Int. J. Appl. Math. Comput. Sci., 2008, Vol. 18, No. 3, 271–277
DOI: 10.2478/v10006-008-0025-8

NEW CHALLENGES IN DYNAMICAL SYSTEMS: THE NETWORKED CASE

PETER H. BAUER

Department of Electrical Engineering
University of Notre Dame, Notre Dame, IN 46556, USA
e-mail: pbauer@nd.edu

This paper describes new technical challenges that arise from networking dynamical systems. In particular, the paper takes a look at the underlying phenomena and the resulting modeling problems that arise in such systems. Special emphasis is placed on the problem of synchronization, since this problem has not received as much attention in the literature as the phenomena of packet drop, delays, etc. The paper then discusses challenges arising in prominent areas such as congestion control, sensor networks, as well as vehicle networks and swarms.

Keywords: networked systems, sensor networks, swarms, cogestion control, synchronization.

1. Introduction

Networked systems have become extremely popular over the last decade and are rapidly increasing their advance into different areas of technology. The use of networking to retrieve or send information is just the beginning in this embedded system revolution, and there is a strong trend towards connecting communication networks to the physical world through sensors and actuators. Ubiquitous sensing and actuation are one of the goals formulated by several researchers (and DARPA) over the last decade. This will eventually lead to drastic changes in how we interact with our environment.

In the most general way, networks have always played a special role in human history. Society itself through the relationships among people constitutes a network that exchanges information. Ancient trade roads that connected different cities of the globe are a network that moved “merchandise”. There are countless examples in biology, economy, sociology, etc. that show the trademarks of “networked systems” (Barabasi, 2003).

Modeling such networks is a non-trivial task and this paper will show some of the reasons for that by using high impact applications of networked systems to illustrate these challenges:

(a) *Congestion control*: Congestion control systems have been used for more than two decades to control and optimize the flow of information through a network of nodes. There are currently numerous research efforts ongoing to improve network performance by using radically new concepts and ideas. Congestion control networks are a very special case of “networked systems” due to the fact that practically all system nodes are “integrator plants”, i.e., data buffers. Congestion control problems have a number of other very special features that will be discussed later.

(b) *Sensor networks*: “Sensor networks” is a relatively new area of research that has become extremely popular over the last decade. It has a significant potential to change the world around us in almost every area and is still in its infancy. One may look at sensor networks of the future as an attempt to network our physical world so that physical information (temperature, light level, sound, etc.) anywhere can be accessed by anybody with a networked computing device. Ubiquitous sensing and global awareness are still far from reality, but we are currently making the first steps towards that goal.

(c) *Vehicle networks and swarms*: This area holds tremendous potential in many areas of everyday life. It is one of the most challenging instantiations of a “networked dynamical system” and shows all aspects of it: computing, communications, sensing, and actuation. Its applicability is as far ranging as micro robotic swarms that cure diseases in human bodies, automatic oil spill clean-up robotic agents that search, find, and remediate contaminated areas, battlefield swarms that search for targets and destroy them autonomously, and many more.

High fidelity models for these type of networked systems can become very complex and hard to analyze. On the other hand, there are many simplified models available that allow a simple linear time-invariant system analysis (Rohrs and Berry, 1997). However, due to model errors, these approximations do rarely result in accurate descriptions of system performance and are sometimes misleading.

This paper is structured as follows: In Section 2, the nature of networked systems will be explained and some simple deterministic models will be introduced. These models are capable of capturing most of the common parasitic networking effects such as time-variant (uncertain) delays of different types, packet drop, data rate fluctuations and synchronization errors. We will place additional emphasis on the area of synchronization, simply because it has not been investigated in great detail in the literature. Section 3 will illustrate the challenges in the network congestion control area in a qualitative fashion. In a similar way, Sections 4 and 5 discuss the areas of sensor and swarm networks, respectively. Special emphasis is placed on networks with simple low complexity nodes and local communication, since this is typical for current and probably also the next generation networks.

2. Nature of networked systems

In comparison with classical interconnected dynamical systems, networked systems introduce a number of parasitic effects. The most important ones are introduced by the communication link: delays, packet drop, link capacity, and link failures. Others are node induced and include quantization, node failure, and synchronization errors. We will discuss these effects one by one and provide some simple models when possible.

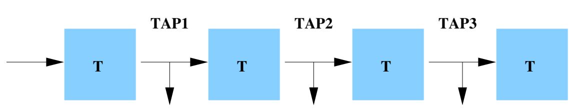

Fig. 1. Time variant delay model: tapped delay line.

2.1. Delays. Communication delays are caused by a variety of physical effects. Commonly one distinguishes between propagation delays, medium access delays, queuing delay, and processing delays. A simple but effective method to model these delays in a discrete time system environment is a tapped delay line with rules on the allowable dynamics of the tap position (Sichitiu *et al.*, 2003). Figure 1 illustrates a delay line with varying tap positions. In such a model reducing the delay is always connected with losing data samples, whereas an increase in the delay with time $\tau(n) = n$ leads to feeding the same data sample into the destination node. If source and destination nodes are discrete time systems with the same clock frequency, this model can produce any desired delay trajectories. Communication link-to-system interfaces can also be included in this model (Sichitiu, 2001) and can handle the problem that a discrete communication link model can produce more than one sample per time unit on the destination side. The rules on the tap position highly depend on the sample handling on the receiver side of the communication interface between the network and the system.

Other popular descriptions include stochastic delay models and piecewise constant delay models (Rayadurgam, 2004).

2.2. Packet drop. If the source node sends a packet to the destination node and this packet gets lost, then modeling the sample(s) being not available at the target node can be done in several ways that highly depend on the process the target node uses to make up for those lost packets. There are two commonly used schemes: holding the last valid received sample until a more recent sample arrives (time stamps assumed if it is not a FIFO structure) or generating zero value samples for the lost packets. The first choice can easily be achieved with the tapped delay line model in Fig. 1 (Sichitiu, 2001) by incrementing the tap position by one with every time instant, hence always using the same sample and feeding it to the target system as long as no more recent sample is available.

2.3. Link capacity. Link capacity limitations appear in two major versions: (a) communication link models that actually describe the volume of data that is transmitted through the network (e.g., in the forward path of a congestion control system), (b) communication link models that describe the characteristics of the very information that is transmitted through the network (e.g., a signal value that needs to be transmitted). In the first case, a capacity limit is simply modeled as a saturation nonlinearity, whereas in the second case it appears indirectly as an increasing delay or a packet drop (Sichitiu *et al.*, 2003). Arising from the second case are the tapped delay line as discussed above, whereas the delay models describing the volume of data moving through a network are very different and actually take the dual form (Bauer *et al.*, 2001) to those mentioned in Section 2.1.

2.4. Link and node failure. Temporary or permanent link failures can in the simplest case (global view) be modeled as "no connection", i.e., setting the path gain to zero. Another option is using a very long delay line with delays that increase linearly with time. This option is a good and simple choice for temporary failures. Short term link failures can also be described using stochastic models for packet drops. A node failure on the other hand is usually permanent and results in the removal of a node (and all its links) from the network. Methods that can capture these effects typically belong to the area of "fault tolerant systems" (Koren and Krishna, 2007).

2.5. Quantization. Quantization (Gersho and Gray, 1991) is an inherent property of all digital systems that have access to the analog quantities of the physical world. An extensive plethora of work describes a large number of quantization schemes, starting from simple fixed point scalar quantizers and stretching all the way to multidimensional quantization lattices and transform techniques that achieve efficient (source)coding. The problem of quantization is directly connected to the problem of rate limitation or link capacity, since data rate is the product of sampling rate and wordlength. Therefore increasing the sampling rate can only be achieved by reducing the wordlength, assuming the compression level is constant. Especially the problem of stabilizing data rates in networked feedback systems has attracted significant attention (Antsaklis and Tabuada, 2006) over the last several years. This work showed that in order to stabilize an unstable system through feedback, the minimum stabilizing rate in the feedback path is dependent on the logarithm of the sum of the unstable eigenvalues.

2.6. Synchronization errors. The area of synchronization error has attracted relatively little attention over the years even though it was shown decades ago (Kleptsyn *et al*., 1984a) that these errors can have drastic effects on the overall system behavior, e.g., leading to instabilities in an otherwise highly robust system (robust with respect to coefficient uncertainties). What is meant by the term "synchronization errors" is the error of the system response generated if different subsystems are not driven by the same clock anymore. Instead, one assumes that each subsystem (possibly a node on a network) has its own clock. Ideally the nodes have identical clock frequencies but in reality these clocks drift over time. Hence, the clock drifts that occur can generate switching patterns that drastically change the overall system behavior, even if the individual clock frequencies are very close (see Fig. 2).

The work (Kleptsyn *et al*., 1984b) showed that certain types of synchronous systems show stability robustness with respect to clock drifts of the different subsystems. For details on modeling, stability and performance of these systems see, (Lorand and Bauer, 2005; Lorand and Bauer, 2006a; Lorand and Bauer, 2006b).

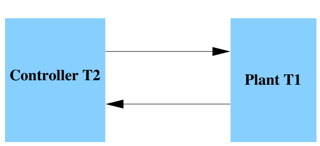

Fig. 2. Two network nodes with different clock periods.

3. Application 1: Congestion control

Congestion control mechanisms are one of the pillars of any modern digital communication network. A good congestion algorithm can drastically improve the throughput in a digital communication network, therefore leading to better network resource utilization (Fig. 3). There are many different congestion control algorithms (Rayadurgam, 2004) that have been designed and analyzed, but only a few are currently used in networks such as the Internet or high speed ATM networks.

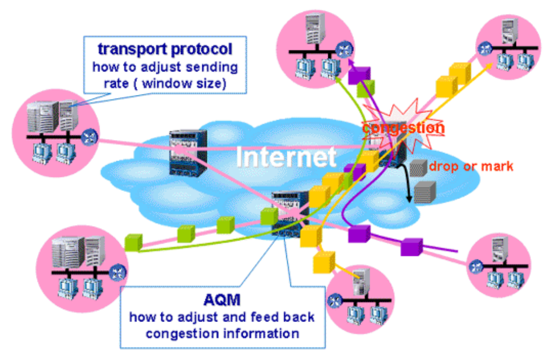

Fig. 3. Internet congestion control.

In essence, a congestion control algorithm varies the rate at which the source sends data to the destination based on certain properties (such as delay, arrival rate, order, etc.) of the data arriving at the destination. The destination then informs the source about these congestion indicators, which in turn adjusts its rate according to the congestion algorithm (see Fig. 4). Typical indicators are the buffer occupancy levels at the network switches (ATM networks), the arrival of packets out of order (TCP IP), missing packets, long delays, etc. One distinguishes two major classes of congestion algorithms: window based schemes (TCP) and rate based schemes (ATM).

In order to obtain a system dynamical description of this problem, the source, destination, and the network in between need to be modeled faithfully. This, of course, is an extremely difficult task, simply because the state of the network is usually not known. (Often, even the network structure is uncertain!) In the available bit rate option (ABR) in asynchronous transfer mode (ATM) networks, only the occupancy level of the most congested node is known (Sichitiu *et al*., 2003). In contrast to ATM, TCP IP based protocols (used for Internet congestion control) do not even have this type of information available. Therefore, coming up with an accurate dynamical system description is an extremely difficult problem. In order to illustrate the fundamental principles of congestion control, a simplified dynamical system model and its major components for the ATM network case (ABR option) are used. Figure 4 shows a single data source and a single destination and the most congested buffer. (In reality, this buffer changes over time.) In the forward path, data travels from the data sources through the time-variant delay blocks to the most congested buffer, which is essentially a rate integrator. Since this buffer physically sits at the most congested switch, data are taken from the buffer at the so-called depletion rate. The congestion controller computes these rates and sends them back through the return (feedback) path to the sources which in turn adjust their rates accordingly. Nonlinearities correspond to buffer occupancy nonlinearities (minimum and maximum buffer length) and rate nonlinearities (rates are bounded by the bandwidth and zero from above and below, respectively).

ATM networks provide very good performance, simply because the system allows measuring key network states. Control mechanisms are then relatively simple to design, even though things are complicated by time variance and nonlinearities. This is in stark contrast to TCP type protocols, where the level of congestion must be inferred by the so-called "indicators".

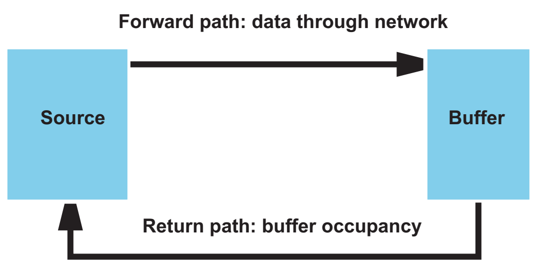

Fig. 4. Congestion control system model.

4. Application 2: Sensor networks

The field of "sensor networks" is another emerging front for dynamical system research. In the most general case, a sensor network consists of sensor-processor nodes that can communicate with a data sink and other sensor nodes. Each such node therefore consists of a processor, sensors, and a communication device. Sensor networks range from low complexity wireless sensor nets of a few nodes that communicate at a few kbits per second to video networks that communicate at rates of Mbits per second and above. Both wireless and wired networks are used for sensor nets. It is, however, wireless low complexity sensor networks that have attracted an immense amount of attention over the last several years. These networks are made up by very simple low cost sensor-processor nodes that communicate over short distances with each other using a digital radio. Due to the low cost of a single node, large networks of hundreds or even thousands of nodes are actually feasible and affordable. What makes the design of such networks challenging are the severe constraints that a single node usually is limited by: finite battery life, limited computational power, limited communication range, bandwidth and throughput, and limitations in sensor accuracy and performance. A number of low cost wireless platforms are commercially available (Crossbow, 2008; Easysen, 2008; Santilla, 2008) and used in real world applications.

Wireless sensor networks suffer from all the network induced effects introduced in Section 2, but link and node failure, packet drop, and synchronization errors are particularly important.

There are a number of applications and features in sensor networks that benefit from a dynamical system approach:

(a) node synchronization and synchronization errors,

(b) distributed system operations,

(c) feedback control in sensor networks.

Other potential problems such as resource allocation, energy aware routing, localization, etc. may also benefit from a system theoretic approach but are not further discussed here.

Practically all decentralized systems suffer from clock drifts of local nodes. This may not be a problem if data collection by a single sink node is the main focus and time resolution of events is not a critical issue. However, in many sensor network applications, data from many different sensor locations need to be correlated in order to pinpoint a particular event in space and time. This requires the notion of a common time, i.e., keeping individual clock drifts small is very important. Especially in distributed system operations in a network this feature is the key to a successful implementation. Imagine a sensor network implemented along a highway to detect certain traffic patterns, cf. Fig. 5. If this network is to implement (velocity) filtering algorithms, it is important that nodes be equi-distantly spaced and all nodes run synchronized. Only then can a centralized algorithm be mapped onto a sensor network in a distributed manner (Dewasurendra, Liang and Bauer, 2006). Especially in sensor actuator networks, introducing feedback over wireless links and closing the loop is a key feature. In this case, synchronization issues are even more pronounced (Li and Rus, 2004). The topic of mobile sensor networks and swarms is a special case of an sensor-actuator network, and this topic will be discussed in greater detail in Section 5.

It has been shown (Dewasurendra, Liang and Bauer, 2006) that classical m-D systems can be implemented in a distributed fashion on a grid sensor network. In such an application of sensor networks, each sensor node senses and processes the data locally according to a local state space model (typically the FM model) that describes the system. However, limitations of each node now become very critical constraints for the execution of any m-D filtering operation. A simple example of a 2-D velocity filter implemented on a regular 1-D grid of sensor nodes is shown in Fig. 5.

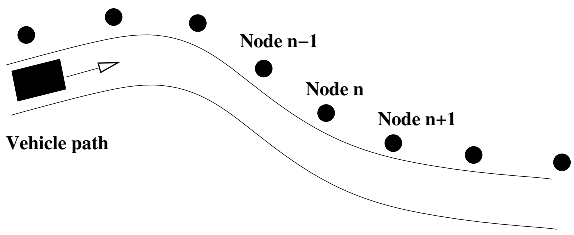

Fig. 5. A 1-D sensor network for velocity filtering.

Now consider the 2-D FM model running in each of the nodes:

$$\begin{aligned} x(n_1+1,n_2+1) &= A_1x(n_1,n_2)+A_2x(n_1+1,n_2) \\ &\quad + A_3x(n_1,n_2+1)+Bu(n_1,n_2), \\ y(n_1,n_2) &= Cx(n_1,n_2)+Du(n_1,n_2). \end{aligned}$$

A 2-D velocity filter would then take the form of the above equation, where n_1 denotes the node number and n_2 equals discrete time. With finite communication speed, we have $A_3 = 0$.

5. Application 3: Vehicle networks and swarms

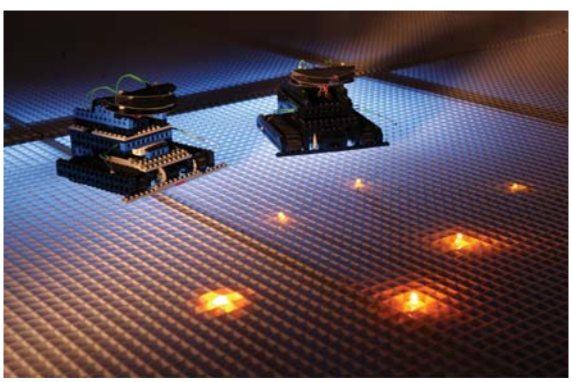

Fig. 6. Two low complexity swarm agents.

Vehicle and mobile sensor networks have received tremendous attention in the literature over the last decade. This is due to the many foreseeable applications in military and civilian tasks. Mobile networks are the prime example of a sensor-actuator network with all the possible difficulties and complications that arise when feedback loops are closed over wireless connections. There are many different classes of mobile networks ranging from high complexity vehicle networks (equipped with high end sensory hardware such as a radar, laser range finders, GPS, etc.) to low complexity swarms, where each swarm agent is extremely simple and is comparable to the low end sensor nodes described in the previous section albeit with the ability to move, cf. Fig. 6. We will mainly focus on this second type of network.

Reynold's three rules (Reynolds, 1987) of swarming basically come from biology and describe local agent interactions: flock centering, obstacle avoidance and velocity matching. An artificial agent swarm therefore makes decisions based on local interactions with its neighbors and there is no centralized node or processing center that makes decisions for the swarm in any way or form. Therefore processing is done in a distributed fashion usually using emergent behavior: simple interaction rules that are identical for each agent result in overall swarm behavior that is self-organizing and useful for problem solving. Emergent behavior occurs only when a large number of agents work together and is usually not observed if the number of agents is small.

It is now clear that each swarm agent must make navigation decisions in real time based on the information it obtains from its direct neighbors, either through direct sensing (measurements) or through communicated measurements from other neighboring agents. Some of the lowest complexity versions of agent swarms do not even possess memory and are purely reactive (Scheutz *et al.*, 2005). Even in this case, a number of important tasks can be solved (Dewasurendra, Bauer, Scheutz and Premaratne, 2006; Scheutz and Bauer, 2006): high value target protection, substance detection, tracking, formation flight, etc.

In the particular case of swarming, the effects of packet drop, time-variant communication delays, and synchronization are especially obvious since deviation from the desired agent trajectory becomes apparent very easily. Node failure can result in collisions and even agent loss, whereas packet loss, long delays, and synchronization errors lead to deteriorating performance and possibly also the loss of stability. (It should be mentioned that in large low complexity self organizing swarms, agent loss is usually acceptable even if that loss rate is substantial.)

Figure 7 shows an ultra-low complexity vehicle swarm from the University of Notre Dame's MOSES lab. It consists of VEX agents equipped with two wireless nodes (TelosB or Tmotes) and a simple SBT30 Easysen sensor board. The VEX platform is a four-wheel platform with a processor that handles tactile sensors and wheel control. Radio beacon induced potential fields are used for the navigation of all agents: fixed waypoints generate a "global" attractive field, whereas each agent generates a local "repellive" field to avoid collisions. Most approaches use artificial potential fields, rather than real beacon induced fields. In order to accommodate repelling behavior locally around other agents and global attractive behavior towards waypoints, the governing agent equations are necessarily nonlinear.

Agents operating on a plane typically take the follo-

Fig. 7. Ultra-low complexity vehicle swarm.

wing analytical form:

$$\underline{\dot{x}_i}(t) = A(\underline{x_i}(t) - \underline{u_i}(t)) + \sum_{j=1;j\neq i}^{N} f_i(\underline{x_i}(t), \underline{x_j}(t)),$$

$i = 1,\ldots,N$, where $\underline{x_i}$ is the location of agent i in the 2-D plane, i.e., $\underline{x_i} \in \mathbb{R}^2$. A is a stable Hurwitz matrix with eigenvalues in the left half plane, $\underline{u_i}(t)$ is an external input (equilibrium point or target location), and $f_i(\underline{x_i}(t), \underline{x_j}(t))$ is a repelling term that is non-zero only if $\underline{x_i}(t)$ and $\underline{x_j}(t)$ satisfy $\|\underline{x_i}(t) - \underline{x_j}(t)\| < \epsilon$. This term ensures collision avoidance. In reality, the above equations are implemented in discrete time units slightly different clock cycles leading to synchronization errors.

6. Conclusion

This paper summarizes new system dynamical challenges in areas at the intersection of networks, systems and control. In particular, congestion control, sensor networks, and swarm networks are singled out to illustrate networked system phenomenas such as synchronization errors, time-variant delays, capacity limitations, packet drop, etc. Emphasis is placed on the role of new high fidelity system and interconnection models that are sufficiently simple to allow for accurate system performance predictions.

Acknowledgment

This work was supported by the Defense Threat Reduction Agency grant DTRA N00164-07-C-8570. The author would like to thank DTRA and CRANE for their support.

References

Antsaklis P. and Tabuada P. (2006). *Networked Embedded Sensing and Control*, University of Notre Dame, Notre Dame, IN.

Barabasi A.-L. (2003). *Linked*, Penguin Group, London.

Bauer P., Sichitiu M. and Premaratne K. (2001). On the nature of time-variant communication delays, *Proceedings of the 19th IAESTED Conference on Modeling, Identification, and Control*, Innsbruck, Austria, pp. 792–797.

Crossbow (2008). Available at: http://www.xbow.com.

Dewasurendra D., Bauer P., Scheutz M. and Premaratne K. (2006). Evidence based navigation in swarms, *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, USA (on CD-ROM).

Dewasurendra D., Liang Y. and Bauer P. (2006). Distributed multi-dimensional evidence filtering, *ICYR Conference 2006*, Zielona Góra, Poland, (unpublished)

Easysen (2008). Available at: http://www.easysen.com.

Gersho A. and Gray R. (1991). *Vector Quantization and Signal Compression*, Kluwer, Boston, MA.

Kleptsyn A. F., Kozyakin V., Krasnoleskii M. and Kuznetsov N. (1984a). Effects of small synchronization errors in stability of complex systems ii, *Avtomatika i Telemekhanika,* translated in: *Automation and Remote Control* **44**(7): 309–314.

Kleptsyn A. F., Kozyakin V., Krasnoleskii M. and Kuznetsov N. (1984b). Effects of small synchronization errors ins stability of complex systems iii, *Avtomatika i Telemekhanika,* translated in: *Automation and Remote Control* **45**(3): 1014–1018.

Koren I. and Krishna C. (2007). *Fault Tolerant Systems*, Morgan Kaufmann, New York, NY.

Li Q. and Rus D. (2004). Global clock synchronization in sensor networks, *Proceedings of the IEEE Conference Infocom 2004*, Hong Kong, pp. 574–575.

Lorand C. and Bauer P. (2005). Distributed discrete time systems with synchronization errors: Models and stability, *IEEE Transactions on Circuits and Systems II: Express Briefs* **52**(4): 208–213.

Lorand C. and Bauer P. (2006a). Clock synchronization errors in circuits: Models, stability and fault detection, *IEEE Transactions on Circuits and Systems I: Regular Papers* **53**(10): 2299-2305.

Lorand C. and Bauer P. (2006b). On synchronization errors in networked feedback systems, *IEEE Transactions on Circuits and Systems, CAS I: Regular Papers* **53**(10): 2300–2317.

Rayadurgam S. (2004). *The Mathematics of Internet Congestion Control*, Birkhäuser, Berlin.

Reynolds C. W. (1987). Flocks, herds and schools: A distributed behavioral model, *Proceedings of the Conference on Computer Graphics ACM SIGGRAPH 87*, Anaheim, CA, USA, Vol. 21, pp. 25–34.

Rohrs C. E. and Berry R. A. (1997). A linear control approach in explicite rate feedback in ATM networks, *Proceedings of the IEEE Conference InfoCom*, Kobe, Japan, pp. 227–282.

Scheutz M. and Bauer P. (2006). A scalable, robust, ultra-low complexity agent swarm for area coverage and interception tasks, *Proceedings of International Symposium on Intelligent Control*, Munich, Germany, pp. 1258–1263.

Scheutz M., Schermerhorn P. and Bauer P. (2005). The utility of heterogeneous swarms of simple UAVs with limited sensory capacity in detection and tracking tasks, *Proceedings of the IEEE Swarm Intelligence Symposium*, San Diego, CA, USA, pp. 257–264.

Sentilla (2008). Available at: http://www.sentilla.com.

Sichitiu M. (2001). *Control of data networks: Models, stability and Controllers*, Ph. D. thesis, University of Notre Dame, Notre Dame, IN.

Sichitiu M., Bauer P. and Premaratne K. (2003). The effect of time-variant uncertain delays in ATM networks with explicite rate feedback: A control theoretic approach, *IEEE/ACM Transactions on Networking* **11**(4): 628–637.

Received: 18 March 2008