8th Conference on Systems Engineering Research
March 17-19, 2010, Hoboken, NJ

Paper #nnn

# Systems Engineering Challenges for Design and Real-time Management of Component-Enabled Wireless Ad-hoc Networks

**Austin M.A., Tabatabaee V. and Baras J.S.,**
Institute for Systems Research,
University of Maryland, College Park, MD 20742, USA.
E-mail: {austin, vahid, baras}@isr.umd.edu

## Abstract

This paper describes systems engineering challenges for the systematic design and real-time management of component-enabled wireless ad-hoc networks. At the front end of development, emphasis is placed on the use of goals and scenarios, and visual modelling abstractions for elicitation of models for required functionality. We assume that network nodes will be implemented as hierarchies of interconnected sub-components. To maximise system reliability and simplify technology upgrades we advocate the use of formal approaches to component interface definition, and to validation of component- and system-level functionalities.

## 1-Introduction

**Problem Statement.** This paper describes systems engineering challenges for the systematic design and real-time management of future component-enabled wireless ad-hoc networks. Good solutions will provide a desirable balance of system functionality, performance and economics (not necessarily optimal in any one dimension), reliable operation in a wide range of environments, and ease of accommodation for future technical improvements.

Wireless network behaviours are determined by protocols; that is, specifications for communication that will occur over a network connection. The step-by-step procedure for development involves: (1) specification, (2) abstraction, (3) verification, and (4) performance evaluation. The specification is a formal (or semi-formal) description for what the protocol will do. Abstractions are concerned with the behaviour, organization and connectivity of the network elements. Verification determines whether or not the protocol is logically consistent. And finally, performance studies provide and assessment of the protocol and network operational efficiency.

While there has been impressive progress in wireless communication radio technologies, we still need systematic methodologies and toolkits for the design, analysis, dimensioning and management of wireless networks to have predictable and controllable performance. Two of the main reasons for this state of affairs are the inherent uncertainty and variability of the wireless medium, and the coupling between the performance metrics of wireless links. In wired networks, link capacities are fixed -- unless there is a failure there is no variation in network topology and link capacities. The operation of wireless networks, in contrast, is complicated by link capacities that depend on a variety of temporal and spatially dependent factors such as mobility, interference, and terrestrial characteristics. Wireless networks also need to be amenable to technology upgrades in both the network hardware and software. A central challenge in the design of wireless networks is formulation of strategies that can overcome the additional complexities introduced by these factors. It is well known, for example, that the design assumption of independent layered protocols is overly simplified, resulting in network performance and agility that is far from optimal. Dependencies between the link capacities and the MAC layer and physical layer interferences have motivated the development of a variety of cross-layer analysis and design methods (Baker 1982, Clark 1990). However, many of the proposed algorithms are either too complex or based on unrealistic assumptions. A second fundamental problem with cross-layer methods is a lack of modularity, which is essential for the agile design and long-term management of any complex system (Kawadia and Kumar, 2005).

**Objectives and Scope.** The long-term objective of our work is development of methodologies and systems for design and management of component-enabled wireless ad-hoc networks.. The following properties and requirements apply:

1- The methods and systems should consider cross-layer effects, i.e., model cross-layer interdependence of the communication layers performance. For both short- and long-time scales, network performance needs to be controllable and predictable (not necessarily optimal) performance.
2- They should provide a formal systematic methodology for testing and verification of designed protocols.
3- Assessments for system functionality should be scenario-driven, i.e., perform analysis and design of networks for a given dynamic scenario and user specification, including mobility pattern, traffic demands, performance requirements, etc.
4- To the extent possible, design solutions should have functionality and performance that is insensitive to natural variations in scenario inputs and modelling assumptions.
5- It should provide an agile and robust mechanism to modify and enhance the components design, and to switch between alternative designs of a component.

At the front end of development, emphasis is placed on the use of goals and scenarios, and visual modelling abstractions for elicitation of models for required functionality. We assume that network nodes will be implemented as hierarchies of interconnected sub-components. To maximise system reliability and simplify technology upgrades we advocate the use of formal approaches to component interface definition, and formal approaches to validation of component- and system-level functionalities.

Section 2 covers the strengths and weaknesses of top-down development of network specifications, and motivates the need for component-enabled approaches to design. Present-day thinking and capability for component-based wireless ad-hoc network design is covered in Section 3. An overview of component-based OLSR Protocol design is given in Section 4. Finally, a list of systems engineering challenges for wireless ad-hoc networks is presented in Section 5.

## 2-Top-Down Development of Network Specifications

**Top-Down Development.** Established approaches to network design correspond to a top-down development of required system functionality.

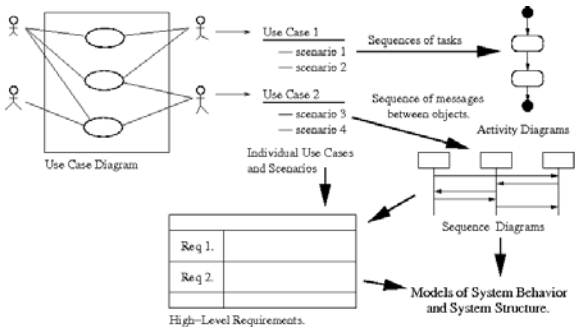


Figure 1. Pathway of system development from operational concepts (i.e., use-cases and textual scenarios) to fragments of system behaviour and requirements (Austin 2009).

Figure 1. shows the essential features of a top-down specification process, beginning with use cases and their elaboration as textual scenarios. Each use case is simply a fragment of required systems functionality. The actors – stick figures shown at the top left-hand side – are external entities who will interact with the network system. They will include the network users, network management, and the network owners. Scenarios are textual descriptions of step-by-step procedures for the implementation of system functionality. Good scenarios will cover normal operation as well as provision for handling of errors such as node and/or link failures. Each scenario is then elaborated visually as either a UML activity diagram or as a sequence diagram. Activity diagrams show the required sequencing of tasks to implement a fragment of system functionality. Sequence diagrams show the message communication among objects to implement required functionality.

Collectively, the use cases, scenarios, and activity and sequence diagrams are statement of required functionality (i.e., what the system will do). Assuming that a network does not exist, metrics of required performance (i.e., how well the system must perform the functionality) are expressed as performance requirements. Interface requirements are implied by support for message passing / communication between objects.

In the lower right-hand side of Figure 1, more complete models of behaviour are created by combining fragments of functionality. System structure alternatives are created in response to connectivity requirements: (1) to communicate with external actors, and (2) for component-to-component communication within the systems. Systems design alternatives correspond to the mapping of system behaviour onto system structure.

**Benefits and Limitations of Top-Down Development.** The two main benefits of top-down development of wireless networks are: (1) ease of development – this is the way humans naturally approach problems, and (2) opportunity for customization – a network can be designed to do exactly what is required and no more. Certainly items 1 and 2 are appealing for the design and management of wired networks.

However, over time, top-down development procedures result in systems that are too costly and too rigid. Costs are high because the model of system development permits each iteration of work to start from scratch. Since reuse of previous work is not mandated at any level, additional costs are incurred from the testing of network nodes and then collections of nodes making up the network. Systems are rigid in the sense of having architectures or components that cannot be easily adapted in response to changing requirements.

**Expectations and Limitations of Object Development.** The benefits and limitations of top-down development have been well known for more than three decades. Then, beginning in the mid 1980s, proponents of object-oriented software development claimed that issues of system scale and reuse could be kept in check through the application of object-oriented principles (i.e., data abstractions and methods, classes and objects, mechanisms of inheritance, polymorphism, and so forth). Significant improvements in system development productivity were also promised. Now, two decades later it is evident that some of these early promises will not work out. While it is true that mechanisms of inheritance and programming by extension lead to efficiency of implementation, the underlying relationships among classes in a class hierarchy are quite rigid. Hence, system development through programming by extension only works well if these relationships are relatively static. A key problem is that during the past two decades our expectations of communications systems have expanded – particularly in the dimensions of mobility and technical capability -- beyond what anyone could have reasonably envisioned.

All is not lost, however. Ousterhout reports that in the software world, the required improvements in productivity can be achieved through the use of scripting languages, where system descriptions are created through a bottom-up composition of software elements (Ousterhout, 1998). Present-day scripting techniques are simplified through homogeneous representations of data. We surmise that future high-performance wireless network systems will have dynamic system architectures composed from heterogeneous elements. Their synthesis will correspond to top-down specification combined with bottom-up assembly of reusable components.

## 3-Component-Based Wireless Ad-hoc Network Design

The main idea behind component-based modelling and design in system engineering is to divide a system into components and sub-components to separate the design concerns and provide a framework for systematic and modular design of large complex systems. Our goals here are to devise a component-based approach for design and analysis of network protocols.

Broadly speaking a component is an independent deliverable piece of functionality that provides access to its services through interfaces. In a departure from the goals of object-oriented system development, system development with components is primarily concerned with design and assembly of solutions as a collection of interacting pieces.

**Component Structure.** As illustrated in Figure 2 below, the wireless network nodes will be implemented as a multi-level hierarchy of interconnected sub-components.

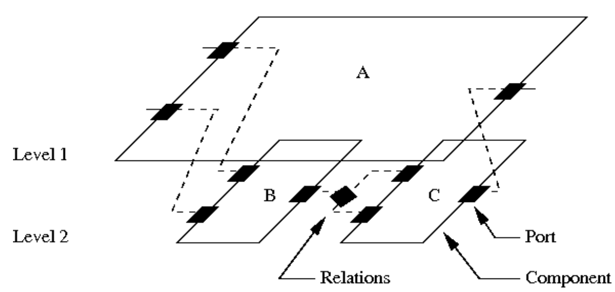


Figure 2. Multi-level interconnect architecture of a wireless network node.

All components and sub-components will have well-defined interfaces through which external entities may interact. Data pathway control can be implemented through logical decision nodes. Components can have sets of attributes whose values can be externally tuneable. For a given set of network requirements, we develop customized solutions, by selecting appropriate combination of components and tuning their parameters accordingly.

We can have alternative solutions with different specifications and performance for each component and use them as building blocks for network solutions. We will develop formal models for checking and validation of each component.

**Component Behaviour.** In object-based systems it is common for software to be implemented as a single thread of control (i.e., as a single executable process). In component-based systems, all components are assumed to operate as active objects – that is, they will each have a thread of control.

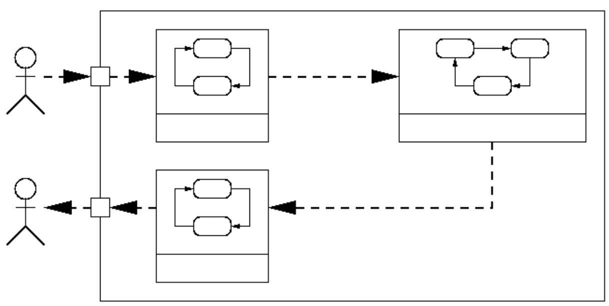

Figure 3. Architecture of an active-object component assembly. The dashed line shows that pathway of data traversal through the component.

As illustrated in Figure 3, this assumption simplifies the implementation of component-level behaviour, but complicates validation of correctness in system functionality because we need to deal with coordination of concurrent (threaded) processes. Once the functionality has shown to be correct, then performance can be evaluated and improved upon using simulation and tradeoff-analysis techniques.

**Interface Based Design:** Interfaces between components play a major role in component based design. Besides components functionality and behaviour, we pay special attention to the component interfaces and their specifications in the formal and performance model. The interfaces of formal models specify how we can use components and interconnect them, and how they interact with environment. The performance models interfaces specify the performance metrics their sensitivities and characteristics that affect other components performance.

## 4-Component-Based OLSR Protocol Design and Modification

The Optimized Link State Routing Protocol (OLSR) is developed for mobile ad hoc networks. It operates as a table driven and proactive protocol, and as such, exchanges topology information with other nodes of the network regularly. A subset of nodes serve the role of multipoint relay (MPR); subsequently, neighbour nodes announce this information periodically in their control messages. Thereby, a node announces to the network, that it has reachability to the nodes which have selected it as MPR. In route calculation, the MPRs are used to form the route from a given node to any destination in the network. The protocol uses the MPRs to facilitate efficient flooding of control messages in the network. OLSR inherits the concept of forwarding and relaying from HIPERLAN (a MAC layer protocol) which is standardized by ETSI (OLSR, 2009).

**Network Components and Architecture.** Two of the critical requirements for wireless tactical networks are adaptability and agility. Accordingly, instead of a holistic approach to network design and modelling, we divide each network layer to components with limited, but specific functionality, and provide methodologies for design and modelling of each component. The main components (layers) are routing, scheduling, MAC and PHY components.

Figure 4. shows the main components, routing sub-components, and the flowchart of functionality (Baras et al., 2009a, Baras et al. 2009b).

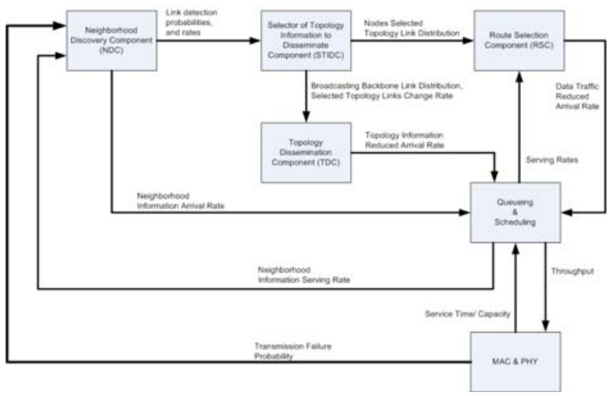


Figure 4. Components and flowchart of functionality for component-enabled OLSR protocol development.

The routing sub-components are as follows: *Neighborhood Discovery Component (NDC)* is responsible for detection of neighbour nodes and other local network node information. *Selector of Topology Information to Disseminate Component (STIDC)* selects local information that each node will broadcast to other nodes, which will be used to select paths and/or next-hop(s) for forwarding the packet. *Topology Dissemination Component (TDC)* is responsible for broadcasting of the STIDC gathered information in the network. *Route Selection Component (RSC)*, based on the local information gathered by NDC and the global information from STIDC, selects the path(s) and/or next hops.

Figure 4 also shows the main inputs and outputs of the corresponding component based performance models, their intra-connections and inter-connections with the other system main units. For example, the inputs to the nodes routing models are serving rate and/or blocking probability of packets. The serving rate applies for connection-less protocols, where it is possible that nodes drop packets due to congestion; blocking probability is used for connection oriented protocols with guaranteed packet delivery for accepted connections.

In practice, the exact attributes of the interfaces and cross-layer connections will depend on the selected network performance metrics and the level of modelling details of protocols and their functionalities. In fact, it is often not possible to derive analytical relations between the properties and performance metrics of components and the network performance. That is why the component-based performance models, which are based on analytical and numerical models of components are essential for compositionality.

**Performance Models:** Performance models are formulated for the Routing, Scheduling, MAC and PHY layers. Each component performance model is a multi-valued function whose arguments take component inputs (including component design parameters and outputs of the other components) to the component outputs and performance metrics. Estimates of network node level performance are obtained by interconnecting the component models, thereby accounting for cross-layer effects.

To capture the underlying dynamic nature of the problem, our models use a time-stamped sequence of network input parameters as inputs and use cases. In this way, we can model mobility, changes in network topology (including node and link insertion and deletion), traffic demand, and environment factors (e.g. radio propagation loss parameters). The derived outputs and system performance will also be dynamic and time-varying. Using these scenarios and base lines, we can study performance of the system and/or modify the design to get desirable performance.

The network performance metrics include delay and loss characteristics of links, paths and end-to-end connections in the network Estimates of performance can be obtained through simulation of a specific network topology. Models are derived from component models and additional equations relating to the network architecture (e.g., inter-component connections). Fixed-point iteration can be used to find a consistent solution to the equations. The consistent solution and the performance functions of components provide an implicit mapping between network performance and input parameter values.

We have developed analytical and numerical models for random access and scheduling based MAC protocols and components of OLSR routing protocols (Baras et al. 2008a, 2008b, 2009a, 2009b).

**Sensitivity Analysis for Design:** Estimating the system performance using the fixed-point solution is neither sufficient for design nor trade-off studies. For the latter we also need to quantify reliability and robustness of the solution. The sensitivity parameters between the system performance and input parameters enable us to quantify robustness and predictability of the derived estimates. However, due to the complexity of relations of the component models, it is not possible to compute the derivatives analytically. Numerical methods such as Automatic Differentiation (AD) and Perturbation Analysis (PA) can be used to derive the derivatives and sensitivity parameters. The AD provides the partial derivative of the performance metric (e.g. throughput) with respect to the defined input parameters (i.e., design variables or parameters). This method allows for very complex design parameters to be implicitly embedded in the input function to the AD module. We can also use the computed derivatives in gradient-based optimization methods to improve the performance.

## 5-Systems Engineering Challenges

The guiding tenet of our research is that future wireless ad-hoc networks will be designed using a combination of top-down and bottom-up strategies, and managed within a platform infrastructure.

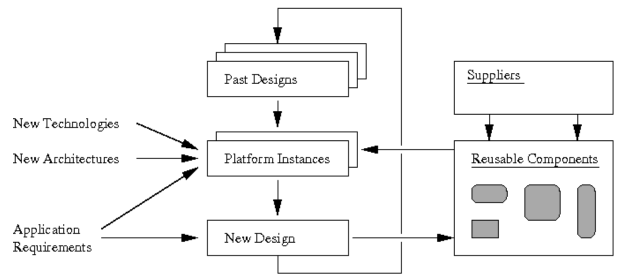


Figure 5. Flowchart for activities in platform-based design.

Figure 5 is a flowchart of activities for the incremental design and implementation of wireless networks based upon network assembly from reusable components provided by suppliers.. To realise this vision, methodologies and systems will need to overcome the following challenges:

**Challenge 5-1. Multi-Layer Organization of Requirements and Design Abstractions**. To simplify the treatment of design and management concerns, Requirements and design abstractions will be organized into a four-layer stack of abstractions – application, architecture, component, and implementation -- as shown in Figure 6.

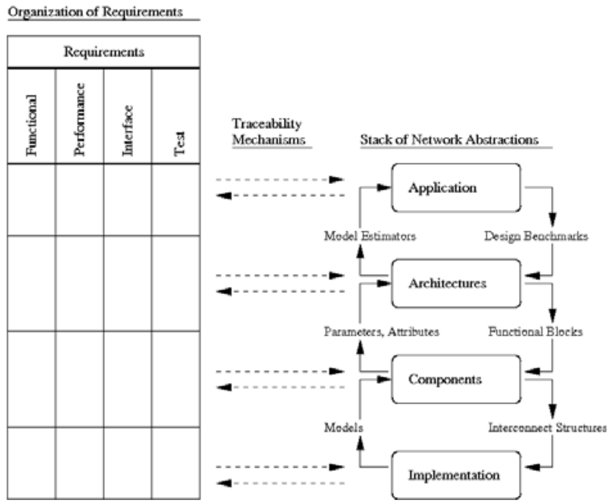


Figure 6. Stack of platform abstractions.

From the top, applications will place (design benchmarks) constraints on the design space of permissible network architectures which, in turn, will generate requirements on required functional blocks and constraints on acceptable interconnect structures (see Figure 2). From the bottom, implementation options provide modelling details, components provide details on implementation parameters and attributes, and architectures, details on network topology structures.

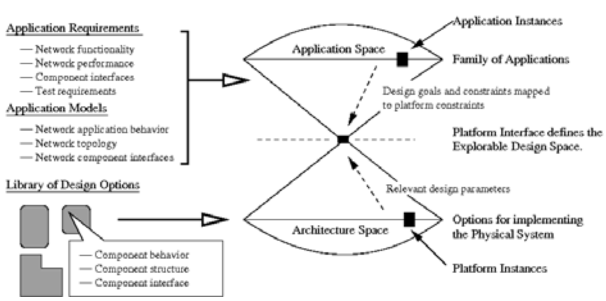


Figure 7. Merging of top-down and bottom-up strategies.

Therefore, as shown in Figure 7, we envision a design process where applications (i.e., descriptions of what is required) are represented by functional, performance, interface and test requirements, and network architectures (i.e., descriptions of potential design solutions) are defined by components and the details of their implementation. The design problem is further complicated by multiple system implementation opportunities assembled from mixtures of hardware and software.

A significant challenge is development of traceability mechanisms that can connect requirements to design abstractions. Present-day tools rely on the manual assembly of these relationships. Due to the scale and dynamic nature of these relationships, we will need tools that can automatically synthesise traceability relationships.

**Challenge 5-2. Superior Accuracy and Understanding Design through Separation of Concerns.** The goals for separation of concerns (SoC) are to pull a design apart and examine it from multiple perspectives.

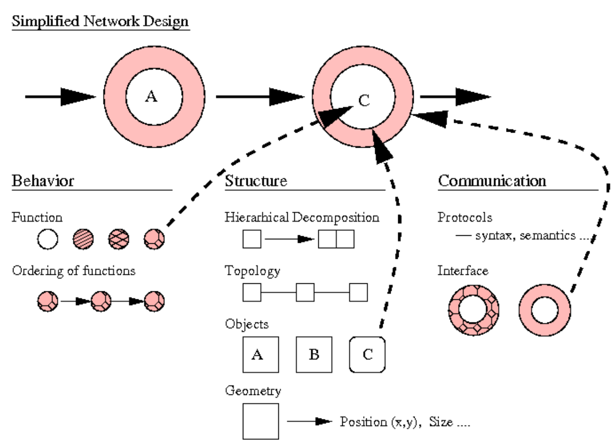


Figure 8. Separation of concerns for design of a component-enabled wireless network.

Figure 8 shows a simplified interpretation of SoC for a network design. A behaviour viewpoint emphasizes functions and their ordering. A topology or structure viewpoint emphasizes component abstractions, their decomposition and connectivity. A communications viewpoint focuses on protocols and the required interfaces for network node-to-node communication. A broader systems engineering perspective of SoC also includes separation of requirements and design abstractions (e.g., see Figure 6).

We surmise that SoC will play a central role in wireless ad-hoc network design and management. For example, network-node component design alternatives can be created by mapping behaviour models onto potentially good system structures. This is function-architecture co-design. When the SoC perspectives are independent (or almost independent), a second benefit of SoC is the opportunity for understanding design sensitivities by computing sensitivity of network performance due to perturbations in the parameters relevant to that perspective (e.g., component behaviour).

**Challenge 5-3. Formal Representations of Components and Interfaces**: Experience indicates that for each specific component, it is practically impossible to have a single design and realization that works well for all missions and networks. Therefore, two important challenges for component-based design are: (1) is development of formal representations for components and their interfaces, and (2) development for algorithms that can automate the selection of sets of components based on a matching of component capabilities against the mission requirements, specifications and network properties. One key challenge here is to define and specify the component interfaces such that any design that conform to the interface specifications can be used as a building block of a protocol with minimal effort. It is anticipated that under this framework, we will have a two-step design process. In the first step, we will select the combination of components that are suitable for the specific mission, and in the second step, we setup and tune the selected components parameters based on the mission requirements and objectives.

**Challenge 5-4. Support for Composable Component Properties and Behaviours.** Present-day component-based system design is a bottom-up approach that starts with low-level modules and sub-components and combines them into higher-level entities. Each team of component designers works independently and provides only the required interfaces. However, to keep difficulties in the analysis of component assemblies and network interactions in check, we note that the long term success of component-based design depends on two key conditions: **(1)** *Compositionality*: meaning that system-level properties (including behaviours) can be computed from local properties (behaviours) of components, and (2) *Composability* meaning that essential component properties do not change as a result of interactions with other components. Hence, perhaps the main challenge in the definition of components, is that their interfaces, and local properties should be done such that these two conditions are satisfied.

**Challenge 5-5. Methods and Tools for Real-time Network Management.** The final challenge is methods and tools for real-time network management. Network management should be provided with a high-level view of the network health, including trouble spots for network congestion and localised failure. A challenge is extension of present-day capability in sensitivity analysis to provide methods and tools for real-time computation of performance sensitivities with respect to the operational parameters, and trade-off analysis and decision making.

## References

Austin M.A., Information-Centric Systems Engineering: Class Notes for ENSE 623: Systems Engineering Validation and Verification, *Institute for Systems Research,* University of Maryland, College Park, MD 20742, Fall Semester 2009.

Baker D.J., J. E. Wieselthier J.E., and Ephremides A., “A distributed algorithm for scheduling the activation of links in self-organizing mobile radio networks," *IEEE Int. Conference Communications*, pages 2F6.1-2F6.5, 1982.

Baras J.S., Tabatabaee V., Papageorgiou G., Rentz N., “Modelling and Optimization for Multi-hop Wireless Networks Using Fixed Point and Automatic Differentiation”, accepted *to 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, WiOpt’08*.

Baras J.S., Perumal S., Tabatabaee V., Somasundaram K., Purkayastha P., “Loss Network Models and Multiple Metric Performance Sensitivity Analysis for Mobile Wireless Multi-hop Networks”, (Invited paper) *International Wireless Internet Conference (WICON 2008*).

Baras J.S., Tabatabaee V. , Purkayastha P., and Somasundaram K., “Componnent Based Performance Modelling of the Wireless Routing Protocols”, *IEEE ICC’09*.

Baras J.S., Tabatabaee V., and Jain K., “Component Based Modeling for Cross-layer Analysis of 802.11 MAC and OLSR Routing Protocols in Ad-hoc Networks”, IEEE MILCOM’09.

Clark D. and Tennenhouse D., “Architectural considerations for a new generation of protocols," *SIGCOMM*, 1990.

Kawadia V., P.R. Kumar, “A cautionary perspective on cross layer design," IEEE Wireless Communication Magazine, pp. 3{11, vol. 12, no.1, Feb 2005.

OLSR, Optimized Link State Routing Protocol. See: http://hipercom.inria.fr/olsr/ Accessed June, 2009.

Ousterhout J., ``Scripting: Higher Level Programming for the 21st Century", *IEEE Computer Magazine*, March. 1998.

## Biography

Mark Austin is an Associate Professor of Civil and Environmental Engineering at the University of Maryland, College Park, with a joint appointment in the Institute for Systems Research (ISR). Mark is Director of the Master of Science in Systems Engineering (MSSE) Program at ISR. Mark has a Bachelor of Civil Engineering (First Class Honours) from the University of Canterbury, Christchurch, New Zealand, and M.S. and Ph.D. degrees in Structural Engineering from the University of California, Berkeley.

Vahid Tabatabaee received Ph.D. in Electrical and Computer Engineering from University of Maryland at College Park in 2003. He is currently assistant research scientist in the Institute for Systems Research, University of Maryland at College Park. His research interests are in the design, analysis and modeling of wired and wireless communication networks, development of scheduling, routing and MAC algorithms and protocols for wireless networks, and on-line tuning and automatic fault detection of parallel computing systems and software applications.

John Baras is Professor of Electrical and Computer Engineering at the University of Maryland, College Park, with a joint appointment in the Institute for Systems Research. He was founding director of the Institute for Systems Research and currently holds the Lockheed Martin Chair of Systems Engineering. John has a B.S. in Electrical Eng. from the Nat. Technical University of Athens, Greece, 1970; M.S. and Ph.D. in Applied Mathematics from Harvard Univ. 1971, 1973.