PATN: Performance Analysis Tool for NoC

Zhonghai Lu and Yang Chen {zhonghai, yc@kth.se}
School for ICT, KTH Royal Institute of Technology, Sweden

What is PATN

With processors increased onto a single chip, and more and more time sensitive applications added to on-chip systems, performance bound analysis becomes essential for QoS Network-on-Chip (NoC) designs and evaluations.

For the purpose of providing the reliable and automated analysis for QoS NoC, we propose PATN (Performance Analysis Tool for NoC), which automatically computes the end-to-end delay bounds of data flows, and backlog bounds of buffers for NoCs with arbitrary network topology and size.

PATN is designed based on network calculus, which lies on solid mathematical foundations and provides well-guaranteed accuracy of the results. Network Calculus based analysis has been successfully employed for various communications networks, such as SpaceWire, AFDX, NoC. For example, Airbus adopted and approved the network calculus based analysis for certification on its aircraft A380.

In this demonstration, we explain the architecture and methodology of PATN. Then, we show the analysis for NoCs with arbitrary network topology and size.

Architecture and Methodology

PATN is constituted by the interface, bound analysis and network calculus layers. Fig.1 depicts the architecture and how the layers cooperate with each other. Analysis interface layer defines the analysed network and data flows. Bound analysis layer conducts flow contentions and aggregations analyses, as well as computation of the performance bounds. The basic network calculus calculations, such as min-plus convolution and deconvolution, are achieved in network calculus layer.

The analysis methodology contains the main steps of contention and aggregation analyses, as well as Equivalent Service Curve (ESC) and Aggregate Arrival Curve (AAC) computations. For example, when it computes the delay bound of a flow, the analysed network elements and data flows are initialized at interface layer. Then, in bound analysis layer it computes the end-to-end system service curve provided for the analysed flow, which is called ESC, hence to obtain the delay bound according to network calculus theory.

When it computes the ESC of the analysed flow, AACs of the contention flows are required. Moreover, AAC of a contention flow is effected by network elements that the flow has travelled. Therefore, computation of the AAC requires the ESC of the contention flow in the travel. PATN solves this nested computation with well designed contention and aggregation analysis.

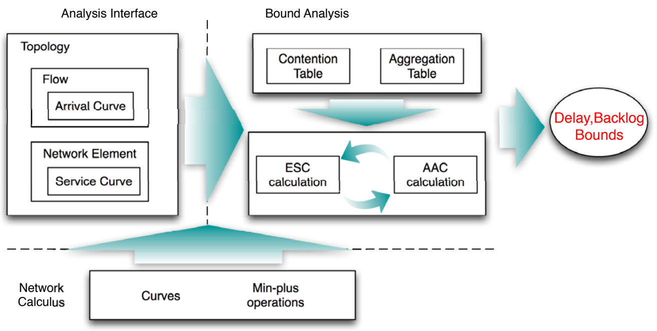

Fig.1 Architecture of PATN

Network Calculus

Network calculus is a theory for analysing performance guarantees, such as delay bounds, backlog bounds, etc., in communication networks. As core of network calculus, it defines an **Arrival Curve** to upper-bound the cumulative data traffic flow; and defines a **Service Curve** to lower-bound the cumulative service behaviour of a system (e.g. a router) processing flows. Then, the delay bound and backlog bound are obtained by computing the maximum horizontal and vertical deviation between the arrival curve and service curve, respectively.

Analysing for Various NoCs

First, we show that PATN analyses for 3 NoCs with different topologies — binary tree, mesh, and hierarchical topology of binary tree and mesh. For each NoC, computations of the per-flow delay bound and per-buffer bound are shown and explained .

Second, we show that PATN analyses NoCs with large network size. It computes the delay bound of the corner-to-corner flow in 5 n x n (n = 10, 20, 30, 40, 50) mesh NoCs with the all-to-one communications. Fig.2 depicts the comparison of delay bounds, total flow number and calculation time with the changing of the network size. It shows that PATN achieves efficient analysis for large NoCs. Even in the 50 x 50 network where 2500 flows are transmitted, the computation finishes in 5 minutes on an ordinary-capacity computer.

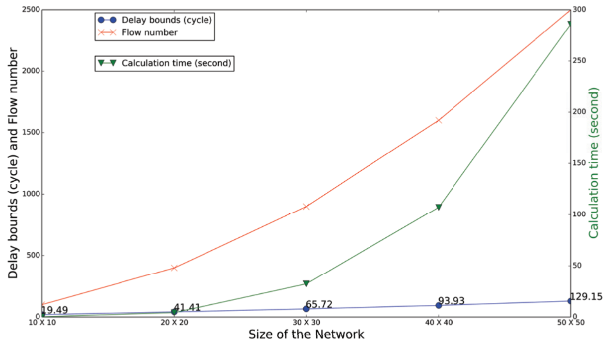

Fig.2 Comparison of delay bounds, flow number and calculation time with the changing of network size in mesh NoCs