Chapter 6
Supporting Multiple Applications

In the previous three chapters, we presented approaches to design the NoC to match the traffic requirements of an application. As technology advances, it becomes cost-effective to integrate several different applications or use-cases onto a single SoC chip. As an example, the PNX8550 (Viper2) set-top box SoC based on the Philips Nexperia platform has multiple resolution video processing capabilities (like high definition, standard definition), multiple picture modes (like split-screen, picture-in-picture), video recording features, high speed internet access, file transfer services, etc.

Current state-of-the-art SoCs also allow several of the use-cases to run in parallel. As an example, in a set-top box SoC, video display, and recording applications can run in parallel, where the recorder could potentially record a different program than what is being displayed on the screen. We refer to such use-cases that run in parallel as compound modes (Figure 6.1). The transition between the single use-case mode to compound mode needs to be smooth. As an example, when we start a new function such as video-recording in a set-top box, the video display that is currently going on should be unaffected. However, when there is a switching between compound modes, there can be a configuration time overhead to load the new set of use-cases, as shown in Figure 6.1. As the different use-cases have different functionalities, the communication characteristics can be very different across the use-cases. As an example, in Figure 6.2, a small fragment of the communication constraints for two different use-cases for the Viper2 set-top box SoC is presented, where the bandwidth requirements for some of the traffic streams for the use-cases are different.

In this chapter, we extend the synthesis approach to design NoCs that support multiple applications. To show the generality of the methods presented in this and preceding chapters, we apply the synthesis procedure to a different NoC design: the Ætheral architecture [30]. We integrate the synthesis tool with the Æthereal design flow [96], similar to the integration into the Netchip flow presented in last chapter.

The proposed synthesis process performs mapping, path selection, and resource reservation in the NoC that satisfies the communication constraints of multiple use-cases of the SoC. We consider compound modes, where two or more use-cases run in parallel, and automatically compute the communication constraints for such modes from the constituent use-cases. When there is switching between the use-cases that are run, there is a possibility of changing the paths and resource reservations in the NoC across the use-cases. The dynamic network reconfiguration can be applied when the use-case switching times are large and it helps in reducing the operating

---

1We would like to acknowledge the contributions of Dr. Andrei Radulescu, Martijn Coenen and Dr. Kees Goossens.
frequency and power consumption of the NoC. In the methodology, we preprocess the use-cases and identify the set of use-cases that need to share the same NoC configuration and use-case switching where the NoC configuration can be changed. We also explore the effect of dynamic voltage and frequency scaling (DVS/DFS) techniques for reducing the power consumption of the network across the different use-cases. We apply the methods to several SoC designs (set-top box, TV processor SoCs) and synthetic benchmarks to validate the design methodology. The methods are scalable to a large number of use-cases and are applicable even when the use-cases have very different communication characteristics.

### 6.1 The Æthereal NoC Architecture

In this section, we present the architecture of the Æthereal NoC, which provides support for predictable communication behavior and the mechanism for dynamic NoC configuration.