SYNCHRONIZING CIRCUIT EMULATION ADAPTERS TO CONNECT UMTS/GSM BASE STATIONS OVER METROPOLITAN GIGABIT ETHERNETS

Olga Bondareva, Rainer Baumann, Ulrich Fiedler
Computer Engineering and Networks Laboratory
Swiss Federal Institute of Technology
ETH-Zentrum, Switzerland
bondareva ,baumann, fiedler@tik.ee.ethz.ch

Abstract

Connecting UMTS/GSM base stations over Metropolitan Gigabit Ethernets has a great potential of cost savings. However, packetizing base stations' signals to emulate telephone circuits over Gigabit Ethernets additionally requires synchronization between the sending and receiving circuit emulation adapter. In this paper, we present and evaluate open source implementations of synchronization algorithms that are based on the concept of estimating the minimal end-to-end delay from sender and receiver time stamps. We show that such implementations meet the ITU-T G.823 requirements for synchronization and generally outperform the synchronization algorithms employed in commercially available circuit emulation adapters.

Keywords
frequency synchronization, circuit emulation, clock skew, Metro Gigabit Ethernet

1. INTRODUCTION

Convergence of telephone and data services has drawn much attention due to augmented flexibility and the potential in cost savings. In metropolitan areas, the deployment of optical fiber and Gigabit Ethernets has lead to an expansion of packet-switched networks with cheap available capacities. These capacities are already used to transport *Voice over IP* (VoIP) traffic. Employing these capacities to connect GSM/UMTS base stations to the core *Plesiochronous Digital Hierarchy* (PDH) network creates additional revenue. However, existing base stations require an emulation of E1/T1 telephone circuits over the Ethernet to be connected (see figure 1). The base stations' E1/T1 signals need to be packetized and encapsulated in a *Circuit Emulation Adapter* (CEA) before being send over the Ethernet.
The problem with circuit emulation is the preservation of the frequency of the E1/T1 signals from across the Ethernet. E1/T1 signals are plesiochronous, i.e. their frequency is only specified within a tolerance of 50 ppm. However, the exact frequency has to be reconstructed at the receiving CEA. This reconstruction is difficult since the receiver's play-out buffer is impacted by both highly variable Ethernet dynamics and the difference in frame sending and buffer read-out frequency.

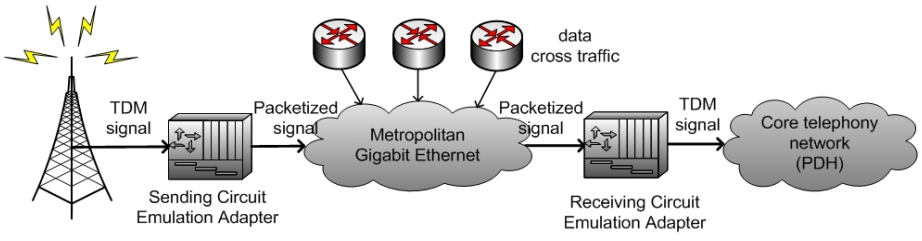

Fig. 1. – Circuit emulation over MAN Gigabit Ethernet: Connecting a base station to the core PDH network requires packetizing the base station's E1/T1 PDH signal into Ethernet frames and synchronization across the Ethernet. This synchronization is difficult since the fill state of the receiver's play-out buffer is impacted by both Ethernet dynamics and the difference in frame sending and buffer read-out frequency.

Much of the work in this field is proprietary and unpublished. However, in this paper we present and evaluate open source implementations of synchronization algorithms that are based on the concept of

estimating the minimal end-to-end delay from sender and receiver time stamps. The reason behind this estimation is its independence from the highly variable Ethernet dynamics (see figure 2). We have verified with Matlab simulation study that minimal end-to-end delay based algorithms generally perform better than well-known *Phase-Locked Loop* (PLL) algorithms. Moreover, in this paper we show that the proposed implementations meet the ITU-T G.823 requirements for synchronization and generally outperform the synchronization algorithms employed in commercially available circuit emulation adapters.

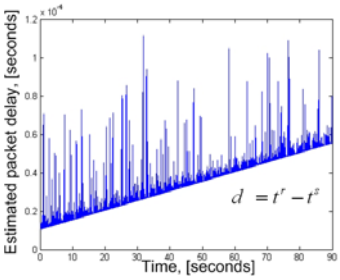

Fig. 2. – Illustration of the effect of the clock skew on end-to-end frame delay estimated from sender and receiver timestamps: minimal end-to-end delay has an increasing trend due to difference in frame sending and buffer read-out frequency.

2. PROPOSED ALGORITHMS

To obtain the trend of the minimum delay and thus to estimate the receiver clock skew, we propose to employ the Convex Hull and the Linear Adaptive Approximation algorithms. **Convex Hull** estimates the lower convex hull to the delay estimates and infers the tangent that minimizes the distance to all estimates. The frequency difference of sender and receiver clock can then directly be inferred from the slope of the tangent. Our implementation uses inner elimination to compute the lower convex hull and the Simplex algorithm to compute the tangent. It is conceptually based on [1]. **The Linear Adaptive Approximation** algorithm first de-noises measured delays and then employs a Linear Controller to infer the receiver skew form de-noised delays. The Linear Controller is adjusted with the stochastic *Normalized Least Mean Square* (NLMS) method [2].

3. EVALUATION SETUP

We have first tested our implementations in a **software framework**. The software framework does not handle PDH signals. We therefore have not implemented packetization and encapsulation. The framework solely implements a timestamping mechanism and evaluation routines to test the performance of synchronization algorithms under various frame delay patterns that model the dynamics of the Ethernet. For delay patterns we have used patterns measured in ETH's campus network [3] as well as artificial patterns. The evaluation routines compute the *Maximum Time Interval Error* (MTIE) metric between the reference sender clock and the synchronized receiver clock. This metric is widely used in telecommunication applications to evaluate the quality of the synchronization. Figure 4 a), b) depict the MTIE of our implementations compared to the requirement mask specified in ITU-T G.823. The traffic pattern used to produce this result contains no special events such as bursts. The initial receiver clock skew has been as large as $5 \cdot 10^{-5}$. We see that both implementations generally meet the ITU-T G.823 requirements. For a comprehensive discussion of results with various delay patterns refer to [4].
As a next step, we have decided to implement the Convex Hull in hardware. The hardware implementation is built on *Field-Programmable Gate Array* (FPGA) boards. Time critical paketization, encapsulation, timestamping and reassembling are performed with constant processing delay. The synchronization algorithm is implemented on a MicroBlaze software processor core. To measure the synchronization performance of this hardware implementation, we have set up a **testbed** as depicted in figure 3. This testbed consists of a PDH traffic generator, the sending and the receiving CEA connected through a network emulator and a PDH analyzer. The PDH analyzer measures the MTIE. To achieve reproducible results, we have employed RplTrc as a network emulator. RplTrc is driven by packet delay traces. As traces, we have used delay patterns measured in ETH's campus network as well as artificial delay patterns. Results (see [4]) show that the hardware CEA based on the Convex Hull generally meets the ITU-T G.823 requirements.

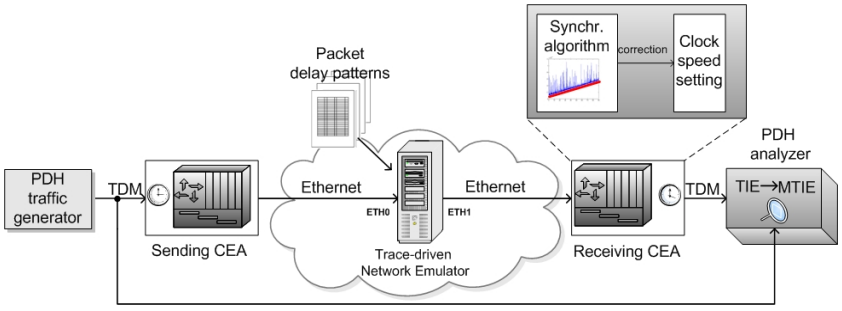

Fig. 3. – Testbed to evaluate the synchronization performance of CEAs: At the sender's side a fully-fledged hardware implementation of a CEA paketizes E1/T1 PDH signal into Ethernet frames. Ethernet frames go through a network emulator. At the receiver's side a peer CEA reassembles PDH signal and adjusts the play-out rate. The network emulator is driven by frame delay traces from real networks.

Moreover, we have compared this implementation against three other CEAs that are available on the market (product A-C). For a comprehensive discussion of results with various delay patterns refer to [4]. For space reasons we restrict to showing figure 4 c). This figure depicts the MTIE mask and evaluation results of our CEA and three commercially available CEAs for a traffic pattern without special events (bursts, etc.) and an initial receiver clock skew of $5 \cdot 10^{-5}$. From our results we conclude that hardware implementation of Convex Hull generally outperforms CEA C and also shows better performance in long-term synchronization stability than CEA B. However, CEA A shows higher quality level for short-term synchronization stability than our implementation, while long-term behaviours are comparable. This better performance of CEA A can be explained with the arithmetic limitations of the board we have used for our implementation.

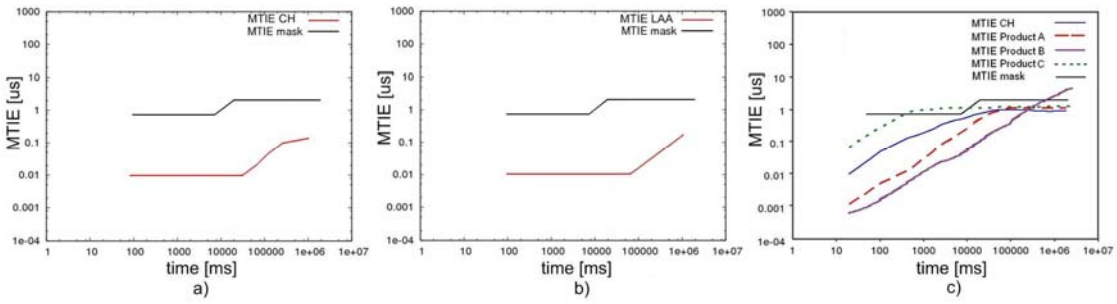

Fig. 4. – Sample results of CEA's synchronization performance in software simulation and testbed: a) MTIE for Convex Hull in software simulation; b) MTIE for Linear Adaptive Approximation in software simulation; c) Resulting MTIE for Convex Hull, Product A, Product B in the testbed.

4. CONCLUSION

We have proposed and evaluated two algorithms to synchronize circuit emulation adapters over Metropolitan Gigabit Ethernet. We have shown that our hardware implementation of the Convex Hull algorithm meets the ITU-T G.823 requirement mask for synchronization and generally outperforms commercially available CEAs.

References

[1] L. Zhang, Z. Liu, and C. H. Xia, *Clock Synchronization Algorithms for Network Measurements*, Proceedings of IEEE INFOCOM 2002.
[2] B. Widrow, S. Stirns, *Adaptive signal processing*, Englewood Cliffs N.J., Prentice-Hall, 1985.
[3] O. Bondareva, R. Baumann, U. Fiedler, *Evaluating of Novel Synchronization Algorithms for Circuit Emulation Adapters*, TIK report 240, ETH Zurich, Switzerland, 2006.
[4] R. Baumann, U. Fiedler, *Towards Connecting Base Stations over Metro Gigabit Ethernets*, Proceedings of IEEE QShine 2006.