Multimedia QoS in low-cost Home Networks

H. Scholten, P. Jansen, F. Hanssen, P. Hartel, T. Hattink and V. Sundramoorthy
University of Twente, Department of Computer Science, the Netherlands

Abstract

This paper describes a new mechanism to garantee quality of service for multimedia streams in low-cost home networks. Quality of service is based on a token, of which the route in the network is determined by a distributed scheduler. The network node that has the token –the active node– can send its data during a predetermined period. The length of this period and which node gets the token next is calculated by the scheduler in the active node. Every node has a scheduler on-board and schedules streams according to stream information from other nodes –contained in the token– and its own streams. Although other types of scheduler could be used, the token scheduler deploys a preemptive earliest deadline first strategy. This guarantees a theoretical maximum bandwidth utilization of 100 percent. The network is simulated and a prototype is built, based on low-cost ethernet hardware. Results show a high throughput with a small overhead of less than one percent per stream.

1. Introduction

Currently, home systems attract much attention, both in research and industry. HAVi [1], Jini [2] and Universal Plug-and-Play [3] are examples of such systems, but there are more. They use different types of network, wireless and wireline, eg. IEEE1394, Bluetooth, ethernet, power line or telephone line. Often these networks do not offer quarenteed real-time behavior and require substantial resources in terms of power consumption, processing power and memory. This makes it expensive to incorperate small devices, like temperature or light sensors, in a home network.

In this paper we will describe work-in-progress on a network, based on low cost hardware, that supports both real-time and non-real-time traffic.

2. Real-time network

In this section we will concentrate on the network proper: RT-net. RT-net is based on the CSMA/CD ethernet protocol [4]. In effect this means that RT-net can be build on any existing ethernet specification and regular ethernet hardware can be used without modifications. CSMA/CD uses an exponential back-off mechanism to resolve collisions, which makes the network non-deterministic, and thus non-real-time. To make it deterministic, a token-based protocol is deployed to avoid collisions, just like RETHER [5] in development by the University of New York. RETHER distributes its token among the nodes in a simple static round-robin algorithm. RT-net however, uses a more sophisticated algorithm, where the token is allocated to nodes according to their bandwidth demands. A preemptive earliest deadline first (EDF) scheduler is used to determine the route a token follows in the network. This type of scheduler has the advantage over other schedulers that it can achieve a utilization of 100 percent. However, in case of overloading its performance will degrade dramatically. Whenever a node has the token –the active node– it may use the network for some time: the token hold time or THT. THT is determined by the scheduler and is likely to be different for each node. Typically the node will send one frame of a periodic multimedia stream. The EDF scheduler is distributed over the nodes and the token is the place where the schedule is kept –nodes will keep backups of the schedule though. If a node has the token and it wants to add or remove a stream, it calculates a new schedule and acts upon it. Calculating a new schedule in case of stream addition means the node does an EDF feasibility analysis to determine if a newly added real-time stream will meet its deadlines without making other streams miss theirs. EDF feasibility analysis is simple [6]:

$$U = \sum_{i=1}^{n} \frac{B_i}{B} \leq 1$$

where B_i is the network bandwidth of stream i and B is the maximum bandwidth of the given network. Actually, the maximum bandwidth is slightly less because of the overhead for token transmission and ethernet packet overhead.

When the scheduler at the active node decides that another node must become active it stores the global schedule information in the token and sends the whole package to the new node. This is a critical action and it should never occur that the token becomes lost, or worse, duplicated. When a token is lost the schedule is lost too and the network stalls. Duplicated tokens mean that more than one node will make schedules and collisions will occur, causing deadline misses and non-deterministic behaviour. To prevent this the concept of monitor is introduced. When the active node relinquishes the token, this node becomes the monitor for the new active node. Monitoring is a three step process: (1) the monitor sends the token to the new

This work is sponsored by the Netherlands Organization for Scientific Research (NWO) under grant number 612.060.111

active node. If the token is garbled the active node will request a new token. This happens only once. If a second token is garbled too, the active node stays inactive, while the monitor times-out and recovers the token. (2) The active node is now in the "transmission state" for the duration of THT. While the active node transmits, the monitor waits for a reply from the active node. It sets a timer for the duration of THT. (3) At the end of THT the active node must send a reply to the monitor signifying that it is still alive and sends the token to a new active node. The old monitor now ends its activity, the old active node becomes monitor, and the new active node begins transmitting. Then the process starts all over. Many things may go wrong. Detectable token loss situations are: token does not arrive at new active node, reply from active node is lost, active node dies before sending a reply, and the monitor dies. Undetectable token loss occurs when: the active node dies after sending a reply, and monitor and active node die simultaneously. Token duplication is difficult to detect, but there are some hints the network will give: a token arrives at a node that already holds a token, and streams miss their deadlines because nodes cannot resolve bandwidth requirements. If a node detects this, it will delete the token. The network will probably end up without token and initiates a reset.

3. RT-net simulation and prototype

To validate correctness, robustness and usability of RT-net a simulation of the network protocols is made. After that a prototype is realized for demonstration and calibration of the simulation. As simulation tool OMNeT++ [7] is used. Reasons for choosing this package are: it is open source; it uses a well-known implementation language (C++); and it provides dynamic visual feedback and statistics of the simulation. The simulation model has three layers:

- *the ethernet layer:* this layer provides basic support for ethernet packets and the CSMA/CD algorithm. Typical bandwidth is 10 or 100 Mbps. Packet loss ratio is configurable;
- *the RT-net layer:* the actual RT-net is simulated in this layer. It uses the underlying ethernet layer to send and receive packets, tokens and control messages. It provides a control interface to the application layer for passing events and for creating and deleting real-time streams. The model is decentralized: every node is simulated by a separate module and is governed by its own parameters. Only start-up configuration parameters, like number of nodes and network bandwidth) are shared between all nodes;
- *the application layer:* this layer controls the RT-net simulation and simulates the behaviour of applications. The current application layer can send the following control messages: go on-line, go off-line, and add stream. This set is far from complete, but sufficient for the simulation.

From the simulation we found the following characteristics: *the smaller the stream periods, the higher the overhead:* this is caused by the token overhead. When a period is low streams become ready more often and a token has to be sent; *the higher the offered load, the higher the overhead (with some exceptions):* this is the effect of ethernet packet overhead. If a stream uses more bandwidth, more ethernet packets are used. The exception occurs when the offered load exceeds 0.8 maximum bandwidth. Streams will be rejected and the number of running streams is lower than the number of offered streams.

Some figures calculated from the simulation are: average overhead per stream related to total bandwidth: 0.54%; average worst-case overhead per stream related to total bandwidth: 0.78%; average overhead per stream related to effective stream utilization: 1.33%; and average worst-case overhead per stream related to effective stream utilization: 1.95%. Feasibility analysis always calculates the worst-case overhead and uses that value to check that total utilization does not exceed the maximum real-time utilization. This was confirmed by the simulation, where worst-case overhead is always higher than the real overhead in every case.

A prototype is realized, based on the Linux operating system and ethernet hardware. Measurements in this prototype confirm the validity of the simulation and its parameters.

4. Conclusion

This paper describes work-in-progress on a new type of real-time network. Unlike other token-based networks RT-net uses a preemptive EDF scheduler to calculate the route of the token and thus the priorities of network communications. This garantees QoS for multimedia streams that are admitted to the network. The network is based on ethernet hardware and ethernet protocols. Because the token mechanism prevents collisions to occur, the network becomes predictable and suitable for hard real-time purposes. Preemptive EDF scheduling enables a 100 percent utilization of the network. Simulations show, confirmed by a prototype, that average overhead per stream is around 0.5 percent of the total network bandwidth. Currently, work is in progress to port RT-net to IEEE1394 (Firewire). Simulation is almost ready and prototype building starts shortly.

5. References

[1] Home Audio and Video Interoperability web site: http://www.havi.org

[2] JINI homepage: http://www.jini.org or http://www.sun.com/jini

[3] UPnP homepage: http://www.upnp.org

[4] Carrier sense multiple access with dollission detections (CSMA/CD) access method and physical layer specifications, ISO/IEC 8802-3 (4th edition), IEEE, 1993

[5] RETHER homepage: http://www.ecsl.cs.sunysb.edu/rether.html

[6] G.C. Buttazzo, *Hard real-time computing systems*, Kluwer Academic Publishers, ISDN 0-7923-9994-3, 1997

[7] OMNet++ homepage: http://www.hit.bme.hu/phd/vargaa/omnetpp.htm