Prioritization-based Bandwidth Allocation for MOST networks

Giuliana Alderisi[*], Giancarlo Iannizzotto[#], Gaetano Patti[*], Lucia Lo Bello[*]

[*]Department of Electrical, Electronic and Computer Engineering, University of Catania, Italy
{giuliana.alderisi, lucia.lobello, gaetano.patti}@ dieei.unict.it

[#]Department of Cognitive Studies, Educational and Cultural Studies, University of Messina, Italy
ianni@unime.it

Abstract

The Media Oriented Systems Transport (MOST) network, the current de-facto standard for in-vehicle multimedia and infotainment, does not provide support for the stream priority and serves bandwidth allocation requests based on their arrival order, irrespective of the stream importance. This work proposes a novel bandwidth allocation approach for multimedia streams on MOST150 networks that introduces the stream priority, which is used to reserve bandwidth for each stream according to its importance. The approach also exploits the ability for dynamically changing the bandwidth needs of multimedia streams through variable bitrate compression and is useful in MOST 150 networks supporting, together with multimedia streams, also bandwidth-greedy applications, such as Advanced Driver Assistance Systems.

1. Introduction and Motivations

Very different networks are used in today's vehicles to develop electrical/electronic architectures depending on their purpose and on the application domains [1]. In-vehicle multimedia and infotainment communication is currently dominated by the Media Oriented Systems Transport (MOST) [2], although several studies indicate Ethernet AVB as a promising alternative for the near future [3-8]. Gateways to connect MOST to other in-car networks are addressed in [10][11]. MOST is optimized for automotive applications and offers a low-overhead and low-cost interface for communicating with a variety of in-car peripheral devices. With the introduction of MOST150, operating at 150 Mbit/s, there are enough resources for multiple high-definition data streams. For this reason, the adoption of MOST technology in applications such as Advanced Driver Assistance Systems (ADASs) is currently investigated.

However, for the sake of reducing wiring complexity and cost, a single network for both ADAS and In-Vehicle Infotainment domains is considered the preferred design option. Moreover, as shown in [12], the number of in-car ADAS devices (e.g., cameras and pre-processing units) is steadily growing, thus increasing the network complexity and the bandwidth requirements. Therefore, if ADAS flows and multimedia streams have to share the same MOST network, special attention has to be paid to bandwidth allocation. In fact, the bandwidth requirements of ADAS flows are more stringent as compared to multimedia, as the image processing and analysis algorithms used to extract information from the video streams are often very sensitive to the degradation introduced by strong (lossy) compression.

Another point that deserves consideration while allocating bandwidth to multimedia streams is that current audio/video encoders produce variable bitrate traffic, therefore a stream, on average, could require less bandwidth than the provisioned one. Moreover, by tuning the compression on the fly, widely used encoders such as MPEG2 and H.264 [9] can lower or increase the bitrate of each produced stream, thus reducing or improving the quality of the streamed media. As a result, a trade-off between bandwidth requirements and media quality for the multimedia streams to improve the network efficiency can be sought. In addition, there is a variety of multimedia streams, with different importance from the end user perspective, so the bandwidth allocation should also take into account the stream importance. However, MOST does not provide support for the stream priority and bandwidth allocation is made based on the arrival order of the bandwidth allocation requests, irrespective of the stream importance.

This paper proposes a bandwidth allocation mechanism for MOST networks that takes explicitly into account the importance of the stream, here represented by the stream priority. This approach is realized through a Prioritization-based Bandwidth Allocation Scheme (PBAS) that exploits the ability for dynamically changing the bandwidth needs of multimedia streams through variable bitrate compression.

The paper is organized as follows. Section 2 provides an overview of the MOST protocol. Section 3 describes the novel approach presented in this work. Finally, Section 4 outlines ongoing work.

2. The Media Oriented Systems Transport (MOST) protocol

MOST [2] is a synchronous network, in which a Timing Master provides the system clock through a continuous data signal. All the other devices, called Timing Slaves, synchronize their operations to the reference signal. The bandwidth allocated for streaming data connections is always available and reserved for the dedicated stream, so data streams are transmitted without interruptions, collisions, or delays.
There are three different MOST versions, this paper addresses MOST150. A MOST150 frame (Fig. 1) is subdivided into three parts that in the standard are called channels and are defined as regions of the frame dedicated to a specific kind of traffic, as follows:

- Control Channel, which transports Data packets for control messages and is generally used for event-driven transmissions characterized by low bandwidth and short packet length.
- Streaming Data Channel, which transports continuous data streams requiring high bandwidth and time-synchronized transmission. Streaming connections are dynamically managed by a network component, called a Connection Manager (CM), through appropriate control messages. Each streaming connection is specified by the connection label and the required bandwidth. Following the request of an initiator, i.e., an entity that, upon reception of an end user command (e.g., to play a movie) requests a stream connection (e.g., from a DVD Player to a monitor), the CM starts a connection establishment procedure between a sender (*source*) and a receiver node (*sink*). First, the CM issues a bandwidth allocation request to the source. Then, the source checks for the available bandwidth and replies to the CM, which communicates the connection parameters to the sink. Finally, after the sink acknowledgement, the connection is successfully established and the CM confirms it to the initiator. This mechanism provides a bandwidth allocation, for streaming connections, in FIFO (First In First Out) order, i.e., requests are served in their arrival order and only if there is bandwidth available, without taking into account the streams importance. Therefore, it might happen that an important stream cannot establish a connection as the bandwidth is fully allocated.
- Packet Data Channel, which is used for transmitting data blocks with large size and for burst transmissions requiring high bandwidth.
- Ethernet Channel, which implements the protocols and mechanisms required to support Internet access in a distributed infotainment system.

A MOST device, as shown in Fig. 2, can be connected to a MOST network via a MOST Network Interface Controller (NIC). At the Application level, a MOST device contains multiple components, called Function Blocks (FBlocks). Each FBlock provides a specific system functionality (e.g., audio/video stream). The Network Service implements an intermediate layer between the FBlocks and the MOST NIC, providing functions to simplify the MOST NIC management.

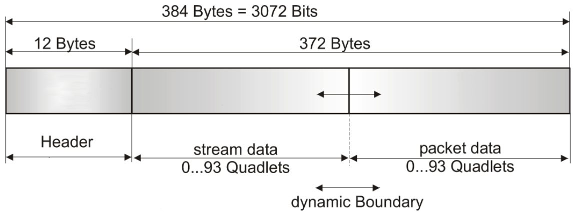

Figure 1 – MOST Frame

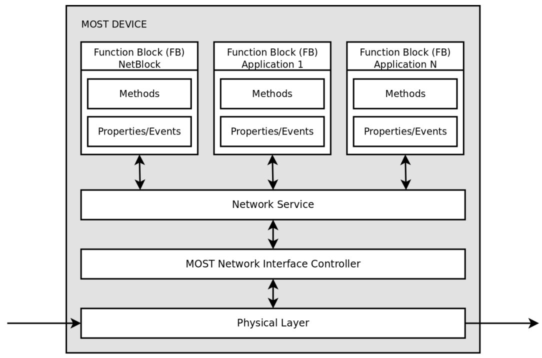

Figure 2 - MOST device functions

2.1. MOST Data

Messages sent over a MOST network can be of the following types:

- Control Data: Messages sent, on the Control Channel, to one or multiple receivers. Any node can transmit a control message.
- Source Data: Any data not transmitted over the Control Channel, such as packet data and streaming data. Streaming data may be either synchronous or isochronous. In the first case, they follow the network sampling rate, while in the second case they are sent or received at a rate that is independent of the MOST system clock. Both synchronous and isochronous data are transmitted from a single source to one or more sinks.

A MOST150 frame consists of 3072 bits (384 bytes). The header, that is 12 bytes long, is used for administrative purposes, and includes 4 bytes for control data. The next 372 bytes are used for streaming data transfer and packet data.

3. Prioritization-based bandwidth allocation

The MOST protocol does not provide any kind of prioritization. An approach to introduce node priority in MOST, for Packet Data Channel only, is outlined in [13], where the prioritization of the packet data transfer is realized via an access disclaimer of the MOST frame: A node with a low priority lets the MOST frame pass by several times before filling it with packet data, thus giving other nodes a higher priority for filling the MOST frame.

Differently of [13], this work does not deal with node prioritization, but proposes a method to implement stream prioritization in MOST. The aim here is exploiting the stream priority to perform bandwidth allocation in a more efficient way comparing with the MOST standard.

As it will be described in the following, priority is assigned to each stream by the Connection Manager via a control message that is sent in the configuration phase. Here priority is a static value, which is determined depending on the type of application the stream is associated to.

The proposed approach focuses on multimedia traffic, which is typically characterized by a variable bitrate. As, due to the variable bitrate, sometimes a multimedia stream does not use all the bandwidth it is reserved, our approach aims at exploiting the unused bandwidth to improve the service provided to the other streams. In the following, by “stream” we mean multimedia stream, unless explicitly specified.

In our approach, other flows than multimedia streams (e.g. ADAS ones) are assigned bandwidth on the Stream Data Channel even if their connections are not yet established. This is possible as the requirements for such critical flows are known a priori. These flows are not affected by the approach here proposed, which only targets multimedia streams.

Our approach is based on the following concepts:

- Assigning a priority to each registered stream;
- Assigning to each stream a reserved bandwidth, sufficient to guarantee a minimum acceptable quality level, based on the stream specifications. The minimum quality level is typically in the range between 70% to 80% compared to the full quality requirements;
- The stream source initially sets its transmission bitrate according to the bandwidth assigned, by tuning the compression parameters.

A fixed fraction of the MOST total bandwidth is managed as a “spare bandwidth” that is distributed to the streams according to both their declared needs (in terms of the bitrate needed to reach their full quality level) and their priority.

Bandwidth distribution is performed by partitioning the spare bandwidth into subchannels, that are part of the Streaming Data Channel, as shown in Fig. 3. The stream with the highest priority will be assigned one or more subchannels according to the number of subchannels available and its declared needs. Then, if there are subchannels left, the second highest-priority stream will also receive additional subchannels to match its declared needs, and so on. If a stream cannot receive enough subchannels to fully meet its bandwidth requirements, it is assigned only the available subchannels. The bandwidth distribution procedure repeats until there are no more subchannels to allocate. This way, each registered stream is not only granted the minimum bandwidth required, but may also receive an additional bandwidth, whenever available, depending on its priority and declared full-quality requirements.

Once the stream source is notified about the additional bandwidth, it increases the bitrate (by tuning ‘’on the fly’’ the compression parameters) so as to improve the stream quality as much as possible.

If a subchannel becomes unused, either because the stream it was assigned to has reduced its bandwidth requirements due to the variable bitrate encoding or because the stream has been turned off, the subchannel is reallocated to the next higher priority stream that needs additional bandwidth.

The approach proposed in this paper improves MOST in two respects. First, here the number of registered streams can be larger, as all the registered streams are initially granted bandwidth to achieve not their maximum, but their minimum acceptable quality level. Second, here not all streams are equal, but a priority value reflects the stream importance. Priority allows the most important streams, i.e., those with the highest priorities, to receive additional bandwidth, according to their needs, only when they actually require it. The described mechanism can be realized in MOST by setting the amount of bytes allocated to a stream in each MOST frame. The new approach foresees that a portion of the Streaming Data Channel of the MOST frame is distributed to the streams based on their priority. This part of the MOST frame is β bytes long and is split into n equally-sized subchannels, as shown in Fig. 3. Each stream can be assigned one or more consecutive subchannels.

3.1. The algorithm

In the proposed approach, a connection between source and sink is established, as foreseen in the MOST standard, following the request of the initiator.

The behavior of the Connection Manager when a new connection request is made by the initiator is described in pseudocode in Fig. 4 by the *newConnectionRequestHandler* method. When a stream connection request arrives to the Connection Manager, it checks if there is available bandwidth to meet the minimum bandwidth requirements for the stream. If so, the CM inserts the request in a list, ordered by the streams priority. The CM contains a mapping table that matches the various applications to their priorities and the table is configurable at run time, taking also into account the end users preferences. Then the subchannels assignment phase starts. Starting

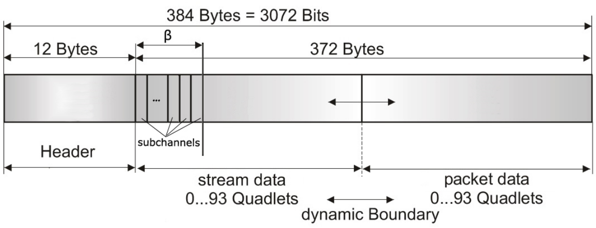

Figure 3 – MOST frame with subchannels

```
function
newConnectionRequestHandler(ConnectionRequest req){

  if(availableBandwidth ≥
  req.minRequestedBandwidth) {

  ConnectionList.insert(req);

  lastAssignedSCIndex = 0;

          foreach(Connection in ConnectionList)
     {

         subChannelNumber =
         min(Connection.requestedSC,
                        (totSC-l
         astAssignedSCIndex+1));
         if(subChannelNumber > 0)
       {
        Connection.startSC = lastAssignedSCIndex+1;
        lastAssignedSCIndex += subChannelNumber;
       }}}}
```

Figure 4 – Subchannel allocation algorithm

from the first connection in the list, (i.e., the highest priority one), the minimum between the number of required subchannels and the number of available ones is assigned to each connection. The ConnectionRequest is a data structure, sent from the initiator and received by the CM, that contains the following fields: the subChannelNumber, i.e., the number of subchannels assigned to the stream source for transmission; the subChannelStart, i.e., the index of the first subchannel assigned; the MinRequestedBandwidth, i.e., the minimum bandwidth required for serving the stream;
the StreamPriority, which represents the priority assigned to the stream.

ConnectionReqList is the list, ordered by the streams priority, of both established and not yet confirmed connections, while LastAssignedSCIndex is a support variable needed, at run time, to keep track of the last subchannel that has been allocated.

The next steps for establishing streaming connections follow the standard [2], and the total amount of bandwidth allocated for a specific stream is the sum of the MinRequestedBandwidth contained in the relevant ConnectionRequest and the bandwidth associated to the subchannels assigned.

The ConnectionRequest data structure contains the information about the portion of the MOST frame that is allocated to a specific stream. The source node can insert in each MOST frame a number of bytes equal to the number of bytes allocated in the Stream Data Channel plus the number of those allocated in the subchannels. If the ConnectionRequest is rejected by the source or the sink, it is deleted from the ConnectionReqList and the allocation of the subchannels is rebuilt.

4. Conclusions and Future work

The approach presented in this work represents a first step. Ongoing work is dealing with further refinements of the PBAS mechanism, such as, exploiting the streams prioritization for deallocating a lower priority stream in order to allocate bandwidth for a newly arrived stream with higher priority. Also, the benefit of PBAS compared to other techniques proposed in the literature, such as, the Elastic Task Model, will be investigated. In addition, comparative assessments between native MOST150, MOST150 with PBAS and AVB will be performed via OMNeT simulations.

References

[1] T. Nolte, H. Hansson, L., Lo Bello "Automotive Communications - Past, Current and Future", in Proc. of the 10th IEEE Int. Conference on Emerging Technologies and Factory Automation (ETFA'05), 985-992, Catania Italy, Sept. 2005.

[2] MOST Media Oriented System Transport, Multimedia and Control Networking Technology, Rev. 2.4, OASIS Silicon Systems, 2005.

[3] L. Lo Bello, "The case for Ethernet in automotive communications," SIGBED Review, vol. 8, n..4, pp. 7–15, Dec. 2011.

[4] G. Alderisi, G. Iannizzotto, and L. Lo Bello, "Towards 802.1 Ethernet AVB for Advanced Driver Assistance Systems: a preliminary assessment," Proc. of the 17th IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2012), Kracow, Poland, Sept. 2012.

[5] G. Alderisi, A. Caltabiano, G. Vasta, G. Iannizzotto, T. Steinbach, L. Lo Bello, "Simulative Assessments of IEEE 802.1 Ethernet AVB and Time-Triggered Ethernet for Advanced Driver Assistance Systems and In-Car Infotainment", in Proc. of the IEEE Vehicular Networking Conference (VNC), pp. 187-194, Nov. 2012, Seoul, South Korea.

[6] J., Noebauer, H., Zinner, "Payload Efficiency and network considerations of MOST and Ethernet", Plastic Optical Fiber Newsletter, 21, 2, April 2011.

[7] J., Noebauer, "Is Ethernet the rising star for in-vehicle networks?" Invited Talk at the 16th IEEE Conference on Emerging Technologies in Factory Automation (ETFA), Toulouse, France, Sept. 2011.

[8] G., Dannhäuser, W., Franz, Dr. W., Rosenstiel, "MOST and AVB: Two Candidates for Next

Generation Automotive Infotainment Networks", Elektronik automotive, Special Issue MOST 2013.
[9] N. K. Ragesh, "Data Traffic in New Generation Vehicles", CSI Communications, 35, 12, March 2012
[10] Z.H., Dong, Z.Y. Piao; I.G. Jang; J.G., Chung, C.D. Lee, "Design of FlexRay-MOST gateway using static segments and control messages", IEEE International Symposium on Circuits and Systems (ISCAS), 2012.
[11] M.Y., Lee, S.M., Chung, H.W. Jin, "Automotive network gateway to control electronic units through MOST network", Digest of Technical Papers, International Conference on Consumer Electronics (ICCE), Las Vegas, pp 309-310, Jan. 2010.
[12] J., Joshi, S., Reiter, A., Viehl, O., Bringmann, W., Rosenstiel, "Virtual Prototypes – Evaluation Framework for MOST-based Driver Assistance Systems", Elektronik Automotive, Special Issue MOST, pp. 44-47, Apr. 2013.
[13] A., Grzemba, MOST Book, "MOST - The Automotive Multimedia Network. From MOST25 to MOST150", Ed.Franzis.
[14] Buttazzo, G.C, Lipari, G., Abeni, L., "Elastic task model for adaptive rate control," Proceedings of the 19th Real-Time Systems Symposium, 286-295, Dec 1998.