DDX Add-On Card: Transforming Any Optical Legacy Network into a Deterministic Infrastructure

Nihel Benzaoui(1), Guillaume Soudais(1), Olivier Angot(2), Phillipe Robineau(2), Sebastien Bigo(1).

(1) Nokia Bell Labs, Nihel_djoher.benzaoui@nokia-bell-labs.com
(2) Nokia, Route de Villejust, 91620, Nozay, France

Abstract *We propose a novel slotted, scheduled, and synchronous add-on modular card to deliver data with truly deterministic performance over legacy optical networks. We achieve ultralow 50ns jitter and 25µs latency end-to-end for edge-cloud scenarios.*

Introduction

Next-generation applications for new real-time user experiences (e.g., augmented reality) or for industry automation (e.g., industry 4.0) will benefit from processing in the edge cloud deployed for 5G networks. Part of these applications, which we refer to as time-critical, require strict deterministic guarantees from the network. They expect a high quality of service (e.g., fixed and low latency, ultralow jitter, ultra-reliability) end-to-end from the application source (e.g., a machine) to its data center processing unit (Fig. 1), regardless of the traffic load.

The typical approaches to engineer Quality of Service (QoS) in legacy optical switched networks such as traffic prioritization and packet pre-emption can, at best, achieve determinism for a single application flow, but performance dilutes as the number of flows increases. While time-critical applications typically require sub 100ms latency and sub 100ns jitter, legacy networks can hardly achieve better than a few 100ms of latency and a few 10µs of jitter per node, both accumulating proportionally with the number of hops. Recent standards proposed alternative approaches for determinism [1]-[6]. However, they have failed to capture broad support, mainly because of their complexity, of scalability limitations, and of the prohibitive cost of renewing the entire infrastructure for time-critical applications, which are predicted to be just a small fraction of the edge cloud traffic.

In this paper, we propose Deterministic Dynamic X (DDX), an add-on card capable of migrating any legacy optical switched network into a deterministic infrastructure. The DDX add-on card is dimensioned to handle only time-critical traffic, while the remaining traffic is managed exclusively by the legacy switching nodes, thereby enabling pay-as-you-grow capacity upgrades towards delivering deterministic performance. Next, we introduce the DDX concept, describe our prototype, and integrate it into a proof of concept network where we demonstrate per application, deterministic, end-to-end paths (sub 25µs latency and 50ns jitter) over multi-hop legacy networks.

DDX card concept and implementation

A DDX card should be deployed at every first and last switching nodes of a deterministic path (Fig.1), and optionally at intermediate nodes if re-aggregation is preferred for greater network efficiency. The DDX add-on card performs the following functions 1) to pre-aggregate time-critical traffic sharing a common in a deterministic way, upstream of the switching node 2) to frame/unframe time-sensitive data for deterministic transport into slotted, scheduled, and synchronous packets, compliant with standard packet (Ethernet) protocols. All switching nodes along the deterministic path are then shortcut by bridging a specific ingress and egress switching port, such that time-critical packets are isolated from any internal contention. For time-critical applications, the switching node is seen as a wire (Fig. 2). Techniques to build dedicated or shared bridge are found later.

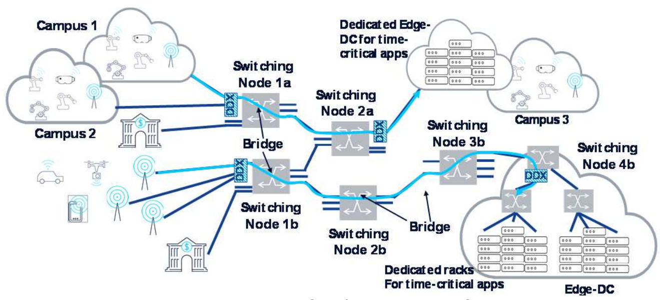

Fig. 1 DDX cards are inserted at first/last nodes of every deterministic path in an edge cloud network

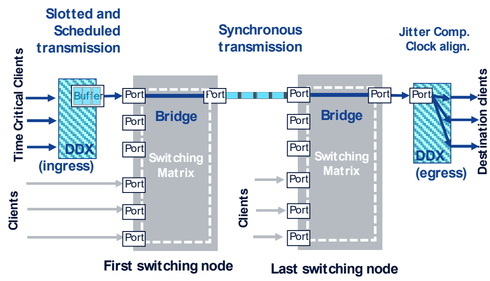

Fig. 2 Deterministic pre-aggregation and switch bridging

978-1-6654-3868-1/21/$31.00 ©2021 IEEE

Fig. 3 DDX (full/control) slot structure

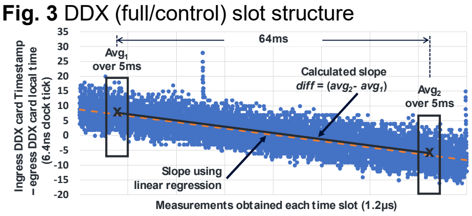

Fig. 4 Clock mismatch evaluation

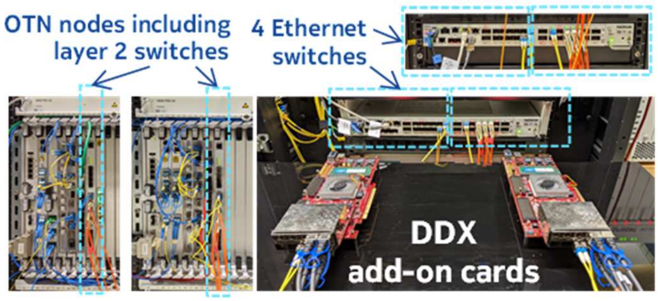

Fig. 5 experimental set-up

We implemented a DDX add-on card prototype on a HTG930 FPGA with eight client ports and two-line ports at 10Gb/s. We encapsulate the client traffic into 1.2µs (1504B) slots and used a centralized controller to schedule slots over a cyclic window of eight slots (9.6µs) (customizable parameter), so as to guarantee bandwidth for a duration just long enough to host the time-sensitive flow. At the egress DDX card, we de-capsulates the client traffic, queue it per destination and send it to the corresponding client. This part of the implementation can be considered as common to any TDM network, including to our previous work in [7]. The innovations of the DDX card reside in:

We define a ***new slot framing***. To prevent alteration of DDX slots by nodes along the deterministic path, we mimic an Ethernet frame by shaping the slot header in an Ethernet format. We hide the control information related to DDX in the payload perceived by the switching node (1504B/1480B overhead). We generate full slots only when necessary. If a slot has no traffic to carry, then only a control slot of 64B is sent (header + control info. + padding to reach minimum Ethernet frame size) as reported in Fig. 3, so that the free room between slots can be reused for non-time critical traffic, while sharing the egress port capacity of the switching node.

We use a ***periodic pacing control sequence***, instead of harder-to-achieve tight clock synchronization, as time reference, for intermediate DDX add-on cards to drop or add their own slots and reaggregate traffic. We insert the pacing sequence as part of the control information at the first ingress DDX card and propagate it all along the path.

We apply an ***original jitter compensation mechanism*** to mitigate latency variations experienced along the path. At the ingress DDX card, we count the idle symbols (20b, equivalent to a 6.7ms separation) between packets belonging to the same flow. We send these counters as part of the control information to the egress DDX card that recreates the original spacing between packets by buffering for the duration of the concatenated idle symbols.

We ***virtually re-align clocks*** of each pair of ingress/egress DDX add-on cards, as shown in Fig. 4. With stability at +/-100ppm, the 6.4ns cycles of the clocks at both ends can slightly drift and the re-creation mechanism of the spacings between packets may turn inaccurate. To correct and track the clock mismatches, we configure the ingress DDX card to send timestamps (every slot) to the egress DDX card, where we store the difference between the received timestamp and the local time and average this difference over a rolling period of 5ms. After comparing values (*diff= avg1-avg2*) every 64ms, as shown in the measurements of Fig. 4, we virtually accelerate or decelerate (idle symbols counter *+/-diff*) the counting frequency at the egress DDX card. We track and cancel any time slope, as if the clock frequencies at both ends were matching down to 0.1ppm precision (64ms= 6.4ns/0.1ppm).

Proof of concept

We now integrate our prototype DDX in a proof of concept network, where eight flows with different service level agreements in two groups, four real-time ones (1Gb/s on-off, e.g., eCPRI) with relatively relaxed performance expectations (~100ms latency) and four time-critical ones (1Gb/s constant bit rate-CBR, e.g., loop control of a robot). We reserve one slot of the cyclic schedule for each flow of the latter group. The eight flows travel into two switching nodes (1a and 2a in Fig.1) before reaching an edge data center. The nodes are typically Ethernet switches or IP routers. We use a 12x10Gb/s Ethernet switch originally set in a basic configuration without QoS management, a typical configuration where unregulated contention occurs. We generate the flows and measure latencies using a Spirent Tester exhibiting 10ns precision. We compare the latency and jitter of time critical flows with and without the DDX add-on cards, performing aggregation and adaptation of these flows to deterministic transport.

In a first scenario, we make a dedicated bridge between the switch ingress port (connected to the DDX card) and the egress port, using a Virtual Local Access Network (VLAN) marking data from the DDX card with a tag. Note that such VLAN creation is available in any packet switch/router. Only packets carrying the

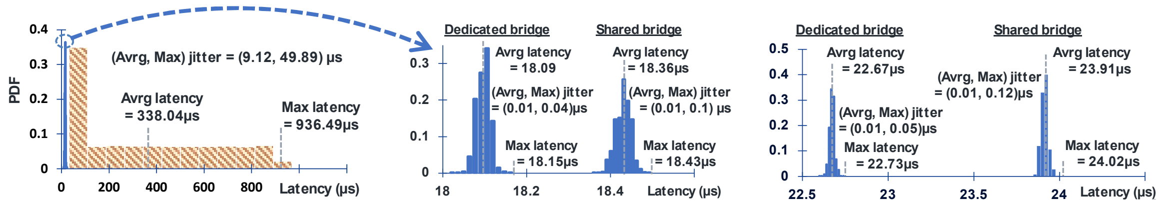

Fig. 6 Latency PDF using legacy network w/o DDX card (a), using a dedicated vs. shared bridge (b), for 2 hops and 4 hops (c)

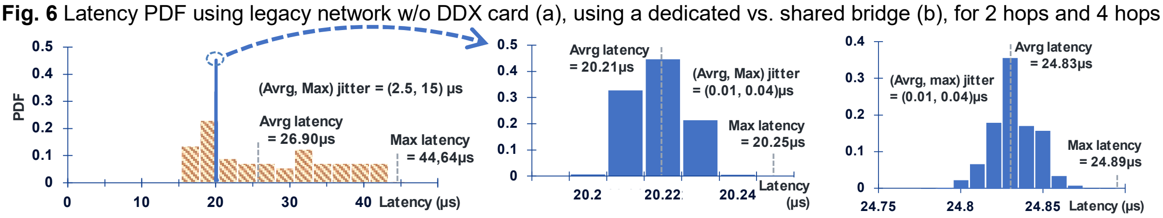

Fig. 7 Latency PDF (a) using OTN with Ethernet switch vs. with DDX card (zoom in (b)), c) for end to end (TPS+OTN)

VLAN tag are allowed to the switch egress port. The remaining non-time critical traffic travels to another egress port of the switch. In Fig. 6a, we report the probability distribution of latencies of time critical flows with and without the DDX add-on cards used for aggregation and adaptation of these flows to deterministic transport. We observe that the DDX card reduces drastically the latency (by 94% on the average and by 98% on the maximum) and its variation (by 99% on the average and maximum), when compared to legacy networking. We attribute the additional latency with the DDX card from the spreading of one packet over two reservation cycles. Overall, using the DDX card we record <20µs latency and ultralow jitter <40ns, hence exceeding the target performance for time-critical flows.

In a second scenario, we make a *shared bridge* by turning on preemption (available as an option in some switches, here a Nokia 1830TPS). While bridging the switch ingress port connected to the DDX card with the egress port, we configure the ingress port as express, so that it interrupts (pre-empts) the transmission of any other non-time critical packet that would share the same egress port. This allows for non-time-critical traffic to travel over the bridge by filling the space left (1440B) when a slot is empty, only filled with control data (see Fig. 3). Since only ~5% overhead is added by the control slots, in a limit of 95% load, no extra port is needed to carry regular traffic. Note however that packet pre-emption is not available in all legacy products.

In Fig. 6b, we compare the latency distribution of time-critical packets using a VLAN-based bridge vs. packet pre-emption-based shared bridge. Despite sharing of the egress port with non-time-critical traffic, determinism is maintained with almost unchanged 18.36µs average and 18.43µs max latency and jitter does not exceed 100ns. Note that this is obtained at the expense of a 60% increase in latency for non-time-critical traffic. When the path is extended with two more switching nodes, we measure larger latencies (Fig. 6c) but by no more than the delay of two additional store and forward operations, while jitter only slightly increases by 20ns with shared bridge and by 10ns with dedicated bridge. Determinism is preserved in a multi-hop path for both bridge types.

Legacy TDM networks such as OTN are known for deterministic data delivery, but generally rely on layer 2 switching (Ethernet) for aggregating small data streams, which results in the loss of deterministic guarantees. We propose now to replace layer 2 switches by DDX cards. To evaluate the benefits, we generate three competing flows (one 1Gb/s time-critical CBR and two 1Gb/s non-time-critical on-off flows), which we send across 2 OTN nodes, using either Ethernet switches or using deterministic pre-aggregation with the DDX card. Fig. 7a shows the measured latency distribution of time-critical packets. Compared to Ethernet, the DDX card significantly reduces latency (<20.24µs) and jitter (<40ns) (Fig 7b). Overall, DDX add-on card extends the deterministic performance of OTN to low granularity flows at the client.

In a final experiment, we assess the advantages of using DDX card through two Ethernet switches over two OTN nodes, in order to showcase an end-to-end, multi-technology proof of concept. The measurements, reported in Fig. 7c, show end-to-end latencies no higher than 25µs and jitter smaller than 40ns, largely meeting deterministic performance targets.

Conclusion

We proposed and demonstrated a new concept to allow legacy optical networks to deliver record end-to-end jitter (<100ns) and low ultra-low latency (<100µs latency excluding fiber propagation). Only two modular add-on cards at both ends of a multi-hop, multi-technology network are needed to deploy a deterministic path. DDX can be the enabler of high-value businesses employing time-critical applications.

References

[1] L. Lo Bello et al., "Schedulability analysis of Time-Sensitive Networks with scheduled traffic and preemption support," *Journal of Parallel and Distributed* Computing, vo. 144, 2020.

[2] B. Varga et. al., "DetNet Data Plane: IP over IEEE 802.1 Time Sensitive Networking (TSN)," February 19, 2021. Available: https://datatracker.ietf.org/doc/draft-ietf-detnet-ip-over-tsn/

[3] B. Varga et. al., "DetNet Data Plane: IEEE 802.1 Time Sensitive Networking over MPLS," February 19, 2021. Available: https://datatracker.ietf.org/doc/draft-ietf-detnet-mpls-over-tsn/

[4] N. Finn, "Time-sensitive and Deterministic Networking Whitepaper," IEEE Mentor User Documentation, July 2017. Available: https://mentor.ieee.org/802.24/dcn/17/ 24-17-0020-00-sgtg-contribution-time-sensitive-and-deterministic-networking-whitepaper.pdf

[5] N. Finn, P. Thubert, B. Varga, J. Farkas, "Deterministic Networking Architecture," IETF draft, Oct. 2019. Available: https://tools.ietf.org/html/rfc8655

[6] S. S. Craciunas, R. S. Oliver, M. Chmelík, W. Steiner, "Scheduling real-time communication in IEEE 802.1 Qbv time sensitive networks", in *Proc International Conference on Real-Time Networks and Systems*, 2016.

[7] N. Benzaoui et al., "Deterministic Dynamic Networks (DDN)," *Journal of Lightwave Technology*, vol. 37, no. 14, pp. 3465-3474, 15 July15, 2019.