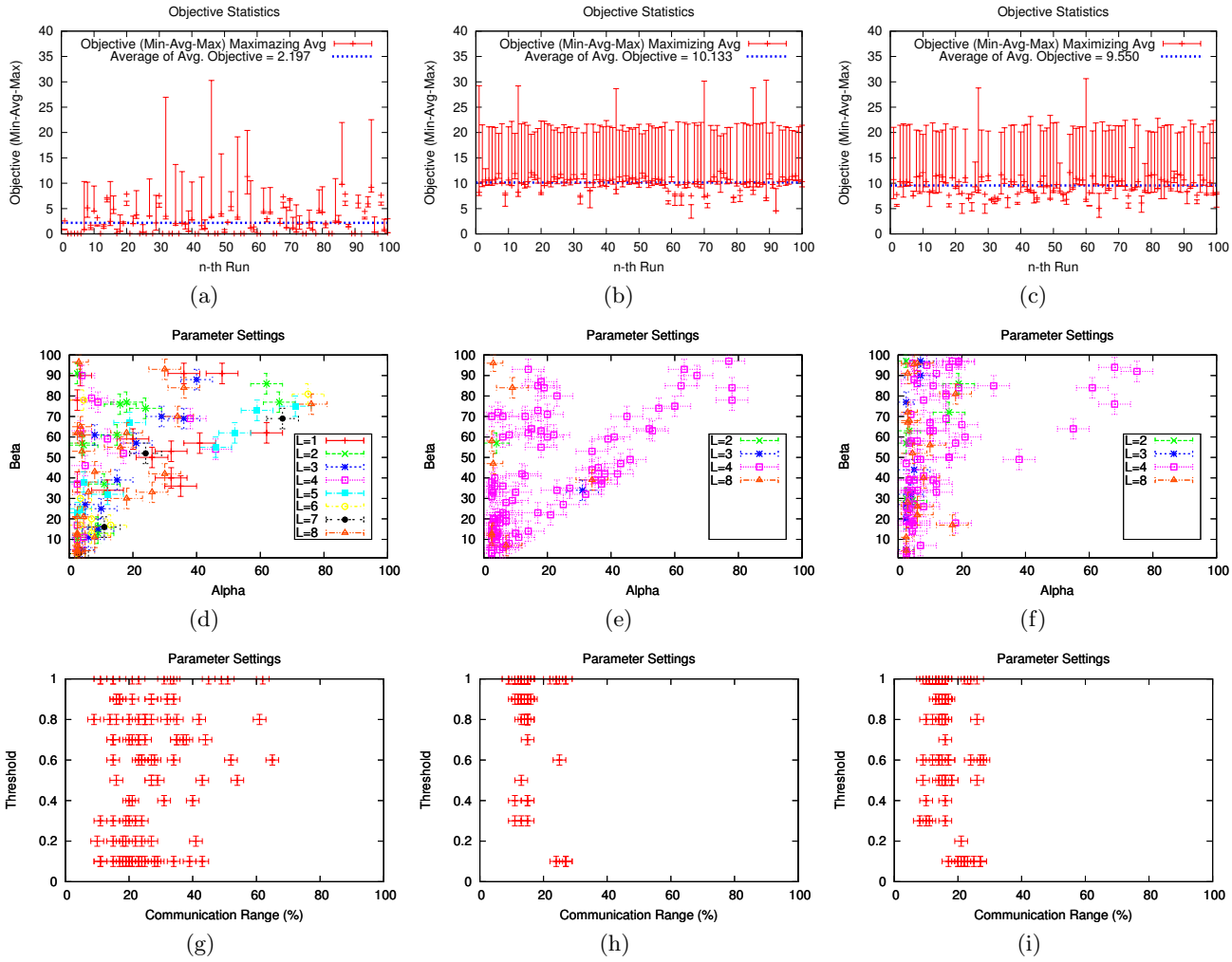

Figure 5: Objective statistics and parameter settings of (a)(d)(g) local vs. (b)(e)(h) global vs. (c)(f)(i) compositional optimization. (a)-(c) show that a local optimization leads to significantly low objective values compared that of global and compositional optimizations. (d)-(f) show that global and compositional optimizations can refine the replication parameter (L) to 4 with appropriate security layer parameter settings (alpha, beta) while a local optimization cannot find the stable parameter settings. (g)-(i) show that both global and compositional optimizations find upper left corner (i.e., higher threshold and lower range) as proper settings for hardware layer parameters (threshold, range).

mizers that can potentially have conflicting objectives and be possibly distributed. Treating local optimizers as black boxes permits processing different objectives in parallel. Different solutions obtained in parallel can be unified by taking the intersection, which corresponds to the conjunction at the symbolic level of constraints.

4.3 Implementation within Haggle

Distributed monitoring involves measurements of local observables, across different layers — e.g., location of content, sources of interest, node density, degree of mobility, residual energy, bandwidth consumption — which can be compactly disseminated as system metadata. Among many observables, we focus on latency, energy consumption, bandwidth, and reliability as a measure of availability. We add information to each Haggle node description with a particular tag (e.g., <attribute=energy, value=80>) to exchange the observables.

Local utility-based optimizers (Section 3) are implemented in various managers of Haggle as we explained earlier. For the cross-layer optimization, we implemented the compositional method in this section within the resource manager of Haggle. In the resource manager, the monitoring module collects the observables in a distributed manner and keeps them up to date with suitable aging mechanisms. The monitoring module maintains the mapping from the parameter settings to the observables. The optimizing module then consults the monitoring module to compute the effect of parameter settings (i.e., objective values). It specifically combines SA with constraint refinement to enable compositional optimization across layers.

5. EXPERIMENTS

We test our prototypical implementation on the CORE network simulator [1]. We assume that 20 nodes move according to the random waypoint mobility model. An esti-

mate of the probability that a node misbehaves (μ) is computing the fraction of nodes to which it is connected. Since this value is different for each node at a particular point in time, we average it across all nodes. The value is normalized over the length of the simulation. The bandwidth used (x_b) is measured by the number of bytes transferred in the simulation. Given μ and parameter settings of α and β, the reliability estimate (x_r) is modeled in the security manager and observed by the monitoring module. The transmission range of a node (r) is a parameter of the simulation, and the residual energy of a node (x_e) is derived using a free space model. We measure the average latency (x_l) of a node at the application level. The fine-grained instrumentation code for collecting the observables (e.g., per content latency) at the Haggle daemon level is ongoing work. As a first step toward online optimization, we pre-train the monitoring module to provide observables instantaneously.

We study the effect of composition as a coordination mechanism for cross-layer optimization. For the baseline, we compare local optimization without interaction and global optimization in terms of the availability and parameter settings in Figure 5. Compositional cross-layer optimization presents reasonably close solutions to the global approach in the sense that the average objective value of the compositional approach resides between that of local and global optimization. The relative execution time of the compositional approach is longer than the local optimization (without any coordination) and shorter than the global approach (most complex). Note that the refined parameter setting as determined by local optimization is very different from that of the global approach while compositional optimization gives similar results.

6. RELATED WORK

A variety of techniques have been developed to trade the use of authentication, signing, and encryption, at various layers in network communication [22]. The techniques may reconfigure application-layer protocols, such as SSH and SSL, Internet protocols, such as IPsec, and MAC layer protocols, such as WEP and WPA, to avoid redundant security assurance [5]. Some of the earlier schemes trade security with other aspects of system utility, much in the way our work does. However, unlike earlier schemes, our work trades the security characteristics of the system using parameterized access control, which is a property of the data objects, rather than the network links they traverse.

Content-based routing and caching solutions for disruption-tolerant networking (DTN) [7, 14, 25, 18] require resource provisioning to determine storing or forwarding of a particular piece of content to maximize its availability. Quantifying the benefit and cost of such operations can be formulated as a utility maximization problem [24, 2]. Our compositional optimization improves content-based utility by treating individual layers as modules, which makes it easier for further generalization to incorporate various local optimizers, such as different routing or caching schemes.

Cross-layer optimization under constraints has been studied in networking previously. Examples include formulating the network resource allocation problem as a cross-layer control of transmission strategies [20] and modeling the network as a utility maximization problem by layered decomposition [3]. While other work focused on the architectural decisions, a resource allocation framework [26] aimed at tuning the system parameters across layers for energy-QoS-security gain. However, the solution requires full awareness of the system dynamics (i.e., global optimization), which leads to high complexity unlike our compositional method that trades local utilities with each other.

7. CONCLUDING REMARKS

By integrating existing policies on secure content dissemination and resource provision across all layers, we have analyzed their availability implications and presented a technique to facilitate information access in a situation- and resource-aware manner. The prototype version implemented in the Haggle framework has the advantage of being agile and flexible, which enables our work to be extended for more complex operating scenarios and protocol optimizations. Even though we mainly focus on the improvement of system availability and the utilization of limited resources in this paper, the proposed approach of distributed monitoring and cross-layer optimization is also more generally useful for systems with significant uncertainty or failures due to unreliable components and physical phenomena as is typical for cyber-physical systems.

The overhead of monitoring (e.g., to what extent full system information is available, and at what cost) and runtime aspects of the approach will have to be included and traded within the parameter space. We are currently extending our composition methods to handle a variety of utility functions (e.g., role-based information access) with composite metrics (e.g., rate and latency of content delivery) for more concrete measure of availability. For higher-level measures, light-weight information fusion and aggregation techniques need to be developed, as they are used in sensor networks. We plan to explore content linkage to reduce the latency proactively (e.g., store in proximity or prefetch). Furthermore, the optimization objective should be adaptive to accommodate the application behavior and interest.

We also plan to improve our models to include real-world implementations of parameterized security and energy management on Android devices. Applying our compositional method with routing mechanisms for efficient content delivery (e.g., potential-based routing [4], interest-driven routing [23]) and network coding for MANETs [15] is another interesting avenue. The declarative networking framework [11] and its logical foundation [12] for control and optimization of cyber-physical systems can also benefit from this approach. For instance, the compositional method can leverage the partially ordered knowledge sharing model by integrating it into the PADO (Parallel And Distributed Optimization) framework [9].

Acknowledgments

This material is based in part upon work supported by the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001, under the auspices of the Institute for Information Infrastructure Protection (I3P) research program. The I3P is managed by Dartmouth College. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security, the I3P, or Dartmouth College.

This material is based upon work supported by the National Science Foundation under Grants CPS-0932397 and IIS-1116414. Any opinions, findings, and conclusions or recommendations ex-

pressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Additional support from the National Research Foundation of Korea (NRF) Grant No. 2010-0026511 funded by the Korean Government (Ministry of Education, Science and Technology) is gratefully acknowledged.

8. REFERENCES

[1] http://cs.itd.nrl.navy.mil/work/core.

[2] A. Balasubramanian, B. Levine, and A. Venkataramani. DTN routing as a resource allocation problem. In *Proc. Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '07. ACM, 2007.

[3] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle. Layering as optimization decomposition:a mathematical theory of network architectures. In *Proceedings of the IEEE*, volume 95, pages 255–312, Jan. 2007.

[4] S. Eum, K. Nakauchi, T. Usui, M. Murata, and N. Nishinaga. Potential based routing for ICN. In *Proc. 7th Asian Internet Engineering Conf.*, AINTEC '11, pages 116–119. ACM, 2011.

[5] F. Foukalas, V. Gazis, and N. Alonistioti. Cross-layer design proposals for wireless mobile networks: a survey and taxonomy. *Communications Surveys & Tutorials, IEEE*, 10(1):70–85, 2008.

[6] A. Gehani and S. Chandra. Parameterizing access control for heterogeneous peer-to-peer applications. *3rd International Conference on Security and Privacy in Communication Networks (SecureComm)*, 2007.

[7] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *CoNEXT '09: Proc. 5th Int. Conf. on Emerging Networking Experiments and Technologies*, pages 1–12. ACM, 2009.

[8] J.-C. Kao and R. Marculescu. Minimizing eavesdropping risk by transmission power control in multihop wireless networks. *IEEE Trans. Comput.*, 56(8):1009–1023, 2007.

[9] J. Kim, M. Kim, M.-O. Stehr, H. Oh, and S. Ha. A parallel and distributed meta-heuristic framework based on partially ordered knowledge sharing. *ELSEVIER Journal of Parallel and Distributed Computing (JPDC)*, 72(4):564–578, 2012.

[10] M. Kim, M.-O. Stehr, A. Gehani, and C. L. Talcott. Ensuring security and availability through model-based cross-layer adaptation. In *UIC*, volume 6905 of *Lecture Notes in Computer Science*, pages 310–325. Springer, 2011.

[11] M. Kim, M.-O. Stehr, J. Kim, and S. Ha. An application framework for loosely coupled networked cyber-physical systems. In *8th IEEE Int. Conf. Embedded and Ubiquitous Computing (EUC'10), 2010*.

[12] M. Kim, M.-O. Stehr, and C. Talcott. A distributed logic for networked cyber-physical systems. In *Proc. IPM Int. Conf. on Fundamentals of Software Engineering*, FSEN'11. Springer-Verlag, 2011.

[13] M. Kim, M.-O. Stehr, C. Talcott, N. Dutt, and N. Venkatasubramanian. Constraint refinement for online verifiable cross-layer system adaptation. In *DATE '08: Proc. Design, Automation and Test in Europe Conf. and Exposition*, 2008.

[14] R. Krishnan, P. Basu, J. M. Mikkelson, C. Small, R. Ramanathan, D. W. Brown, J. R. Burgess, O. L. Caro, M. Condell, N. C. Goffee, R. R. Hain, R. E. Hansen, C. E. Jones, V. Kawadia, D. P. Mankins, B. I. Schwartz, W. T. Strayer, J. W. Ward, D. P. Wiggins, and S. H. Polit. The SPINDLE disruption-tolerant networking system. In *IEEE Military Communications Conference*, 2007.

[15] U. Lee, J.-S. Park, S.-H. Lee, W. W. Ro, G. Pau, and M. Gerla. Efficient peer-to-peer file sharing using network coding in MANET. *J. Communications and Networks (JCN), Special Issue on Network Coding*, 10(4):422–429, Dec. 2008.

[16] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, July 2003.

[17] L. Nolle, A. Goodyear, A. A. Hopgood, P. D. Picton, and N. S. J. Braithwaite. On step width adaptation in simulated annealing for continuous parameter optimisation. In *Proc. Int. Conf. 7th Fuzzy Days on Computational Intelligence, Theory and Applications*, pages 589–598, 2001.

[18] E. Nordstrom, P. Gunningberg, and C. Rohner. A search-based network architecture for mobile devices. Uppsala University, 2009.

[19] J. Reich and A. Chaintreau. The age of impatience: Optimal replication schemes for opportunistic networks. In *Proc. Int. Conf. Emerging Networking Experiments and Technologies*, CoNEXT '09, pages 85–96, New York, NY, USA, 2009. ACM.

[20] M. V. D. Schaar and S. Shankar. Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms. *IEEE Wireless Communications*, 12:50–58, 2005.

[21] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[22] S. Sharma, R. Mishra, and K. Singh. A survey on cross layer security. In *IJCA Proceedings on National Conference on Innovative Paradigms in Engineering and Technology (NCIPET 2012)*, number 5. Foundation of Computer Science (FCS), 2012.

[23] I. Solis and J. J. Garcia-Luna-Aceves. Robust content dissemination in disrupted environments. In *Proc. Third ACM Workshop on Challenged Networks*, CHANTS '08, pages 3–10. ACM, 2008.

[24] T. Spyropoulos, T. Turletti, and K. Obraczka. Utility-based message replication for intermittently connected heterogeneous networks. In *WOWMOM*, pages 1–6. IEEE, 2007.

[25] V. Kawadia, N. Riga, J. Opper, and D. Sampath. Slinky: An adaptive protocol for content access in disruption-tolerant ad hoc networks. In *ACM MobiHoc 2011 International Workshop on Tactical Mobile Ad Hoc Networking*, 2011.

[26] W. Wang. Quality-driven cross layer design for multimedia security over resource constrained wireless sensor networks. Ph.D. dissertation, University of Nebraska, Lincoln, Dept. of Computer and Electronics Engineering, 2009.