The Matrix - A framework for real-time resource management for video streaming in networks of heterogenous devices

Larisa Rizvanovic
Mälardalen University
Department of Computer Science and Electronics
larisa.rizvanovic@mdh.se

Gerhard Fohler
University of Kaiserslautern
Department of Electrical and Computer Engineering
fohler@eit.uni-kl.de

Abstract

Efficient transport of streams with acceptable playout quality in in heterogenous, dynamic environment (e.g. home networks) requires management of both networks and CPUs. This paper presents a framework for applying real-time resource management methods for decoupled video streaming of heterogenous devices. It is based on a global abstraction of device states, which reduces system state information and decreases overheads for its determination and dissemination. It provides access to the entire system state in acceptable fresh way, enabling system wide optimized decisions to be taken. This work has started as part of a FABRIC EU IST project. 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, the FABRIC aimed to handle the complete network to satisfy End-to-End Quality of service (QoS) requirements. 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

In real-time systems, methods for scheduling and allocation have been developed to provide resource management including guarantees. These are capable of determining whether resources are sufficient for the timely completion of tasks and if so, provide reservation methods to ensure these resources are available for the guaranteed task when executing. Real-time resource management can thus manage all resources: it keeps track of the state of resources, provides admission control and resource guarantees. The resource management methods from real-time systems can provide key capabilities for the handling of streams and devices in home network.

However, the area of home networks introduces a number of crucial differences and issues compared to classical "process control" real-time systems. For example, home networks are *heterogeneous* , where different local schedulers for CPUs and networks exist on diverse devices. An approach with tight coupling between global management and diverse local schedulers is not appropriate, since it will have unfeasible high overheads and require a fixed, known set of schedulers. Then in Home networks have to cope with *limited resources* where real-time activities, including schedulers, execute on the same resource they are handling. Optimum scheduling can easily take more CPU cycles than the execution of the scheduled software takes. Finally, in home networks we have *highly fluctuating resources*, possibly demanding adaptation in resource allocations. Thus, media applications are highly dynamic due to the dynamic nature of the audio/video media content. When processing a video stream there are two types of load fluctuation due to data dependency; temporal and structural [6]. A temporal load is also called a stochastic load and it can be caused by different frame size of stream. A structural or systematic load is often due to a scene change. On the other hand on a wireless networks we talk about long- and short-term bandwidth variations. The long-termed variations can be caused by another application in the system. The short-term changes are often present in the system due to radio frequency interference, like microwave ovens or cordless phones. In contrast, classical real-time systems respond to a limited number of events, such as task arrivals, activations, and completion.

Key issue for enabling resource handling with real-time methods in home networks are efficient representation of the fluctuating system state, resource allocation decisions, and dissemination of orders. 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.

Hop-by-hop vs global view decisions

Efficiency of distributed resource management is critical threatened by overheads, as devices have to exchange information to determine a global system view for resource management 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 hop-by-hop 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 resources 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 later 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. 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.

A further issue affected by the limited local state knowledge concerns the decomposition of end-to-end delay of the stream into deadlines for each of the devices on the route, forming deadlines for task scheduling and transmission delays. If these device sub deadlines are chosen without consideration of the state of all devices at the time, e.g., of equal length, the following may happen. Suppose device A is heavily loaded, B not at all. With the equal assignment, the deadline on A is likely to be missed; B will meet its deadline easily. Setting dl(A) significantly larger and dl(B) correspondingly smaller, both deadline would have been met. Complexity increases as the load on each device varies over time.

Chosen approach

We believe that key issues to enable resource handling with real-time methods in home networks are the interfacing between devices and resource management, providing a relevant view of the system state and diffusion of decisions to the devices. That is way we propose We propose an 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 a 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. Furthermore, failures in devices or communication will not block or delay resource management.

The Matrix provides a logical abstraction of the view of the system state, not an actual centralized implementation requirement. Rather, we envision that the Matrix will be 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.

The rest of this paper is organized as follows. In the next section we give an overview of the related work. Section 2 describes the Matrix approach. Section 3 presents the current implementation status of the Matrix framework, while the results of the Matrix approach's evaluation are presented in 4. And finally we summarize work presented in this paper in 5.

2 MATRIX overview

The basic idea of the Matrix is to provide a global abstraction of device states as representation of the system state for resource management and to decouple device scheduling and system wide resource allocation.

As we have already mentioned, we want to use the minimum relevant information about devices states as needed for resource management, in order to reduce the system state presentation, and to abstract over fluctuations, which could overload scheduling of resources. Thus, we use the notion of a few *abstract QoS levels* that represent a resource's availability and an application's quality (see Figure 1). For example, the variations in the quality of network link connection between two devices can be represented by e.g., three abstract QoS level values, (L)ow, (M)edium and (H)igh. H means that the data can be transmitted through the link with full available capacity, while L indicates severe bandwidth limitations. Likewise, quality of each application using certain resources is mapped to a finite number of application QoS levels. In this work, we apply linear mapping between the resources and the QoS levels, e.g., based on experimental measurements [5]. A more advanced mapping could, for instance, use fuzzy logic to provide a larger number of QoS levels with finer granularity, but QoS mapping is an ongoing work and it is out of the scope of this paper.

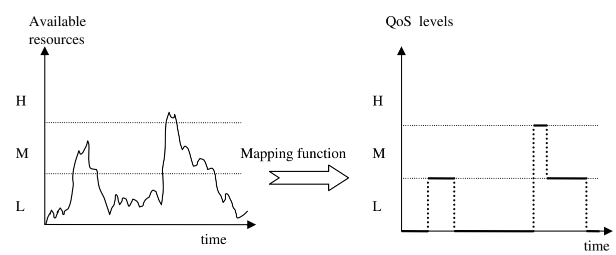

Figure 1: Reduced system state in the Matrix

In the Matrix we also provide an interface to decouple device scheduling and system resource allocation. Instead of the resource manager probing local schedulers for state information, the devices provide information about relevant state information and estimations about changes with appropriate, individual granularity themselves.

Consequently, contribution of the Matrix framework can be summarized in two parts:

1. *Efficient system state presentation* (reduced system state); comes as the result of the mapping approach.

2. *Interface to decouple device scheduling and system resource allocation* (resource allocation decisions and dissemination of orders); the result of the Matrix's architecture.

2.1 Architectural design aspects

The Matrix is composed of several entities that constitute an effective mechanism for scheduling and monitoring of available resources in the system. Figure 2 shows the

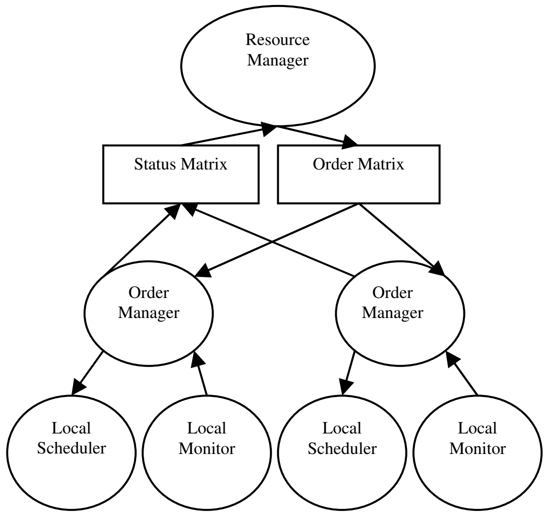

Figure 2: Information flow between Matrix's components

data flow (information flow) between the Matrix components. The functions of these components are further discussed in the following subsections.

2.1.1 Resource Manager

For each "domain" [1] there is one *Resource Manager*. The Resource Manager will be used to schedule and reserve resources within the domain. One part of scheduling of streams 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. 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, it 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.

2.1.2 Status Matrix

A *Status Matrix* contains information about available resources in the system, which is provided by the *Order Managers*, located on the devices. For each type of shared

[1] A subnetwork within which common characteristics are exhibited, common rules observed, and over which a distribution transparency is preserved

resources, there is an Order Manager responsible for publishing the current resource availability on the device in the Status Matrix. As we already mentioned above, available resources will be mapped to just a few different QoS levels. This implies that we do not need to bother about updating the Matrix with each oscillation around a value. Instead, Order Mangers will update the Status Matrix when a change to a different quality level occurs for a significant amount of time.

Furthermore, 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.

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.

2.1.3 Order Matrix

A *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). It is a sub delay for one device in the stream path.
- value (out of the limited number range, QoS performance levels)

Thus, The Order Matrix contains orders for resources allocation, made by the resource manager. Source and destination devices/resources, as user priority request, are specially marked in both matrices.

2.1.4 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 Local Schedulers. 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:

1. collecting information about the available resources at the devices
2. transforming various kind of traffic specification into a few QoS levels and providing the Status Matrix with them
3. allocating resources at devices and providing parameters to Local Schedulers.

2.1.5 Local Scheduler

A *Local Scheduler* is is responsible for scheduling of a local resources, e.g., a network packets scheduler that adjusts the packet sending rate according to available bandwidth. 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.

2.1.6 Local Monitor

The information about available resources is provided to the Order Manager through *Local Monitors*. Thus, Local Monitors are responsible for continuous monitoring of a resource availability on a device, e.g., the available CPU or the network bandwidth. The accuracy of the information depends on a chosen temporal granularity.

2.1.7 One Operational Scenario

Consider the following motivating example: *A person uses a PDA (Personal Digital Assistant) to watch a video stored on a local video server, which is delivered to the PDA through a wireless network. As the person moves around with the PDA, the amount of the available bandwidth varies, which can result in video interruption due to packet lost. Fortunately, the video server is able to provide the stream with different qualities, and the user can continuously watch the stream on his/her PDA, although with the varied quality.*

For the sake of simplicity, in this scenario we consider just one type of resource, available network bandwidth. We assume the scenario network like one presented in Figure 3, where connection between the PDA and the video server is archived via one wireless router. Hence, in our example scenario we have two connection link, A and B. We also assume that from the beginning the quality of these two connections are high, i.e., it corresponds the level H.

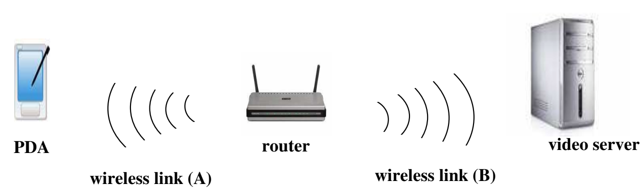

Figure 3: Scenario network

Now, assume that the person with the PDA starts moving further from the router. The Local Monitors on the one of the involved devices, i.e., either on the router or on the PDA, will detect the decreased available bandwidth on the link. The adjustment of resource usage will be performed, with the following steps involved:

1. The Local (Bandwidth) Monitor on the PDA observes the current bandwidth value of the wireless network (on the link A).

2. The Local Monitor reports the current bandwidth value to the Order Manager on the PDA.

3. The Order Manager compares the observed bandwidth value with the threshold values imposed by current quality level, i.e., H. Since the currently observed quality level L, is different from the previously published one, the Order Manager will publish it in the Status Matrix.
   a) Contrary, if the observed quality level would be the same as previous one (H), there will be no need to publish it again.

4. After publishing the new quality value for connection link A in the Status Matrix, the Resource Manager is automatically notified about this. (That is enabled by High Level Architecture (HLA), see **??**)

5. The Resource Manager has an overview of whole resource situation (i.e., for all involved devices). It knows that there is no point in sending a high-quality stream from the server to the router through the link B, because the router cannot forward it with the same quality via the link A (which is L). The Resource Manager decides to degrade the quality of the link B to level L. Now, the link B will not use the same amount of bandwidth as initially, which means that this spare capacity can be reused by some other connections instead. Thus, the Resource Manager sets qualities of the both links to low (L).

6. Then, the quality changes are published in the Order Matrix and become visible for Order Managers on all devices (again thanks to HLA).

7. The Order Managers on respective devices are informed about the new values. Then they translate the abstract levels to application specific concrete values to be used by the Local Schedulers. In the case of the server, this mean also that a lower quality of the stream has to be sent out.

8. The Local Schedulers adjust the transmitted packets rate according to these new values.

2.2 Streaming without versus with the Matrix approach

Theoretically, it can be envisioned that a single device does not participate in the Matrix approach. However, the higher abstraction level provided by the Matrix and decreased overhead would be lost. Obviously, the resulting complexity would effect the quality of the decisions and the overheads as well as impact the rest of the system. Devices either join the Matrix fully, i.e., provide input to the Status Matrix and follow the Order Matrix, or not at all.

2.3 Control aspects - feedback

The adaptation to varying resource availability may result in feedback effects. Suppose availability goes down, to which resource management might respond with a less demanding stream. As effect, the availability would raise again, as less is needed for the new stream. Thus, the manager could decide to increase stream demand, and so on. Control theory provides ample results for similar effects in the time dimension, but fails to provide results for such event-based systems. Such issues are beyond the scope of FABRIC. We believe future research could apply a heuristic approach to avoid some effects, e.g., by limiting the adaptation frequency. For example, the application of fuzzy controllers monitoring system state appears promising.

2.4 Granularity issues

As discussed earlier, granularity issues are essential to provide an acceptably fresh view of the resources' states. On one side, it cannot be too small, else the overheads for state and order updates will overload the system. On the other side, being too large reduces the freshness of the system view and can introduce the aforementioned control problems. The determination of (temporal) granularity is done within the order manager, as it is closely coupled to the actual resource and scheduler.

3 Implementation

The Matrix framework is quite complex and we are still working on its full implementation. However, we have implemented a mock-up of Matrix approach within FABRIC, by using HLA (High Level Architecture). HLA is a middleware standard for connectivity and data sharing in simulation applications. [7, 1].

3.1 The Matrix and HLA

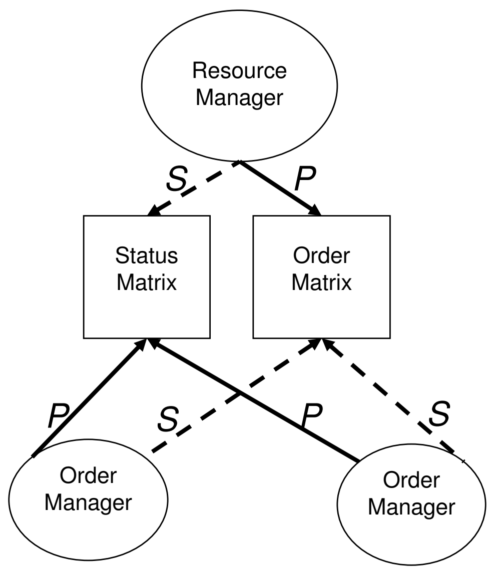

Figure 4: Matrix and HLA (*P stands for publish, S for subscribe*)

The HLA is an interoperability standard, based on an anonymous publish/subscribe mechanism. Its publish-subscribe model connects anonymous information producers (publishers) with information consumers (subscribers). In the terminology of HLA, individual simulations are known as federates. The collection of federates brought together to simulate a complex environment is known as a federation. In the Matrix's mockup, Resource Manager and Order Mangers are implemented as federates. The Status Matrix contains information produced by various nodes in the system. Therefore the Status Matrix can not be a federate. Rather, we present the Status Matrix as a collection of objects. Thus, the elements of Status Matrix are published by the Order

Managers placed on nodes. Each Order Manager adds an entry in the Status Matrix and becomes owner of this object. The Resource Manager, is interested in information published in the Status Matrix, i.e. it subscribes to the Status Matrix.

The Order Matrix is represented as a collection of objects too. The elements of the Order Matrix will be published by the Resource Manager. Hence, it is a task of the Resource Manager to update entries in the Order Matrix. The subscribed Order Managers are reflected accordingly (see Fig 4).

3.1.1 The Matrix classes

MatrixCell - This class is base class for StatusMatrixCell and OrderMatrixCell classes. The following attributes are provided:

- *OrderManagerID* - The uniq indentifier of an order manager
- *StreamID* - Each stream ID is allotted by the order managers on the device that can produce the stream. If the device can produce one stream with different qualities, each version of stream will get a unique ID.

StatusMatrixCell - This class represents an entry in the Status Matrix. This class is subclass of the MatrixCell class and inherits all its attributes. Each resource is represented by the following attributes:

- *Current value* - Current resource availability. This value is out of the limited number range (QoS performance level).
- *Current granularity* - It is the time interval until which the current value is likely to not change.
- *Likelihood* - This attributes gives the probability that the given value under attribute "current granularity" will hold.

OrderMatrixCell - This class represents an entry in the Order Matrix. This class is derived from a MatrixCell class. Each this entry is one order from the resource manager for resource allocation. The attributes of this class are:

- *Delay* - It is a sub delay for one device in the stream path
- *Value* - This value is out of the limited number range (QoS performance level).

3.2 Implemented Modules

The hierarchical architecture and the loose coupling between system modules makes it possible to work on different parts independently of each other. Current implementation includes Local Monitors and Local Schedulers for CPU and network bandwidth.

Local Network Scheduler – For network scheduling we use the traffic shaping approach, which provides different QoS by dynamically adapting the transmission rate of nodes, to match the currently available bandwidth of a wireless network. The Traffic Shaper adjusts the outbound traffic accordingly to input parameters (i.e., the amount of available bandwidth assign to the Local Scheduler). Please see [5] for full implementation details of the Traffic Shaper.

Local Network Monitor – For monitoring and estimation of available bandwidth (over 802.11b wireless Ethernet), we use a method that provides us with the average bandwidth that will be available during a certain time interval. The architecture consists of a bandwidth predictor that first uses a simple probe-packet technique to predict the available bandwidth. Then, exponential averaging is used to predict the future available bandwidth based on the current measurement and the history of previous predictions. Also, we refer to [5] for details.

Local CPU Scheduler – The allocation of CPU to the applications depends on the scheduling mechanism that is used. We have developed a predictable and flexible real-time scheduling method that we refer to as *slot shifting* [4]. The basic idea is to guarantee a certain quality of service to applications before run-time, and then adjust it at run-time according to the current status of the system. The full details about the scheduler and its application in the context of media processing can be found in [3].

Furthermore, we are even looking into other scheduling methods. One suitable technique that fits our needs *elastic scheduling approach* [2]. It handles overload through a variation of periods and in that way manage to decrease the processor utilization up to a desired level.

Local CPU Monitor – Since we use a real-time scheduling mechanism, the CPU monitoring is very simple to achieve. The *spare capacity* mechanism of slot shifting provides easy access of the amount and the distribution of available resources at run-time [4].

4 Evaluation

We have evaluated our method in the context of video streaming. Here we present results from a 15 minutes video streaming simulation using our integrated approach for global and local adaptation. We simulate usage of 10 devices in the system and show how a MPEG-2 video stream is adapted based on current resource availability (network bandwidth).

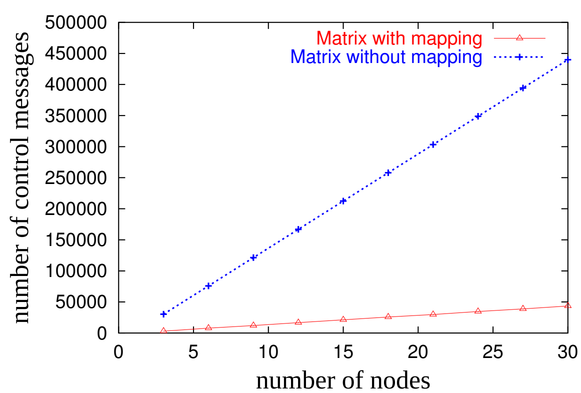

Figure 5: Reduced system state presentation

The simulator is composed of a number of processes, representing one Resource Manager and different devices (more correctly , Order Managers on devices). Local Schedulers and Local Monitors are omitted. Information about available bandwidth on wireless network is obtained from a file which is a result of another research project about bandwidth estimation algorithms for wireless networks [5].

Figure 5 shows how the number of the control messages increase with the increased number of devices in the system, with and without the Matrix. It is clear that we have achived a significant reduction of the number of the control messages with the Matrix mapping approach.

Figure 6 shows the possibly wasted resources due to limited local system view on device, i.e. difference between QoS levels decided on the device and global resource manager. Here we have simulated usage of 10 devices in the system and show how a MPEG-2 video stream is adapted based on current resource availability (network bandwidth). We use the following quality levels for available bandwidth (given in Mbps):

$$
\begin{aligned}
q_1(BW) &= [1.5, 2.5] \ \ (L)\\
q_2(BW) &= [2.5, 4] \ \ (M)\\
q_3(BW) &= [4, 11] \ \ (H)
\end{aligned}
$$

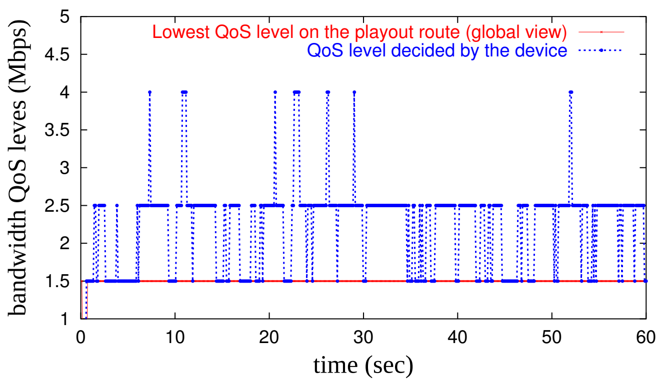

Figure 6: Local vs global system view

Thus, Figure 6 illustrates efficiency of our Matrix approach where adjustment of resources is not just based on the limited local system view of one device (like in hop-by-hop approach), but also on the current available resources of all involved devices. In that way, our approach enables a system wide optimization.

5 Conclusion

In this paper, we presented the Matrix, the adaptive framework for applying real-time resource management methods for decoupled video streaming of heterogenous devices. 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. Hence, the Matrix is an efficient system state presentation and an interface to decouple device scheduling and system resource allocation.

Finally we want to stress that 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.

References

[1] IEEE standard for Modeling and Simulation, High Level Architecture (HLA) - federate interface specification, no.:1516.1-2000.

[2] G. Buttazzo, G. Lipari, M. Caccamo, and L. Abeni. Elastic scheduling for flexible workload management. *IEEE Transactions on Computers*, 51:191–302, March 2002.

[3] D. Isovic. *Flexible Scheduling for Media Processing in Resource Constrained Real-Time Systems*. PhD thesis, 2004.

[4] D. Isovic and G. Fohler. Efficient scheduling of sporadic, aperiodic, and periodic tasks with complex constraints. In *21st IEEE RTSS*, USA, November 2000.

[5] T. Lennvall and G. Fohler. Providing adaptive qos in wireless networks by traffic shaping. In *Resource management for media processing in networked embedded systems (RM4NES)*, Eindhoven, Netherlands, March 2005.

[6] C. O. Perez, L. Steffens, P. van der Stok, S. van Loo, A. Alonso, J. F. Ruíz, R. J. Bril, and M. G. Valls. QoS-based resource management for ambient intelligence. In *Ambient intelligence: impact on embedded system design*. Kluwer Academic Publishers Norwell, MA, USA, 2003.

[7] L. Rizvanovic and G. Fohler. The MATRIX: A qos framework for streaming in heterogeneous systems. *International Workshop on Real-Time for Multimedia, Catania, Italy*, 2004.