## Multicast Routing using Delay Intervals for Collaborative and Competitive Applications

Andrus, J.<sup>1</sup>, Banik, S.<sup>2</sup>, Swart, B.<sup>3</sup>, and Verdicchio, M<sup>4</sup>

<sup>1</sup>Undergraduate Student, Department of Mathematics and Computer Science, The Citadel, <sup>2</sup>Associate Professor, Department of Mathematics and Computer Science, The Citadel, <sup>3</sup>Assistant Professor, Department of Mathematics and Computer Science, The Citadel, <sup>4</sup>Associate Professor, Department of Mathematics and Computer Science, The Citadel baniks1@citadel.edu

**Keywords:** QoS, Graph Algorithm, Multicast Routing, Delay Interval, Collaborative Applications

**Abstract:** Collaborative and competitive applications require that participants receive messages almost simultaneously and before a specified time. These requirements have been addressed by the Delay Variation Bounded Multicasting Tree (DVBMT) problem. In our research, we propose the Interval Multicast Subgraph problem (IMS) to address these requirements. IMS addresses these constraints with an interval of acceptable delay values for paths as user input from source to a destination, eliminating the need to optimize delays for a variation value. By solving IMS rather than DVBMT and other variants of DVBMT, we are able to find solutions for larger graphs more efficiently. Our proposed Interval Multicast Algorithm (IMA) accounts for an interval of acceptable delay as user input and guarantees the weight of each path from the source to a distinct destination is within the given interval, if that path exists. We provide proofs of correctness and complexity of IMA, as well as simulation experiments, to illustrate the effects of various parameters on our algorithm. Simulations show that IMS is significantly less costly than finding the minimum variation for the average and best case. By remodeling the DVBMT problem to IMS, we have created a new problem that addresses the Quality of Service (QoS) requirements of multicasting and is able to be solved efficiently for the average case for relatively large graphs.