THE FOLLOWING EXAMPLE AND DISCUSSION WAS ADDED TO SECTION 5 TO ADDRESS A COMMENT BY REVIEWER 3. THE EXAMPLE WAS DELETED TO RESPOND TO THE STRICT PAGE RESTRICTIONS.

Example 4: Comparison to SBB calculus.

The next example compares network calculus results from this paper with the *stochastic bounded burstiness* analysis developed in [S00], referred to as *SBB calculus*. In the SBB calculus, arrival bounds take the form

$$Pr\Big\{A(t+\tau) - A(t) \geq \rho\tau + \sigma\Big\} \leq f(\sigma)$$

where $f(\sigma)$ is a function such that the n-fold integration of f, denoted by $(\int_\sigma^\infty du)^n f(u)$, is finite. Arrival models in this class include the FBM traffic model. In the network calculus, the effective envelope for SBB arrivals of a flow are given by $\mathcal{G}_i^\varepsilon(t) = t + \sigma(\varepsilon)$, where $\sigma(\varepsilon)$ is obtained by solving $f(\sigma) = \varepsilon$.

The analysis in [S00] considers a single-node work-conserving system, and derives bounds on backlog and output burstiness. The following example uses parameters from an example in ([S00], Sec. IV.C). As in [S00], we consider a single-node system with service rate C, where all flows have a rate $\rho = 1$ and a burstiness bound of $f_i(\sigma) = e^{-2.197\sigma} + 10^{-4} \cdot e^{-0.543\sigma}$. We modify the example from [S00] in that we consider a node with capacity $C = 6$ and with five flows, indexed $i = 1, \cdots, 5$. For any work-conserving service discipline, the backlog bound is computed with Theorem 3 in [S00].

Table 1: Comparison of bounds for the aggregate backlog.

$\varepsilon = 10^{-3}$		$\varepsilon = 10^{-6}$		$\varepsilon = 10^{-9}$	
THIS PAPER	SBB	THIS PAPER	SBB	THIS PAPER	SBB
30.2	20.5	100.4	66.8	168.5	130.4

We first consider the aggregate backlog. In Table 1, we compare the aggregate backlog from all flows, as obtained from Theorem 2 in [S00] with those obtained with our Theorem 1 and Lemma 1 for various values of the violation probability ε. The table shows that the SBB calculus provides tighter backlog bounds for the aggregate. The reason is that the derivation for the total backlog in the SBB calculus are done in a single estimate, whereas the network calculus makes one estimate for the busy period and another estimate for the backlog bound in Theorem 1.

The advantages of the network calculus approach become evident when we investigate the backlog of individual flows. Here, we obtain an effective service curve for a flow using the leftover service curves from Section 4. The resulting service curves are functions of the form $S_i^\varepsilon(t) = \Big[R \cdot t - X(\varepsilon)\Big]_+$, where R and $X(\varepsilon)$ are obtained from the SBB characteristics of the other flows. The backlog bound for a flow leaving the system is given by $b(\varepsilon) = \mathcal{G}^\varepsilon \oslash S_i^\varepsilon(0)$ following our Theorem 1.2. We analyze backlog bounds for all scheduling algorithms considered in this paper. For the SP service discipline, we assign flow i a priority index i. For GPS, we set the weight parameter equal at each node. For EDF, we set the flow-i delay index equal to i.

Table 2: Backlog bounds for individual flows.

	$\varepsilon = 10^{-3}$			$\varepsilon = 10^{-6}$			$\varepsilon = 10^{-9}$		
FLOW ID	SP	GPS	EDF	SP	GPS	EDF	SP	GPS	EDF
1	3.72	6.06	8.19	12.97	20.06	50.2	27.21	33.7	91.1
2	5.09	6.06	12.19	18.24	20.06	54.2	34.91	33.7	95.1
3	7.07	6.06	15.19	25.09	20.06	57.2	45.75	33.7	98.1
4	10.37	6.06	17.19	36.38	20.06	59.2	63.84	33.7	100.1
5	18.19	6.06	18.19	60.2	20.06	60.2	101.1	33.7	101.1

Table 2 shows the results of the backlog analysis. A comparison of the per-flow backlog bounds in Table 2 with the backlog bounds for the total traffic indicate that the per-flow bounds are much improved. In particular, note that with SP scheduling the backlog bounds all flows, including that for the lowest priority flow (Flow 5), are below the aggregate backlog bounds from Table 1. This demonstrates that the service description in Lemma 3 captures properties of the particular scheduling algorithm.

Remarks: The SBB calculus in [S00] does not offer delay bounds or multi-node results, and has not been developed for non-FIFO scheduling algorithms. While it may be feasible to extend the SBB calculus framework to consider per-flow bounds in various scheduling algorithms, and derive delay bounds, such derivations will require a similar effort as the derivations in a min-plus algebra as done in this paper. It remains open whether the statistical calculus can be strengthened to a degree that it yields backlog bounds that are identical to those of the SBB calculus (from Table 1). For a subclass of so-called *exponentially bounded burstiness* (EBB) [Y93] the question has recently been answered in [Ciucu05], which showed that a statistical network calculus can faithfully reproduce single node results of the EBB calculus. For a multi-node setting, a comparison of end-to-end performance bounds computed with the techniques from [Y93] to those obtained with the statistical network calculus showed that delay bounds from [Y93] scale with $\mathcal{O}(H^3)$, where H is the number of nodes, whereas the corresponding results in the statistical network calculus are bounded by $\mathcal{O}(H \log H)$.