Network-Adaptive Pervasive Applications

Vincenzo Liberatore*

Introduction. *Pervasive computing* enables us to monitor and affect local and remote physical environments (Figure 1). However, the network can introduce unreliable and time-dependent levels of service in terms of, for example, delays, jitter, or losses. Quality-of-Service (QoS) can ameliorate the real-time network behavior, but the network behavior is still subject to interference (especially in wireless media), to routing transients, and to aggressive flows. In turn, network vagaries can jeopardize the stability, safety, and performance of the units in a physical environment. The primary objective of our research is to devise integrated control and communication algorithms to compensate for the vagaries of network service. Such strategies are targeted toward the application-layer and their objective is to deal with packet losses, delays, jitter, and network unreliability.

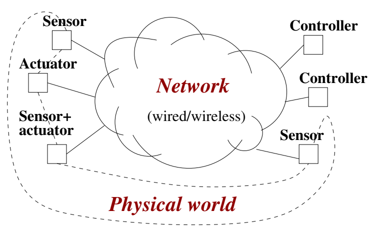
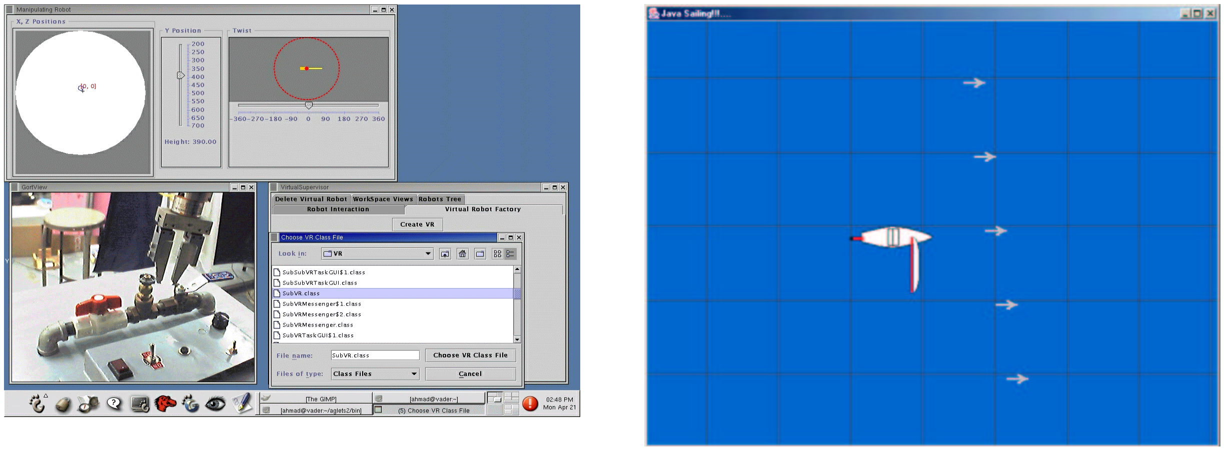


Figure 1: The *pervasive computing* vision.

A General Framework. Our general contention is that network-adaptive applications necessitate an integrated approach that combines networking (e.g., network measurement, modeling, and simulation; reliability) with decisions based on sensor data (e.g., feedback control). Furthermore, it is our conviction that although strategies have to be adapted to the specific application areas, it is possible to develop a general methodology by drawing from the foundations of Systems Theory and of Networking [2]. Our methodology is unified in two respects. First, it relies solely on a system-theoretic description of the environment where sensors and actuators are deployed, and thus it is general and encompasses specific applications as particular cases. Second, it strives to optimize directly system-related metrics, such as stability or performance. These metrics are influenced by packet losses, delays, and jitter but do not coincide with any one of these network-oriented metrics. As a result, levels of service are expressed in application-related metrics, as opposed to packet-related metrics, and at the same time the system-theoretical approach allows any specific application to be viewed as a parameter in the system description.

*Division of Computer Science, Case Western Reserve University, 10900 Euclid Avenue, Cleveland, Ohio 44106-7071, USA. Fax: (216) 368 6039. E-mail: `vincenzo.liberatore@cwru.edu`. URL: `http://vincenzo.liberatore.org/`. This work has been supported in part under NSF grant ANI-0123929, NASA grants NAG3-2578, NAG3-2799, and a Microsoft unrestricted gift.



(a) Internet robotics (b) Distributed simulation

Figure 2: Network aware pervasive applications range from Internet-enabled robotic manipulation (left) to network simulations and gaming (right).

Internet Robotics. The broad vision of Internet robotics is to enable end-users to affect a remote physical environment though network connectivity and action units. Applications range from distributed instrumentation [1] to home robotics [3]. Our approach is based on software components (*agents*) that can move among hosts in response to changes in the robotic task and in network conditions. Figure 2(a) shows a current prototype: the window on the bottom left corner is a live WebCam view of the robot work space. The top window enables an end-user to directly tele-operate the robot through mouse movements a keyboard strokes. The bottom right windows enable the creation of new mobile agents by directly instantiating the definitions contained in appropriate Java class files. This GUI also allows the logical interconnection of such agents in a hierarchical structure.

Distributed Simulations. The environment can be located in the physical world, but it can also be numerically simulated. The result is a distributed interactive simulation, with applications ranging from medical training to networked videogames. Figure 2(b) shows one such application: a sailing simulator. A "player" can set the tack angle and boom position, which in turn set the direction and speed of the boat as a function of the wind speed and direction and of water currents. More challenging applications include complex physical scenarios with non-linear dynamics, such as collisions, distributed interactive simulations, and physics-based groupware. The simulation application should compensate for the vagaries of network levels of service. Compensation includes traditional techiniques such as dead-reckoning, as well as novel methods based on network characterization and on control-theoretical stability theorems [2].

References

[1] A. Al-Hammouri, A. Covitch, D. Rosas, M. Kose, W. S. Newman, and V. Liberatore. Compliant control and software agents for internet robotics. In *Eighth IEEE International Workshop on Object-oriented Real-time Dependable Systems (WORDS)*, 2003.

[2] M. S. Branicky, V. Liberatore, and S. Phillips. Co-simulation for co-design of networked control systems. In *American Control Conference*, 2003.

[3] Marco L. Ngai, Vincenzo Liberatore, and Wyatt S. Newman. An experiment in remote robotics. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2190–2195, 2002.