The MATRIX: A QoS Framework for Streaming in Heterogeneous Systems

Larisa Rizvanovic
Department of Computer Engineering
Mälardalen University
P.O.Box 883
S-721 23 Västerås, SWEDEN
larisa.rizvanovic@mdh.se

Gerhard Fohler
Department of Computer Engineering
Mälardalen University
P.O.Box 883
S-721 23 Västerås, SWEDEN
gerhard.fohler@mdh.se

Abstract

The work presented here was carried out within the FABRIC project[1], which aimed at the integration of middleware standards used in home networks to provide high quality streaming over heterogeneous networks.

In this paper we propose an adaptive QoS framework for efficient resource management, called the Matrix approach. The Matrix is a concept to abstract from having detailed technical data at the middleware interface. In stead of having technical data referring to QoS parameters like: bandwidth, latency and delay we only have discrete portions that refer to levels of quality. The underlying middleware must interpret these values and map them on technical relevant QoS parameters.

1 Introduction and Rationale

The aim of the FABRIC project was to develop *an architecture* in which several interoperability standards and technologies in the home networking context can be integrated. In addition, a FABRIC handles the complete network to satisfy *End-to-End Quality of service (QoS)* requirements.

Efficient transport of streams with acceptable playout quality in the FABRIC network requires management of both networks and CPUs. As resources will be limited in the home environment, *guarantee* mechanisms for continuous stream transport are demanded, e.g., it will not be acceptable to interrupt a football game for the start up of another stream.

One of the key issues for resource management is an efficient representation of the fluctuating system state and resource allocation decisions, to provide a small interface to decouple device scheduling and system resource allocation.

Event based vs. controlled state update

The overhead to transport the information needed for a 100% accurate view of the system with very fine grain granularity capturing highly fluctuating resources such as wireless networks will be prohibitively high on the network; scheduling activities for all events will overload CPUs. In addition, such information would be too fine-grained fluctuating, as resource management has to operate at larger granularity. We believe that the tradeoffs between accuracy of system state information and efforts to transport and process have to focus on efficiency providing the minimum relevant information for resource management only.

Step-by-step vs. system resource management

A distributed resource management approach will be severely impeded by overheads, as devices have to exchange information to determine a global system view for scheduling decisions. In addition, scenarios such as high fluctuations on a network link demand more scheduling activities, which in turn will create more network overhead, resulting in increased fluctuations.

A step-by-step approach, in which decisions about how much of a resource is dedicated to a given version of a stream are taken locally by the devices in sequence overcomes some of the problems of the fully distributed approach. It is impeded, however, by several shortcomings as well: the decisions taken on each device suffer from the limited view of the state of the device and the next one on the route. Suppose the resource on both devices can provide ample availability at the moment, resulting in the choice of a high quality/high bandwidth version of the stream to be transmitted. If any of the devices on the route to the play-out can handle only a lower quality version, the resources used here for the high quality will be wasted. Transcoding complicates the issue and is not considered. Propagating the state information back and forth along the route results in the overhead given earlier and delays the actual scheduling in each device further. Rather, a sender-based approach with global knowledge is appropriate.

[1] FABRIC is the European IST project IST-2001-37167.

Interfacing device – system resource management

Decoupling communication and synchronization between devices and resource management enables operation without explicit consideration of intricate details of device schedulers. By providing an abstraction level suitable for resource management, it facilitates a component-based approach as well. Instead of the resource manager probing local schedulers for state information, the devices can provide information about relevant state information and estimations about changes with appropriate, individual granularity themselves. Failures in devices or communication will not block or delay resource management.

Chosen design

We propose a global abstraction of device states as representation of the system state for resource management and as interface to decouple device scheduling and system resource allocation.

This global abstraction, called Matrix contains information about device states in a format appropriate for resource management. The accuracy of the information represented is suitable for resource management, abstracting over fluctuations or changes, which will overload scheduling: the very fine grain resolution of values is mapped into a very small number or discrete values. Devices are responsible for providing information about their states, only for changes relevant for resource management, i.e. in the pre-processed reduce value range. Thus the overhead to keep system wide state information fresh is dramatically reduced and no explicit communication or synchronization between resource management and local schedulers is needed. Devices update relevant information at the appropriate pace to the Matrix, on which resource management bases decisions.

Diffusion of these decisions to devices is carried out via the Matrix as well, i.e., orders for resource allocation on individual devices is put in high level abstractions of limited value ranges into the Matrix, from where the devices pick up orders to translate them into local scheduling policies or parameters. Thus global resource management is independent of detailed knowledge about local schedulers, which can be replaced easily, supporting a component based approach.

Without the need for explicit costly communication and negotiation between devices, decision about which version of streams to transport or the decomposition of end-to-end delays can be performed by global resource management, reducing overheads and resource waste due to limited local device knowledge.

The Matrix provides a logical abstraction of the view of the system state, not an actual centralized implementation requirement. Rather, the Matrix is represented in a distributed way. Not necessarily do all devices in the system have to use the Matrix, but a mix of direct explicit communication and Matrix abstraction is conceivable for resource management, although benefits of the data abstraction would obviously be reduced.

2 Operation

The Matrix enforces complex stream quality demands to be translated into a few resource interfaces.

QoS levels

The information in Matrix is prepared, ie. largely reduced in range, suitable for resource management, abstracting over fluctuations or changes, which will overload scheduling. Thus, the Matrix will contain only a very small, discrete range of few Quality of Service (QoS) performance levels, for example High, Medium and Low (see Figure 1).

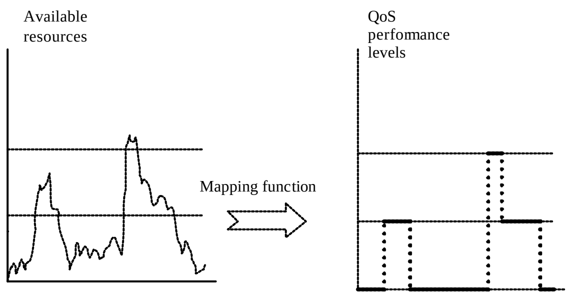

Figure 1: Mapping various kind of traffic specification to few QoS levels

It is crucial to map various kinds of traffic specifications to these few QoS levels in a manner that gives a fair abstraction of the system while reducing overheads. We assumed a linear mapping between bit rate and quality of a stream as the complex relationship between user perceived quality and resource demands was beyond the scope of the project. However, a comprehensive QoS mapping mechanism that translates representation of QoS at different system layers (i.e., application, network) and also considers user perceived quality [6, 7, 8] is a task of future work.

The fact that the Matrix approach has to present a valid picture of resources could involve a need for constant updates coming from the bandwidth fluctuations. But in our opinion, only two occurrences will cause the update of the Matrix elements:

- A new connection (a subscription by render device to a certain video content)
- A significant change of available resources in the system (e.g. when a change to a different quality level occurs for a significant amount of time).

In the Matrix, information about available local resources will be used, by the "resource manager". The resource manager will try to achieve an optimal (feasible) scheduling solution for all streams in the system, with respect to their priorities. Then, this global view of resources, end-to-end time constraints, will be transformed to a few scheduling factors for each link in the path of streams. These decisions, expressed in a very small number of discrete values, are spread to devices by the Matrix as well, from where the devices pick up orders to translate them into local scheduling policies or parameters.

3 Architectural design aspects

This section introduces a more detailed overview of the Matrix approach. The Matrix is composed of several entities that constitute an effective mechanism for monitoring and scheduling available resources in the system. Figure 2 shows the data flow (information flow) between the Matrix components. The functions of these components are further discussed in the following subsections.

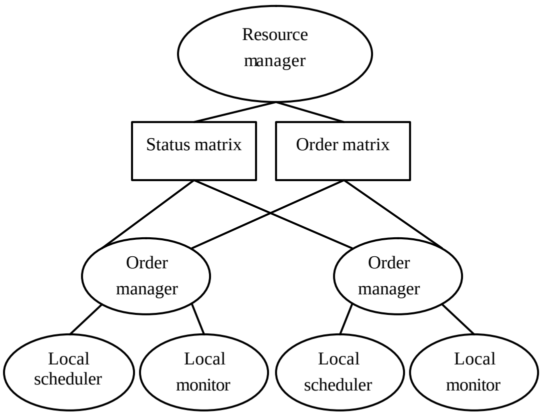

Figure 2: Information flow between Matrix's components

3.1 Resource manager

The resource manager will be used to schedule and reserve resources, within system. One part of stream scheduling is providing end-to-end timing constraints. So, the resource manager will provide sub deadlines (sub delays) for each device in the system by real-time methods such as outlined in [2].

Each time a new connection is about to be made, the resource manager has to determine if sufficient resources are available to satisfy the desired QoS of the new connection, without violating QoS of existing connections. Likewise, each time a resource variation occurs, that affects an active video stream, the resource manager has to make an adjustment of streams and resources.

In order to deal with resource reservation, the resource manager has to have knowledge about currently available resources in the system. This information is obtained from the Status Matrix. If there are enough resources to support the requested connection, the resource manager puts orders for resource reservation into the Order Matrix. Orders can be seen as an interface between the resource manager's global view of resources and set of entities (order manager, local scheduler and local monitor), which called "local enforcement mechanism".

If resources are not sufficient to carry on with a connection, the requested connection might be impossible to accommodate. The only solution will be to drop some of the existing connections. If the policy states it, this will be done as in the case of a sudden reduction in bandwidth by dropping of existing ones. In the case the user gives priority to a stream, it will be reduced in quality or dropped last. These policies and user interaction are outside the scope of this work.

3.2 Status Matrix and Order Matrix

The Status and the Order Matrix act as an interface between the resource manager and devices in the system. For each resource in the system, there are a two (sub) matrices (Status and Order).

The Status Matrix contains information about available resources in the system. Each resource is represented by its

- current value (out of the limited number range)
- current granularity, i.e., the time interval until which the current value is likely to not change, and
- likelihood that 2) holds.

These values are provided by the devices (order managers). A single link on, say, wired switched Ethernet, will have a high granularity interval and high likelihood, whereas a wireless link in a mobile environment might result in small values for each. While accurate and correct predictions will not be possible, these values support better estimates for the decisions of the resource manager than very pessimistic values only. Should the granularity interval be less than is useful for resource management, the associated value for the device state can be assumed 0.

The *Order Matrix* contains directions for resources reservation on the devices, made by the resource manager. Each device is presented by one element in the Order Matrix, from where the device picks its order in form of

- delay (sub-delay)
- value (out of the limited number range, QoS performance levels)

Source and destination devices/resources, as user priority request, are specially marked in both matrices.

3.3 Order manager

An order manager is responsible for allocating resources at a device. It maps global resource reservation constraints (orders), made by the resource manager, to the concrete scheduling specification for a local scheduler.

Another task of the order manager is to provide the Status Matrix with information about locally available resources, in form of well defined QoS levels; for both CPU and bandwidth.

The information about available resources is determined repeatedly, but not periodically by the order manager. The accuracy of the information depends on a chosen temporal granularity. Hence, one order manager is responsible for

- determining of the available resources at the devices
- transforming various kind of traffic specification into a few QoS levels and providing the Status Matrix with them
- allocating resources at devices and providing parameters to local schedulers

3.4 Local scheduler

A local scheduler is responsible for scheduling of device's local CPU resource or outgoing network packets. It is placed on a device and together with the order manager, it enforces local resource reservation.

As mentioned before, the order manager provides parameters to the local scheduler, which then performs the actual scheduling of the streams from this node.

3.5 Local monitor

A local monitor will perceive changes in the bandwidth availability class with a given granularity. Received and specified performances will be compared by the local monitor and the outcome of the comparison will be sent as feedback information to the order manager. If the significant resource change has been observed, it will be reported to the Status Matrix, otherwise it is ignored.

4 Evaluation

The FABRIC application interface is derived from the HLA API. HLA is an interoperability standard that is based on an anonymous publish/subscribe mechanism that allows for distributed applications (federates) to exchange data in a loosely way [5].

Federates express their interest in receiving data of specific object class (attributes) by using "subscribe to" mechanism. Likewise, federates that own object attributes can update their attributes values by using "publish".

We have implemented the Matrix approach using HLA within FABRIC.

The resource manager and the order managers are implemented as federates. The Status/Order matrix is a collection of objects.

The resource manager subscribes to quality requests, expressed in the Status matrix, that are published by streaming applications. The order manager subscribes to the Order matrix to receive orders from the resource manager.

By using HLA, messages are only sent when attributes of the matrix elements are actually updated.

The HLA offers the dynamic addition and removal of members of a federation. This HLA's characteristic is well suit for a redundancy of devices in the Matrix approach. Federates (order managers or a resource manager on a device) can fail without a major impact on the overall functionality. The lost federate (device) can be replaced during run time with some shadow federate (device).

5 Acknowledgments

The Matrix approach is developed within the FABRIC project. The authors would like to acknowledge our partners in the project: Philips(NL), Thomson(FR), TNO(NL), TUE (NL), CSEM(CH), INRIA(FR), SSSA(IT) and UP(DE) .

References

[1] Deliverable D3: Application Requirements, FABRIC project, http://www.extra.research.philips.com/euprojects/fabric/

[2] Deliverable D4: Streaming specification, FABRIC project, http://www.extra.research.philips.com/euprojects/fabric/

[3] Deliverable D8: Streaming design: analysis, control and scheduling, FABRIC project, http://www.extra.research.philips.com/euprojects/fabric/

[5] IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Interface Specification, No.: 1516.1-2000

[6]Tatsuya Yamazaki and Jun Matsuda "On QoS Mapping in Adaptive QoS Management for Distributed Multimedia Applications"

[7] "A Survey of Quality of Service Architectures" Aurrecoechea, Campbell and Hauw, Center for Telecommunication Research, Columbia University, NY

[8] Fukuda K., Wakamiya N., Murata M., and Miyahara H., "QoS Mapping between User's Preference and Bandwidth Control for Video Transport", Proc. 5 th International Workshop on Quality of Service (IWQOS'97), Columbia University, New York, US