

## **A KALMAN FILTER-BASED PREDICTION SYSTEM FOR BETTER NETWORK CONTEXT-AWARENESS**

James Haught  
Kenneth Hopkinson  
Nathan Stuckey  
Michael Dop  
Alexander Stirling

Institute of Technology  
Department of Electrical and Computer Engineering  
2950 Hobson Way  
Air Force WPAFB, OH 45433, USA

### **ABSTRACT**

This article investigates the use of Kalman filters at strategic network locations to allow predictions of future network congestion. The premise is that intelligent agents can use such predictions to form context-aware, cognitive processes for managing communication in mobile networks. Network management is improved through the use of context-awareness, which is provided through rough long or mid-term plans of operation and short-term predictions of network state and congestion levels. Research into incorporating an intelligent awareness of the network state enables a middleware platform to better react to current conditions. Simulations illustrate the advantages of this techniques when compared to traditional mobile network protocols, where the general assumption is that nothing is known about the mobility or communication patterns of the mobile entities and the network is often treated as an opaque black box. Our approach shows promise for improved network management.

### **1 INTRODUCTION**

Current networking technology limits a network's ability to adapt to changes and interactions in the network, often resulting in sub-optimal performance. Limited in state, scope and response mechanisms, the network elements (consisting of nodes, protocol layers, policies and behaviors) are unable to make intelligent adaptations to meet network-wide goals. Communication of network state information is stifled by the layered protocol architecture, making individual elements unaware of the network conditions experienced by other elements. Any response that an element may make to network stimuli can only be made inside of its limited scope. The adaptations that are performed are typically reactive, taking place only after a problem has occurred.

There exists, across the field of computer networking the need to achieve network-level objectives in the face of increasing network complexity. Particularly in wireless networks, there has been a trend towards increasingly heterogeneous and dynamic environments. Researchers at the Air Force Institute of Technology have been investigating a radical new paradigm, the cognitive network using distributed intelligent agents, to autonomously and dynamically achieve complex network level objectives in a wireless network. A network with distributed intelligence utilizes cognition in the network – defined loosely as the ability to perceive current conditions, and then plan, decide and act on them – to learn to make decisions that take into account end-to-end goals. The agents in the cognitive network cooperate in a peer-to-peer manner to create an intelligent distributed system. Making this vision a reality requires advances in intel-

ligent network optimization, wide-area network monitoring, and distributed routing. This article concentrates, in particular, on a wide-area network monitoring and prediction system, which can form an important input into the type of distributed agent framework that we envision.

To provide the network level objective that these decisions support, this research investigates the broader problem of linking users to the networking infrastructure that they operate on. Our main idea is that by giving users a better picture of the status and overall capabilities of the network, the cognitive (i.e. intelligent) processes in the distributed network will be given clearer, more accurate inputs of the mission objectives. An illustration of this is given in Fig. 1, showing how current conditions are combined with user preferences and network objectives to determine the best network management to meet user goals. A major premise in this work is that, while this type of network management may not easily apply directly to large chaotic networks, like the Internet, many special-purpose networks for corporations, critical infrastructure management, and for military command and control could be better predicted and controlled than they are today with the right inputs and distributed management framework.

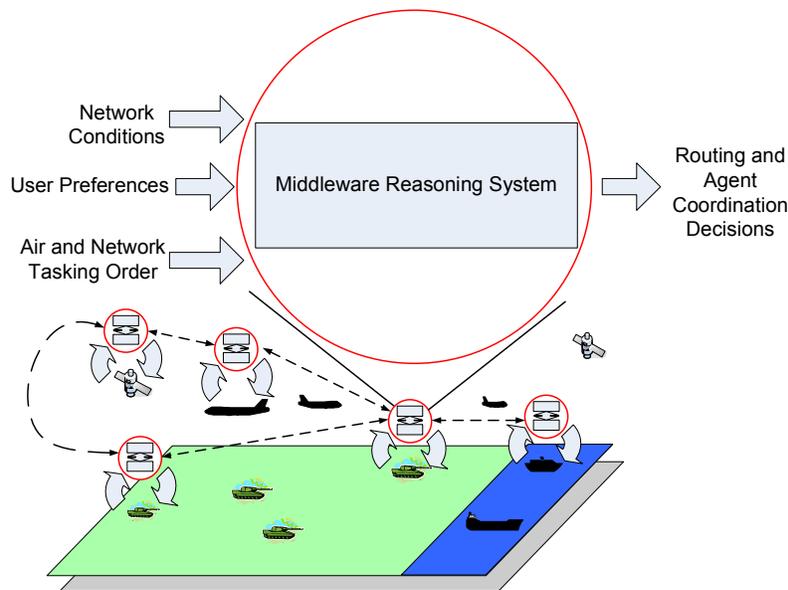


Figure 1: An illustration of a mobile network using middleware to enhance the reliability and quality of service (QoS) properties of the system

Major elements of this vision have been created over the past few years. Military tactical environments are typically planned at least a day ahead, in a document called an Air Tasking Order, such information could be incorporated into what we call a Network Tasking Order in order to feed medium to long-term information into our agent-based network management framework. Previous articles have shown that such information could enable more optimal network outcomes if it were available (Compton, Hopkinson, and Graham 2008; Gocmen, Hopkinson, and Compton 2009; Tiwari et al. 2009). Past work has also demonstrated the ability of an agent-based framework to optimize network behavior using long-term, mid-term, and short-term estimates of network behavior (Pecarina 2008).

This article looks at ways to make medium-term estimates of the network, over a time period of perhaps a minute and a half, available by using Kalman filters in outbound queues in routers. While router queues tend to be relatively small, simulations show that their behavior can be used to predict future behavior with accuracy that is often good enough. Previous work looked at toy networks with just two nodes to test this idea for Kalman filtering prediction (Stuckey 2007). This article expands on that work to look at the feasibility of using such predictions in more complex networks.

## 2 ESTIMATION AND CONTROL THEORY

This section reviews Kalman filter-based network prediction algorithms, which previously appeared in (Stuckey 2007; Stuckey et al. 2007). Network control algorithms have some form of feedback such as Figure 2, which shows the basic structure of a feedback controller (Maybeck 1982). The goal of a feedback controller is to provide the control input  $\mathbf{u}$  to the dynamic system such that the controlled variables  $\mathbf{y}_c$  match the reference signal  $\mathbf{y}_d$  as closely as possible. Dynamic disturbances  $\mathbf{n}$  also effect the dynamic system, usually in an undesirable way. In order to observe these disturbances, measurements  $\mathbf{Z}$  of the dynamic system are taken and are fed back to the controller. These measurements may correspond to the controlled variables, but in many cases not all of the controlled variables will be measured. The control  $\mathbf{u}$  is computed based on feedback the measurements provide about the state of the dynamic system. The measurements in general are not perfect due to the measurement corruptions  $\mathbf{n}_m$ .

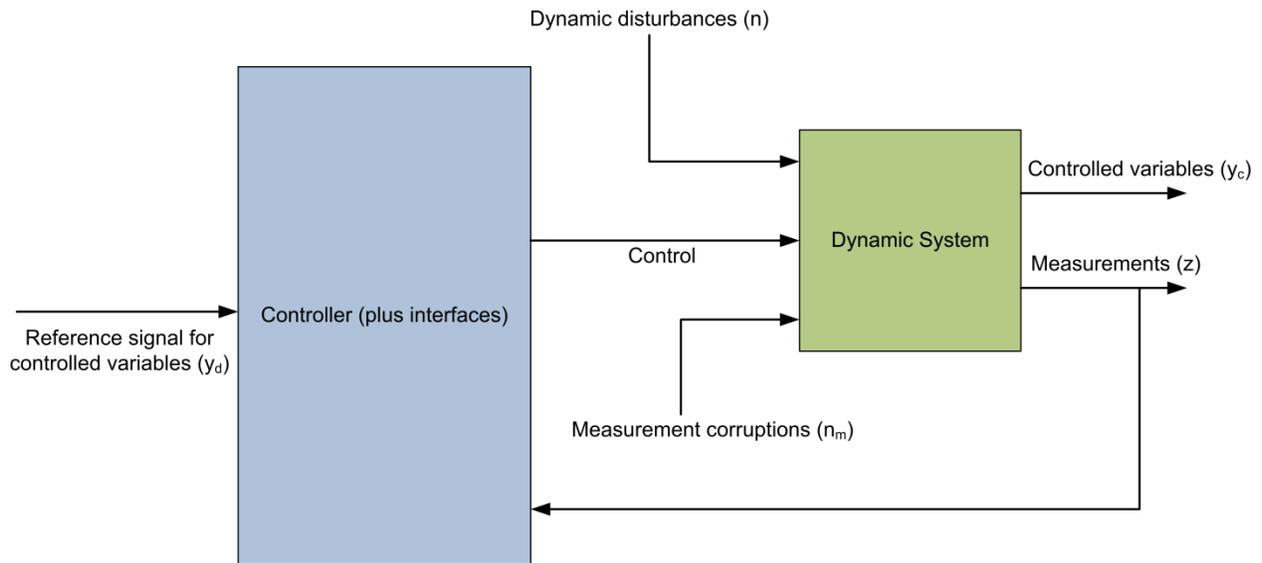


Figure 2: Controlled system configuration (Stuckey 2007)

In general, computer networks can be thought as queuing networks. To demonstrate the feasibility of applying stochastic control theory to computer networks, a controller is developed in this article which regulates queue sizes by controlling the packet arrival rate to the network queues. As part of this controller, a Kalman filter is developed to estimate both the size of a network queue and the total packet arrival rate to a network queue, given sample data measurements of the queue size.

The network model is a nonlinear discrete model with discrete measurements. In order to develop a feedback control system, a discrete-discrete (discrete input and output) extended Kalman filter is required. Equation 1 and 2 provides a dynamic discrete-time system model that describes the transient behavior of the network queue of the Kalman filter.

$$\mathbf{x}(t_i) = \Phi(t_i, t_{i-1}, \mathbf{x}(t_{i-1})) + \mathbf{w}_d(t_i) \quad (1)$$

Let  $n$  be the number of states and  $m$  be the number of measurements. Then,  $\mathbf{x}(t_i)$  is an  $n$ -dimensional vector describing the state of the system at time  $t_i$ . The non linear dynamics equation for the transition of the states of the system for time  $t_{i-1}$  to  $t_i$  is  $\varphi$ , which is also an  $n$ -dimensional vector. The dynamics noise  $\mathbf{w}_d$  represents the unknown system dynamics not included in  $\varphi$  and is an  $n$ -dimensional function containing discrete-time white Gaussian noise of zero mean and covariance kernel

$$E\{w_d(t_i)w_d^T(t_j)\} = \begin{cases} Q_d & t_i=t_j \\ 0 & t_i \neq t_j \end{cases} \quad (2)$$

where  $Q_d$  is an  $n$ -by- $n$  matrix representing the covariance of  $w_d$ . The discrete-time measurement model is given by Equations 3 and 4; linear measurements are assumed sufficient:

$$z(t_i) = Hx(t_i) + v(t_i) \quad (3)$$

where  $z$  is the  $m$ -dimensional measurement vector and  $H$  is the  $m$ -by- $n$  measurement matrix. Note that  $H$  is assumed constant for this development, but in general can be time varying. The measurement noise, which represents the uncertainty of measurement, is an  $m$ -dimensional vector containing discrete-time white Gaussian noise of zero mean and covariance kernel

$$E\{v(t_i)v^T(t_j)\} = \begin{cases} R & t_i=t_j \\ 0 & t_i \neq t_j \end{cases} \quad (4)$$

where  $R$  is an  $m$ -by- $m$  matrix representing the covariance of  $v$ . The dynamics noise  $w_d$  and measurement noise  $v$  are reasonable assumed to be independent. The linearized state transition matrix, which is an  $n$ -by- $n$  matrix, can be found by

$$\Phi(t_i, t_{i-1}) = \left. \frac{\partial \phi}{\partial x} \right|_{x=\hat{x}(t_{i-1}^+)} \quad (5)$$

where  $\hat{x}(t_{i-1}^+)$  is the state estimate provided by the Kalman filter at time  $t_{i-1}$  after the measurement update. From Eq. (5), the element of  $\Phi$  are calculated by

$$\Phi_{ij}(t_i, t_{i-1}) = \left. \frac{\partial \phi_i}{\partial x_j} \right|_{x=\hat{x}(t_{i-1}^+)} \quad (6)$$

Since  $\phi$  is not continuous, Eq. (6) must be approximated by replacing the partial derivative with a two-sided difference equation. For example,  $\Phi_{11}$  is calculated by

$$\Phi_{11} = \left. \frac{\partial \phi_1}{\partial x_1} \right|_{x=\hat{x}(t_{i-1}^+)} \quad (7)$$

$$\approx \frac{\Phi_1 \begin{bmatrix} x + \begin{bmatrix} \Delta x_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \end{bmatrix} - \Phi_1 \begin{bmatrix} x - \begin{bmatrix} \Delta x_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \end{bmatrix}}{2\Delta x_1} \Bigg|_{x=\hat{x}(t_{k-1}^+)}$$

where  $\Delta x_1$  is a small perturbation in  $x_1$ . The value of  $\Delta x$  must be chosen small enough such that the difference equation will give an accurate approximation to the partial derivative, but not so small that numerical precision difficulties occur.

The Kalman filter is a recursive estimator that is constantly in one of the distinct phases, *predict* and *update* which are illustrated in Figure 3. This means that only the estimated state from the previous time-step and current measurement are needed to compute the estimate for the current state with no history of observations or estimates required. The predict phase uses the Kalman filter's state estimate from the previous iteration.

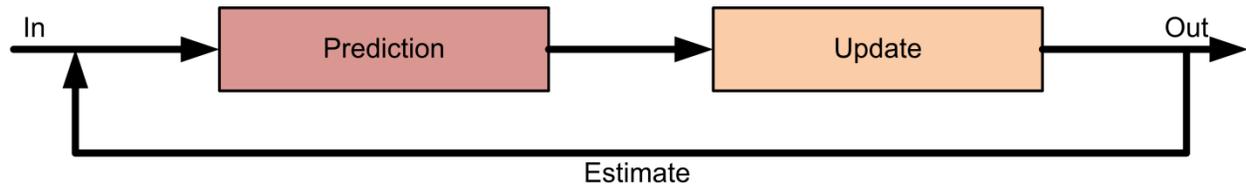


Figure 3: Kalman filter cycle

The state of the Kalman filter is represented by two variables:

- $\hat{x}_{k|k}$  is a *posteriori* state estimate at time  $k$  given observations up to and including at time  $k$ .
- $P_{k|k}$  is a *posteriori* error covariance matrix, which is a measure of estimated accuracy of the state estimate.

The notion  $\hat{x}_{n|m}$  represents the estimate of  $x$  at time  $n$  given observations up to, and including time  $m$ .

The Kalman filter uses a form of feedback control that estimates the process state at some time and obtains feedback in the form of measurements. The Kalman filter uses two types of equations which are the time update equations and measurement update equations. The time update equations project the future state of the current state of the system while error covariance attempts to obtain estimates for the next time step. The measurement update equations updates the innovation covariance, the *posteriori* state estimate, *posteriori* state error covariance, and computes Kalman gain.

### 3 SIMULATION OF KALMAN FILTER PREDICTION

As mentioned earlier, previous studies only looked at Kalman Filter prediction in simple networks with two nodes. In this article, a simulation has been generated in NS2 that reflects the topology in Figure 4. The traffic generated on this network is created and varied by random exponential numbers with a mean of 2000 bytes. Random uniform numbers are used to determine traffic generation rate on the links, with values ranging from a lower bound of 0.16 milliseconds to an upper bound of 0.84 milliseconds. Fifteen total nodes and 33 simulated links were created with this experiment, as well as 85 TCP flows and 15

UDP flows. Each link has a bandwidth of 4 megabytes and a propagation delay of 25 milliseconds. The goal is to create a larger network with more chaotic traffic flows to see how effectively the Kalman Filter system behaves. The Kalman Filter system in this article predicts 5 seconds into the future. Preliminary studies show that such predictions can remain accurate enough for an optimizer to use profitably a minute and a half into the future under typical conditions.

Kalman filters are placed at key locations of the network. In this simulation, Kalman filters were placed the links depicted in Figure 4 because they have the potential for the most frequent traffic congestion, dropped packets, and farthest from optimal bandwidth utilization. The Kalman filters process traffic on those specific links and record data at specified intervals. In this simulation, the sample rate for the Kalman filters is set to 1 second. At each interval, the filter measures the size of the queue and computes the packet arrival rate. Using the current queue size and the previous queue sizes, the Kalman filter produces an estimated prediction on what the queue size will be at the next prediction time. Using these predictions from the Kalman filters, the Routing and Agent Coordination Decision middleware reasoning system will be able to make more informed decisions to increase the utility of the network.

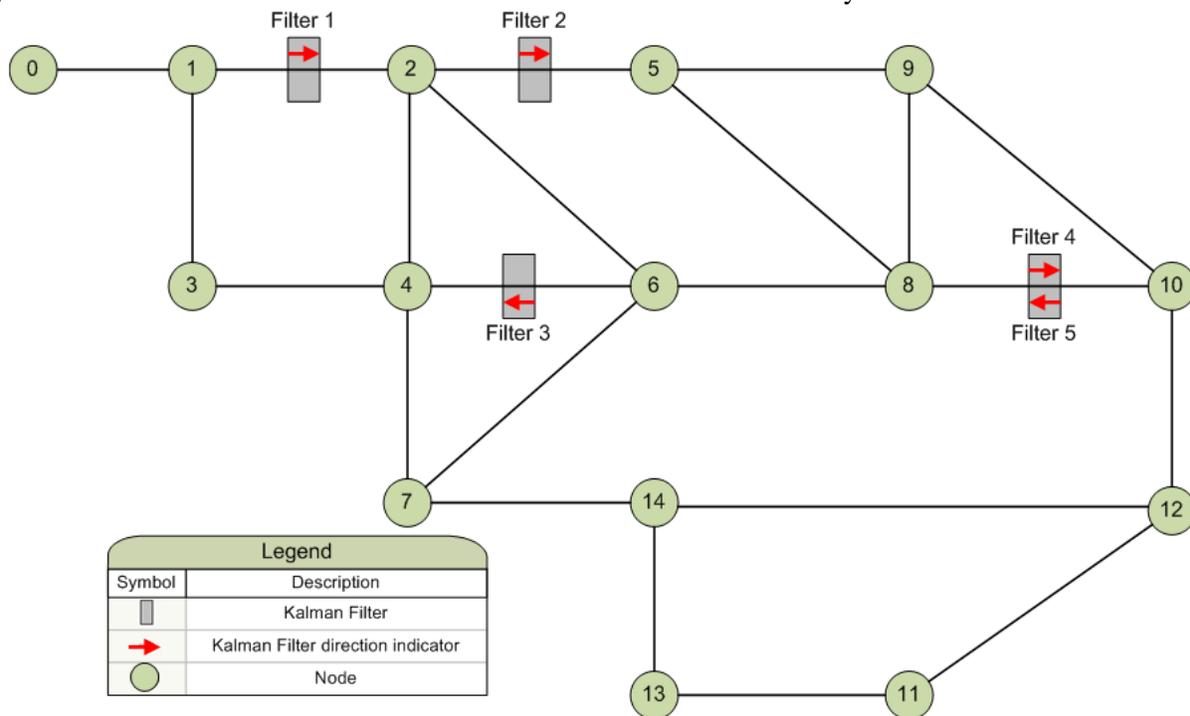


Figure 4: Graphical representation of the network used in the simulation

## 4 RESULTS

The simulation described in section 3 was executed 15 times to ensure consistent performance with random traffic generation and the results from those simulations were averaged to show the accuracy of the predictions. Each one of the Kalman filters maintains a record of its actual queue size and its predicted queue size at every sample time of the simulation. See Figure 5 for the graph of the predicted values versus the actual values. In this simulation, the Kalman filters predicted the queue size 5 seconds into the future and they generated predictions every 5 seconds. As indicated in Table 1, the highest percent error for this simulation is 26.412 percent and the lowest percent error is 2.984 percent. The average percent error for all of the Kalman filters in these 15 iterations is 9.962 percent with a standard deviation of 9.646. Three of the five Kalman filters predicted the queue size within 5.556 percent. The Kalman filter with the highest percent error was off by less than 1 packet on average.

Table 1: Average Actual and Average Predicted queue size, Difference of Averages, and Percent Error for each Queue

	Queue 1	Queue 2	Queue 3	Queue 4	Queue 5
Actual Queue Size Average	32.846	6.962	3.665	7.922	64.358
Predicted Queue Size Average	33.826	7.703	4.633	8.256	60.782
Difference of Averages	0.980	0.741	0.968	0.334	3.576
Percent Error	2.984%	10.643%	26.412%	4.216%	5.556%

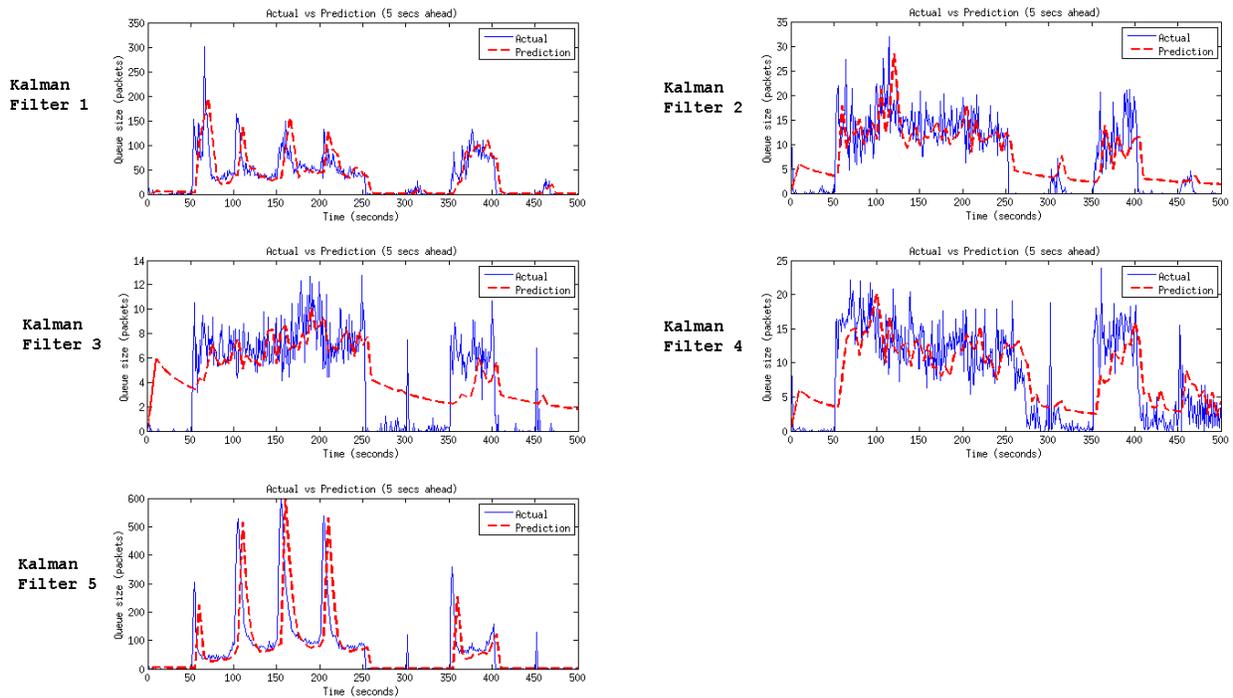


Figure 5: Matlab output showing Actual vs. Predicted queue size for each Kalman Filter (merged from 15 separate runs)

## 5 CONCLUSION

This article has expanded on a Kalman filtering technique for accurately predicting the future state of the network. A Kalman filter is developed within a network controller to estimate both the size of a network queue and the total packet arrival rate given sample data measurements of the queue size. The intent is to use these predictions in conjunction with user preferences and mission objectives to feed into a network management optimization system. Good future predictions over a short term can allow such systems to adjust longer term plans according to current realities.

## DISCLAIMER

The views expressed in this document are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

## REFERENCES

- Compton, M., K. Hopkinson, and S. Graham. 2008. The Network Tasking Order (NTO). IEEE Military Communications Conference, San Diego, CA, USA.
- Gocmen, M., K. Hopkinson, and M. Compton. 2009. The Benefits of a Network Tasking Order in Combat Search and Rescue Missions. IEEE Military Communications Conference (MILCOM), Boston, MA, USA.
- Maybeck, P. S. 1982. "Stochastic Models, Estimation, and Control."
- Pecarina, J. M. 2008. Creating an Agent Based Framework to Maximize Information Utility. Electrical and Computer Engineering. WPAFB, OH, USA, Air Force Institute of Technology. Master of Science: 122.
- Stuckey, N. 2007. Stochastic Estimation and Control of Queues within a Computer Network. Electrical and Computer Engineering. WPAFB, Air Force Institute of Technology. Master of Science.
- Stuckey, N. C., J. R. Vasquez, and S. R. Graham. 2007. Estimating Queue Size in a Computer Network Using an Extended Kalman Filter. IEEE Aerospace Conference, Big Sky, Montana, USA.
- Tiwari, A., A. Ganguli, A. Kothari, S. Avadhanam, M. Compton, and K. Hopkinson. 2009. Is Communication Planning Using Mission Information Logistically Feasible? IEEE Military Communications Conference (MILCOM), Boston, MA, USA.

## AUTHOR BIOGRAPHIES

**JAMES HAUGHT** is currently pursuing a Master of Science in Computer Science at the Air Force Institute of Technology. He received a B.S. in Computer Science from Wright State University in 2009. His interests are in the areas of networking and simulation.

**KENNETH HOPKINSON** received his PhD from Cornell University in Computer Science in 2004. Since that time, he has been an Assistant Professor at the Air Force Institute of Technology. His interests lie in fault-tolerant and reliable distributed systems, networks, and simulation.

**NATHAN STUCKEY** received his Master of Science degree in Electrical Engineering from the Air Force Institute of Technology in 2007. He received a B.S. in Electrical Engineering from Purdue in 2002. His research interests include stochastic estimation and control and computer networks.

**MICHAEL DOP** was an intern at the Air Force Institute of Technology in the summer 2009. He recently received a B.S. in Computer Engineering from Embry-Riddle Aeronautical University. He is interested in computer networks.

**ALEXANDER STIRLING** is an intern at the Air Force Institute of Technology. He is currently pursuing a B.S. in Electrical Engineering at the University of Toledo. His research interest is in the area of computer networks.