

# On Static WCET Analysis vs. Run-time Monitoring of Execution Time

*Charles D. Cavanaugh*  
*The Center for Advanced Computer Studies*  
*University of Louisiana at Lafayette*  
*cdc@cacs.louisiana.edu*

Dynamic, distributed, real-time control systems control a widely varying environment, are made up of application programs that are dispersed among loosely-coupled computers, and must control the environment in a timely manner. There are four variables to consider in a real-time control system: the environment, system load, resources required, and timing. The environment, which is the independent variable, affects the system load. The system load affects the system's processing time; the greater the system load is, the longer the processing time is. More resources are needed to handle a relatively high system load, and fewer resources are needed to handle a relatively low system load if the processing time is to be controlled. The motivating example application is an anti-air defense system operating in the environment of a battleship. The number of potential threats, such as airplanes or missiles, in range of the battleship can vary from none to a potentially large number. Consequently, the system load varies from light to heavy. The environment determines the number of threats; thus, it is difficult to determine the range of the workload at design time. While a system is lightly loaded, it is wasteful to reserve resources for the heaviest load. Likewise, it is also possible that the load will increase higher than the assumed worst case. A system that has a preset number of resources reserved to it is no longer guaranteed to meet its deadlines under such conditions. Other applications include robotics. In order to ensure that such applications meet their real-time requirements, a mechanism is required to monitor and maintain the real-time quality of service (QoS): a QoS manager. The QoS manager should maintain the system's ability to meet its deadlines under dynamically varying loads. The QoS manager monitors the processing timing (latency) and resource usage of a distributed real-time system, forecasts, detects and diagnoses violations of the timing constraints, and requests more or fewer resources to maintain the desired timing characteristics. In particular, the system consists of real-time paths, collections of time-constrained and precedence-constrained applications that may be distributed across multiple, heterogeneous computers and networks. The QoS manager uses the concept of a path by monitoring the end-to-end processing timing from the start of the path to the end of the path. To enable better control over the system, The goals are as follows: 1) Gather detailed information about anti-air warfare and air-traffic control application domains and employ it in the creation of a distributed real-time sensing and visualization testbed for air-traffic control. 2) Identify mathematical relationships among independent and dependent variables, such as performance and fault tolerance vs. resource usage, and security vs. performance. 3) Uncover new techniques for ensuring performance, fault tolerance, and security by optimizing the variables under the constraints of resource availability and user requirements.