Implicit Flow QoS Signaling Using Semantic-Rich Context Tags

Roel Ocampo[1,2], Alex Galis[2], Hermann De Meer[3], and Chris Todd[2]

[1] Department of Electrical and Electronics Engineering, University of the Philippines, Diliman, Quezon City, 1101 Philippines
[2] Department of Electronic and Electrical Engineering, University College London, Torrington Place, London WC1E 7JE, U.K.
[3] Faculty of Mathematics and Computer Science, University of Passau, 94032 Passau, Germany

Abstract. An important feature of future context-aware and adaptive networks would be the ability to provide QoS to user flows. Our approach enables end-hosts and other devices to expose and provide context information to the network to support underlying QoS mechanisms, including adaptation. We discuss the key elements of our approach and demonstrate its use in an experimental scenario.

1 Introduction

Current proposals for providing QoS in networks often expect end-hosts to either explicitly signal their requirements and undertake resource reservation, or for them to have sufficient knowledge about the underlying QoS model in order to map application flows to existing QoS classes. While QoS signaling messages may contain detailed description of QoS characteristics and requirements of a flow [1], they are often not sufficient to paint a "big picture" that better describes the desired interaction between the user and the network.

In this paper we briefly describe some aspects of our on-going work, which explores the application of concepts and techniques in *context awareness* to networks, particularly through the use of context tags that describe flow context. Although we envision a wide range of applications for context tags, we will focus here on their possible use in implicitly signaling the QoS characteristics and requirements of network flows.

2 The Context of a Flow

Our definition of flow context is derived from the domain of pervasive and ubiquitous computing [2]. We define the context of a network flow as any information that can be used to characterize its situation, including information pertaining to other entities and circumstances that give rise to or accompany its generation at the source, affect its transmission through the network, and influence its use at its destination. This includes not only the intrinsic, low-level characteristics of a flow, but also the nature of the applications, devices, and the activities, intentions, preferences and identities of the users that produce or consume the flow.

From a QoS perspective, flow context may be used in the following ways: (1) to decouple end-hosts and applications from the underlying domain-specific QoS model, (2) to provide or expose additional information about the flow to the network in an explicit way to facilitate flow classification for QoS purposes, (3) to trigger QoS adaptation directly on the flow, and (4) to identify and label suspicious and malicious flows, or those that are in violation of QoS contracts.

3 Tagging Flows with Context

Our approach consists primarily of tagging network flows with context information. The following are the key elements of this approach:

1. *Architecture.* Context sensing is performed in a distributed fashion, at end-hosts and network devices such as middleboxes. Context tags are then assembled and injected along the path of the flow and are intercepted and processed by devices downstream. This may lead to a control or management action, a service, or an adaptation function being triggered within an attached forwarding device such as a router. End-hosts may also process context tags.
2. *Tag structure.* Tags are formatted using Extensible Markup Language (XML) [3] and transported within UDP datagrams. The IP packet header contains the IP Router Alert Option as described in RFC 2113 and RFC 2711 [4].
3. *Tag aggregation.* Tag processing also results in the aggregation of information coming from multiple tags accumulated over time, or from multiple flows, resulting in higher-level context information that provides a more complete contextual description of a single flow or a flow aggregate (macroflow).
4. *Flow context ontology.* Declarative semantics within an ontology encode contextual relationships and properties, and facilitate the use of reasoning within the tag aggregation process. They likewise provide a means by which QoS characteristics and requirements may be inferred from context.

4 Experimental Scenario

Figure 1 illustrates the use of context tags for QoS adaptation in a simple experimental scenario. (A) A user, initially allocated 500 kbps, views a video stream. (B) The context tag within the stream results in a new bandwidth allocation of 1.5 Mbps, allowing the video stream to rise to its characteristic level of around 850 kbps. (C) The user requests an additional video stream with a higher priority. (D) The combined traffic saturates the bandwidth allocation, resulting in degraded video for both streams. (E) Receipt of the new context tag and aggregation allows the streams to be prioritized, and the network further adapts by applying transcoding on the lower-priority stream. (F) The lower priority stream now operates at a lower average bitrate after transcoding, while the higher priority stream occupies its natural traffic level (G). (H) The total bandwidth consumed stays well within the 1.5 Mbps allocation.

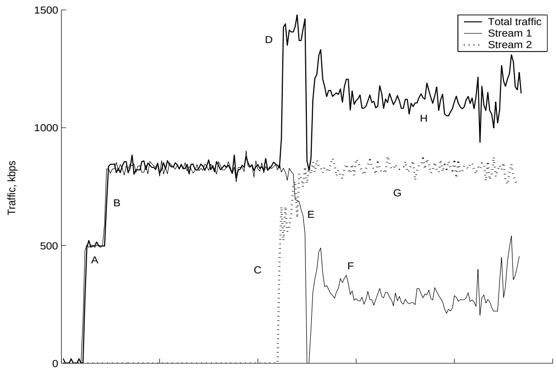

Fig. 1. Inbound traffic on end-host

5 Conclusion and Future Work

The ability to provide QoS to flows in a manner that decouples end-hosts from the underlying QoS model is an important aspect of context-aware and adaptive networks. We have demonstrated using a simple experimental scenario how context tags may implicitly signal QoS requirements. Work is ongoing on the further development of the flow context ontology and its dynamic linkage with the tag processing and aggregation component of our architecture.

Acknowledgment. This paper describes work partially undertaken in the context of the E-NEXT - IST FP6-506869 project, which is partially funded by the Commission of the European Union. The views contained herein are those of the authors and do not necessarily represent those of the E-NEXT project. Roel Ocampo acknowledges support from the Doctoral Fellowship Program of the University of the Philippines.

References

1. L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: A New Resource ReSerVation Protocol. *IEEE Network*, September 1993.
2. A. K. Dey, D. Salber, and G. D. Abowd. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction (HCI) Journal*, 16 (2-4), 2001
3. T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler, and F. Yergeau (editors). Extensible Markup Language 1.0 (Third Edition). *W3C Recommendation 04 February 2004*, http://www.w3.org/TR/REC-xml
4. D. Katz. IP Router Alert Option. *Request for Comments 2113*, February 1997.