Bandwidth-Efficient Resilience in Metro Networks - A Fast Network-Processor-Based RPR Implementation

Andreas Kirstädter, Member, IEEE, Axel Hof, Siemens AG, Corporate Technology, Information and Communication
Walter Meyer, Erwin Wolf, Siemens AG, ICN

Axel.Hof@siemens.com
Siemens AG, Corporate Technology
Information and Communications
Otto-Hahn-Ring 6
81730 München, Germany

***Abstract*—Resilience is becoming a key design issue for future packet networks. The IEEE 802.17 Resilient Packet Ring protocol has been developed to provide the necessary cost and bandwidth efficient restoration mechanisms for traffic aggregation networks in the metropolitan area. In this paper we describe our network-processor based implementation for RPR together with a bandwidth-efficient protection steering mechanism we selected. Measurement results from system integration test and a field trial prove excellent reconfiguration times well within 50 milliseconds for this solution based on reconfigurable hardware.**

***Index Terms*— High-speed LAN, Network processors, Protection, Simulation.**

I. Introduction

The economic success of single companies as well as of whole countries is more and more depending on the Internet. Further, the development of new real-time connection-oriented services like streaming technologies and mission-critical transaction-oriented services make network resilience a key issue in the design of IP based networks. Of especial importance is the restoration speed in case of network element failures.

This is not only the case in long-haul and backbone networks. Within Metropolitan Area Networks efficient packet level aggregation structures are necessary as aggregation points for current and future access networks. Here, the need for fast failure recovery meets stringent economic requirements. City and regional networks mostly consist of interconnected SDH/SONET ring structures. Although SDH/SONET theoretically may operate their own protection switching principles the corresponding mechanisms are rarely implemented in practice due to economic restrictions. Another aspect is the fact that any protection operating purely on the transport network level (SDH/SONET) will not be able to protect against failures within the packet switching infrastructure.

This situation within the metropolitan area was the starting point for the Institute of Electrical and Electronic Engineers to begin the development the Resilient Packet Ring Protocol (RPR, IEEE802.17) in December 2000 with the intention to create a new Media Access Control layer for metropolitan area networks. The very bandwidth-efficient protection steering mechanism was defined in addition to the well-known ring-wrapping mechanism for RPR.

In order to accommodate protocol changes due to the still open standard at the start of our development process and to allow a rapid product development and a short time to market for novel products we selected a network processor unit (NPU) as the basis of our RPR implementation [1].

In this paper we investigate the performance of the protection steering mechanism on the basis of this reconfigurable hardware. Since steering is known to be somewhat slower than wrapping we put especial emphasis on the reconfiguration time behavior.

After a brief description of RPR, the next sections shortly describe our RPR card and its integration into the system. This is then followed in section V by a detailed discussion of the possible resilience mechanisms in the RPR layer. Section VI discusses the function split for the different processes running in the NPU and in its controlling GPP in case of the protection steering mechanism selected. Finally, we provide some interesting results concerning the restoration speed achieved by our solution in the case of several network element failures.

II. Resilient Packet Ring Overview

The Resilient Packet Ring (RPR) is a draft standard to

transport data traffic over ring-based media with link data rates scalable up to many gigabits per second in Local or Metropolitan Area Networks.

Two counter-rotating buffer-insertion rings build up an RPR [2, 3]. Physically, adjacent nodes may be interconnected via a (fiber) link pair or transport network paths, as shown in Figure 1 .The link bit-rate of an RPR can take values in the range from 155 Mbit/s up to 10 Gbit/s [3].

Among many other deployment areas, RPR rings are therefore especially attractive for the use within SDH/SONET Add-Drop Multiplexers in Metropolitan Area networks.

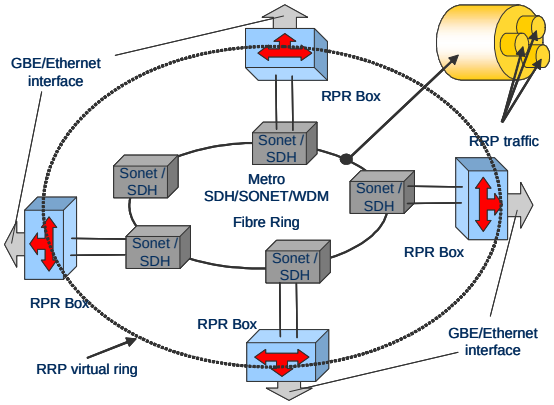

Figure 1: RPR topology on the basis of SDH links.

On the medium-access sub-layer the RPR employs a buffer insertion mechanism combined with a message-based fairness control triggered by fill levels in the single ring nodes.

III. RPR Card and SDH Add-Drop Multiplexer

The SDH/SONET add-drop multiplexer is a multi-service system that is configured in a rack with multiple flavors of line cards. A SDH/SONET back plane provides the inter-working among the cards across a switch fabric. A control processor card manages the operation of the system. The line cards run with OC-3, OC-12 and OC-48.

The RPR line card described in this paper offers on the tributary-interface side the choice between 10/100 Mbps and 1 Gbps Ethernet. On the (SDH) ring side, either VC-4 paths or VC-4-4v paths can be supported. The different alternatives are selected by downloading corresponding configurations into the network processing unit (NPU) [4].

As shown in Figure 2 the RPR card mainly consists of the NPU (C-5 from C-Port/Motorola [5]), a GPP, and some interface and memory chips. The GPP controls the network processor. It consists of a Power-PC processor connected via a PCI bridge to the network processor. The GPP takes care about the generation of the routing table, the alarm handling and bandwidth reservation. Via the PCI bridge the GPP can access a part of the data memory of the NPU, e.g. for downloading routing tables to the NPU or reading of some statistical data.

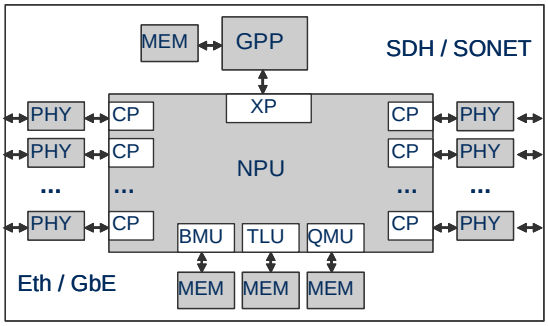

Figure 2: Architecture of the RPR line card.

IV. Network Processor Architecture

The C-5 network processor from Motorola contains 16 parallel channel processors (CP). Each of them consists of a RISC core together with a Serial Data Processor (SDP) for the bit and byte processing [5]. The RISC core controls cell and packet processing in its channel via the execution of a MIPS™ 1 instruction set (excluding multiply, divide, floating-point operations).

The Executive Processor (XP) serves as a centralized computing resource for the C-5 NPU and manages the system interfaces. Typical XP functions include:

- Chip initialization and code download from the GPP.
- Routing/Switching table maintenance (either building tables or importing updates from the GPP).
- Statistics harvesting from CPs and the TLU.
- Fault detection/recovery.
- Non-critical-path forwarding functions.

Other special-purpose units on the C-5 manage packet buffering (BMU), queuing (QMU), and table lookups (TLU), as shown in Figure 2.

The 16 parallel channel processors (CP) are ordered into four clusters of four processors each. The four processors in one cluster can run the same application and share an instruction memory of 24 kByte that also can be subdivided so that each CP gets a dedicated 6kByte sub-array.

Packet buffering and queuing in the C5 is handled as follows: The payload of the incoming packet is stored in the external memory, which is controlled by the Buffer Management Unit (BMU). The BMU controls the storage of the payload and returns a descriptor of the memory block for the payload storage to the CP. After the lookup at the Table Lookup Unit (TLU) the CP sends the descriptor of the payload buffer to the Queue Management Unit (QMU) and appends to the queue of the transmitting CP.

A library of service functions [6] exists for the communication of the code running on a CP with the QMU, BMU and TLU. The library functions, example applications, and the complete tool environment are part of the C-Ware Software Toolset (CST).

V. RPR Protection Methods

Several alternatives exist for the protection of RPR rings. A pure protection on the SDH level - below the RPR protocol - is surely the fastest way but occupies a lot of protection bandwidth and does not cover failures of the packet node or on the Ethernet level. **Figure 3** depicts the RPR traffic forwarding without protection switching. The traffic flow is send from the source node S to the destination node D.

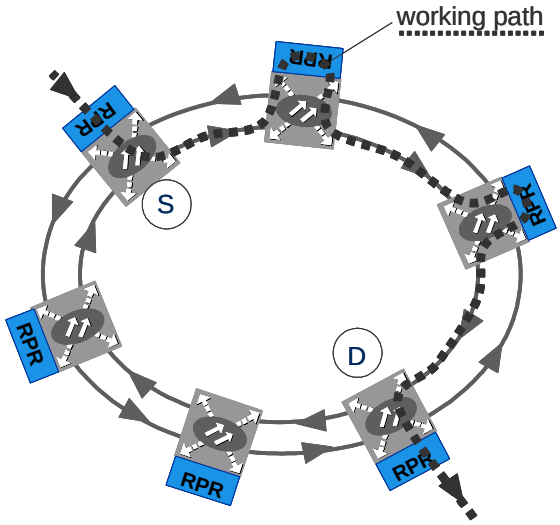

Figure 3: Original RPR traffic flow from S to D

The IEEE 802.17 protocol itself supports wrapping and steering for ring protection, the latter allowing the spatial re-use of bandwidth.

The faster alternative is wrapping - being less bandwidth effective due to the wrapping loops. Wrapping occurs locally and requires two nodes to perform protection switching. As shown in Figure 4 the two nodes neighbouring the failed span have to loop the traffic onto the other ring. The solid line symbolizes the protection path after the wrapping: From S clockwise to the wrapping node W and then back counter-clockwise to the destination D. Fast wrapping generates the lowest packet loss on the cost of higher bandwidth consumption.

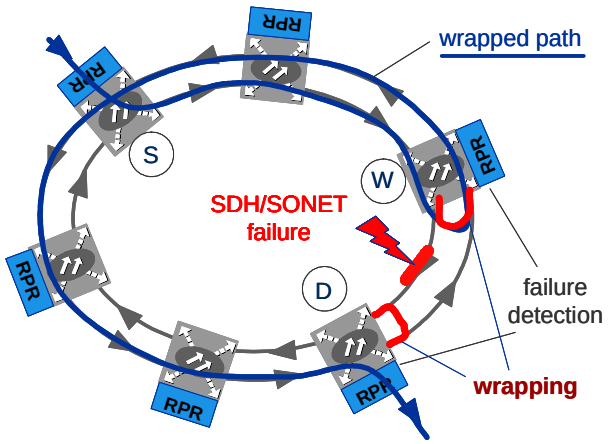

Figure 4: Wrapping for ring protection

Steering reacts to the failure by modifying the routing tables (shown in Figure 5 together with the original traffic flow) in all nodes. Therefore it is more bandwidth efficient but also slower due to the messaging and the generation of the tables.

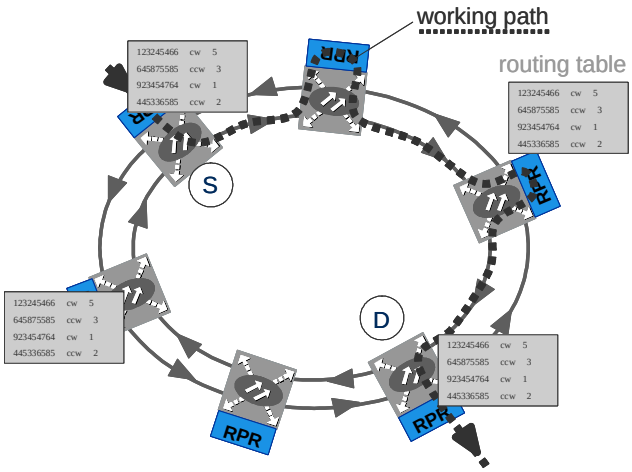

Figure 5: RPR traffic forwarding and routing tables for original flow

In the RPR layer the two neighboring nodes would signal to other nodes span-status changes via control messages carried on opposite ring. Instead of wrapping the ring each node then independently reroutes the traffic it is sourcing onto the ring using the updated topology. The resulting rerouting with the traffic going in counter-clockwise direction is shown in Figure 6: Wrapping loops are avoided.

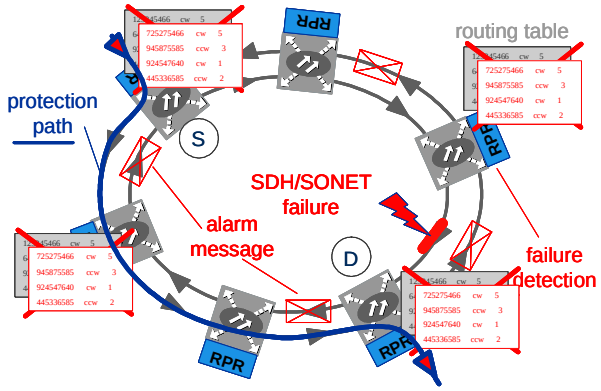

Figure 6: Steering for ring protection

The RPR foresees a recovery time of 50 milliseconds in event of fiber/node failure on the ring. Steering will be the lowest common denominator when both steering and wrapping nodes are on the ring.

In our implementation we selected a steering mechanism and implemented it in the GPP and NPU software. Link failures are detected by SDH alarming. The nodes neighboring the failure inform all other nodes via alarm messaging on the RPR level. In the GPP of each single node the routing tables are recalculated and then downloaded into the NPU as explained in the next section.

VI. Function Split between NPU and GPP

Due to the limited size of the XP instruction memory (24 kByte) the inclusion of the routing table maintenance to the XP code was not possible. Therefore, the maintenance of the routing tables became part of a GPP process. For table generation the GPP triggers the emission of a control packet via the NPU into each ring direction. These packets are addressed back to the sending node and on their way every receiving node has to add its node ID to the payload of the control packet. Thus, each node has knowledge about the whole ring topology by receiving its own topology packets. From this knowledge about the ring topology the GPP generates the routing tables and writes them via the PCI bridge into the memory of the NPU.

The alarm processing is part of the SDH framer in the NPU channel processors and the SDH physical device (see Figure 2). The SDH framer generates and verifies the SDH overhead. The SDH framer of the C-5 recognizes all errors like the AIS, PLM or UNEQ with are detected by the path trace of the SDH overhead. The physical SDH devices outside the NPU detect the other errors within the SDH frame like the LOM, LOF, LOP or LOS (see section 8).

The XP collects the detected path errors from the SDH framer and sends an alarm message to the GPP after a short integration time of 3 frames. The GPP reads the other alarms directly from the physical device.

The generation of an alarm control packet starts in the GPP immediately after the reception of an alarm signal. The GPP sends an alarm packet via the NPU on the failure-free ring direction to the other nodes. Every ring node adds its own node ID to the packet payload. Therefore, the processing is similar to the routing table generation. Every ring node receives two alarm packets from the alarm detecting nodes. With the information of the two alarm packets each node has the knowledge about the new ring topology and generates the new routing table. As soon as the NPU received the new routing table from the GPP it transmits the tributary traffic into the correct ring directions.

VII. Simulations during System Development

The C-5 tool environment includes a cycle-accurate simulator together with a performance analyzer. For the performance analysis trace points have to be inserted into the code. Time consuming code segments can be detected easily by measuring clock cycles between these trace points. For each of them the analyzer stores the cycle count and the passed trace points in a data file. Via this tool we made a first rough estimation of the workload on the network processor.

Additionally to the cycle-accurate simulations the overall system behavior had to be verified. The cycle-accurate simulations deliver a very detailed picture of the internal operation of the C5 running the RPR protocol. But for exact statements on the protocol behavior itself a separate simulator had to be developed since the system of several RPR nodes had to be observed for larger time intervals.

The system simulator was programmed in C++ using the CNCL library (Communication Networks Class Library [8]). It is event based and built in a very modular manner. As the protocol and also the simulator itself were specially adapted to the behavior of the C-5- NPU we could make exact predictions of system behavior for different load patterns, packet distributions, and system configurations.

VIII. Protection Time Measurement Results

System measurements with different SDH failures verified ring protection times well within 50 milliseconds that did not impair at all the multimedia applications we selected for payload generation. The reaction to the following SDH/SONET failure signals was investigated:

- The Loss of Signal (LOS) alarm is raised when the synchronous signal (STM-N) level drops below the threshold at which a BER of 1 in 10^3 is predicted. This could be due to a cable cut, excessive attenuation of the signal, or equipment fault. The LOS state will be cleared as soon as two consecutive framing patterns are received and no new LOS condition is detected.
- The Alarm Indication Signal (AIS) for STS-3c (AU4) is an all-ONES characteristic or adapted information signal. It is generated to replace the normal traffic signal when it contains a defect condition in order to prevent consequential downstream failures being declared or alarms being raised.
- The Loss of Multi-frame (LOM) state occurs when the incorrect H4 values for 8 frames indicate lost alignment.
- The unequipped (UNEQ) alarm is raised when a certain number of consecutive frames contain the all-ZEROS activation pattern in the unequipped overhead.

Table 1 presents the results for failure insertion and removal. In all cases of failures removal the protection switching time stays below 20 milliseconds. All error detections have no integration time to keep the delay as low as possible.

Failure	L2 Protection Switching time (msecs.)	
	Failure Insertion	Failure Removal
LOS	44	15
AU4-AIS	44	15
UNEQ	20	15
LOM	20	15

TABLE 1: RPR PROTECTION TIMES FOR DIFFERENT SDH FAULTS

The system integration was followed by a field trial at a lead customer side. Delay measurements in a 12 nodes ring in Austria (Vienna, Salzburg, Klagenfurt) made up the main part of the trial. For the delay and packet loss measurements we used frame-sizes according to RFC 2544. We observed no packet loss and delays between 6.33 and 6.98 milliseconds depending on the frame size.

IX. Conclusions

During the system development phase it was very helpful to have both the cycle-accurate and the system-level simulators at hand. Especially the system-level simulations delivered important details on the operation and optimization of protocol features that otherwise could not be verified in advance.

The system tests verified the fast protection switching with the usage of SDH alarms. This is a very important result since we used the steering principle together with a general hardware platform (NPU) instead of specialized ASICs. The generation of the rerouted tables in the GPP and the PCI transfer into the NPU showed to be sufficiently fast without the need any additional specialized hardware. Additionally, the fairness algorithm guaranties the appropriate subdivision of the link bandwidth between the single flows even in protection state.

The field trial provided us with long-time measurements and asserted the smooth system behavior of the RPR line card. Since some months the system is delivered to customers.

The next steps in our project will comprise a faster routing table generation, e.g. by speeding up the interconnection between a NPU and GPP [1] which will lead to a shorter rerouting time. Also we plan to use special HW for alarm detection. Furthermore our system will be extended to other packet services and the usage of other transport systems. Finally it will be made fully compliant to the finalized IEEE 802.17 standard.

References

[1] T. Wolf, “Design of an Instruction Set for Modular Network Processors,” IBM Research Report, RC 21865, October 27, 2000

[2] IEEE 802.17 Resilient Packet Ring Working Group Website, http://www.ieee802.org/rprsg/.

[3] H.R. van As, “Overview of the Evolving Standard IEEE 802.17 Resilient Packet Ring,” 7th European Conference on Networks & Optical Communications (NOC), Darmstadt, Germany, June 18-21, 2002.

[4] N. Shah, “Understanding Network Processors,” Master’s Thesis, Dept of Electrical Engineering and Computer Science, Univ. of California, Berkeley, 2001

[5] C-5e Network Processor Architecture Guide Silicon Revision A0, Motorola, http://e-www.motorola.com/brdata/PDFDB/docs/C5EC3EARCH-RM.pdf

[6] C-5e Application Documentation, Motorola, http://e-www.motorola.com/webapp/sps/site/prod_summary.jsp?code=C-5E#applications

[7] F.T. Hady, T. Bock, “Platform Level Support for High Throughput Edge Applications: The Twin Cities Prototype,” IEEE Network Magazin, July/August 2003

[8] RWTH Aachen, http://www.comnets.rwth-aachen.de/doc/cncl/