# HopliteRT Source Queuing Bound Correction

Ian Elmor Lang
ielmorlang@uwaterloo.ca

Rodolfo Pellizzoni
rpellizz@uwaterloo.ca

Nachiket Kapre
nachiket@uwaterloo.ca

***Abstract*—We present a correction to the analytical source queuing bound for HopliteRT [1], [2], which addresses the counter-example put forward in Section IV-D of [3] by taking the effect of the in-flight jitter suffered by data flits into account. We reproduce the evaluation experiments from [1], [2] related to source queuing with this corrected approach, observing bounds that are $1.2\times$ to $1.7\times$ larger than those originally reported.**

## I. Background

The HopliteRT NoC architecture includes a token bucket regulator present at the injection point of each client, which ensures the number of flits of a flow $f$ injected over a time interval $t$ is upper-bounded by the arrival curve $\lambda_f(t)$:

$$\lambda_f(t) = \lambda^{\sigma_f,\rho_f}(t) = \min(t, \sigma_f + \lfloor \rho_f \cdot (t-1) \rfloor) \quad (1)$$

Let $f$ be the flow under analysis in a HopliteRT NoC. The source-queuing analysis presented in [2] gives an upper bound $T^s$ for the injection of a sequence of flits of $f$ as:

$$T^s = \left\lceil \frac{\sigma\left(\Gamma_f^C\right)}{1-\rho\left(\Gamma_f^C\right)} \right\rceil \quad (2)$$

For the first flit and:

$$\max\left(\frac{1}{\rho_f}, \frac{1}{1-\rho\left(\Gamma_f^C\right)}\right) \quad (3)$$

For each subsequent flit, as long as the length of the sequence is smaller than or equal to the burstiness $\sigma_f$ of the flow $f$.

The parameters $\sigma\left(\Gamma_f^C\right)$ and $\rho\left(\Gamma_f^C\right)$ correspond to an arrival curve that upper-bounds the arrival, at the link where $f$ is injected, of flits from of all flows belonging to the set $\Gamma_f^C$ of flows that may interfere with $f$. The analysis in [2] calculates $\sigma\left(\Gamma_f^C\right)$ and $\rho\left(\Gamma_f^C\right)$ in terms of the arrival curves enforced by regulators at the injection points of the flows $g \in \Gamma_f^C$:

$$\sigma\left(\Gamma_f^C\right) = \sum_{g\in\Gamma_f^C} \sigma_g \quad (4)$$

$$\rho\left(\Gamma_f^C\right) = \sum_{g\in\Gamma_f^C} \rho_g \quad (5)$$

This approach to the calculation of $\sigma\left(\Gamma_f^C\right)$ and $\rho\left(\Gamma_f^C\right)$ corresponds to an assumption that the arrival curve for the injection of a flow $g \in \Gamma_f^C$ is also an upper-bound for the arrival of flits from $g$ at the link where $f$ is injected.

## II. In-flight Jitter

As pointed out in [3], the assumption described at the end of the previous section does not necessarily hold. This is due to the in-flight jitter $J_{gf}$ on the time it takes for a flit of $g$ to travel from its injection point to injection point of $f$. This jitter is introduced by deflections, which may not occur uniformly to all flits of $g$.

As a result of the in-flight jitter, a sequence of flits that were not sent consecutively from the source of $g$ may arrive consecutively at the link where $f$ is injected, effectively increasing the burstiness of $g$ experienced at said link. The counter-example from Section IV-D of [3] is a construction of one such situation.

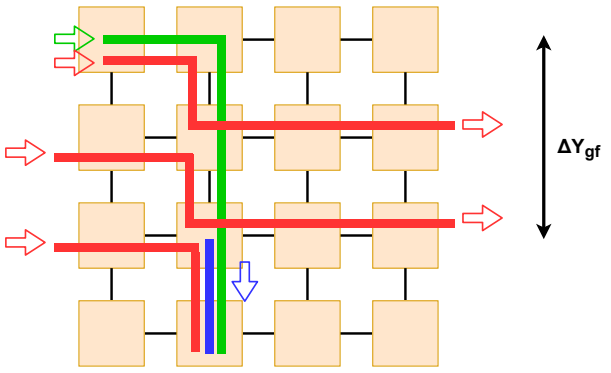
In particular, the bounds for the in-flight latency shown in [1], as well a the method for constructing the set of interfering flows $\Gamma_f^C$, remain unchanged, as they do not depend on the burstiness of interfering flows.

The value of $J_{gf}$ depends on the relative positions of the injection points for flows $f$ and $g$:

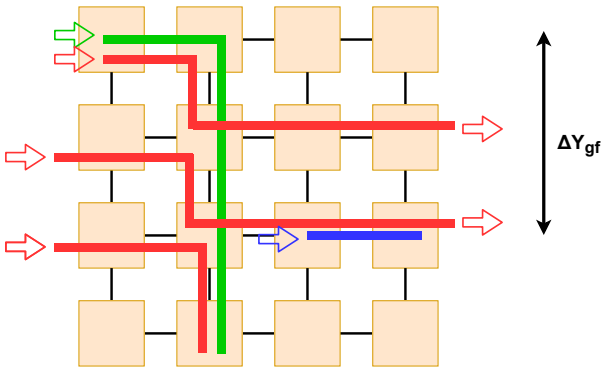
- If the router where $g$ is injected is in the same row as the router where $f$ is injected, $J_{gf} = 0$, regardless of whether $f$ is injected East or South. This is because, if both are in the same row, a flit from $g$ will only ever travel horizontally, with maximum priority, up until it interferes with $f$.
- Otherwise, if $f$ is injected South, $J_{gf} = n_{def}(g,f) \cdot W$, where $n_{def}(g,f)$, is the number of deflections that a flit of $g$ might suffer before interfering with $f$, and $W$ is the NoC width. Each possible deflection adds $W$ time units to the jitter, as that is the extra time it may add to the trajectory of a a flit of $g$ through the network.
- In the remaining case (where $g$ is injected East) $J_{gf} = (n_{def}(g,f) - 1) \cdot W$. We count one deflection less than the previous case because the deflection of $g$ in the row where $f$ is injected must necessarily happen if $g$ is to interfere with $f$. Because this deflection must necessarily happen if there is interference, it cannot be part of the variability in the time it takes a flit of $g$ to reach the injection point of $f$ (the in-flight jitter).

Figure 1 illustrates the second and third cases described above: the green line shows the path of flow $g$ without deflections, the red line shows the path of $g$ with maximum deflections, and the blue line shows the injection of $f$. Note how, in the case with $f$ injected East, the interference only happens if the second deflection of $g$ occurs.

The figure also shows $\Delta Y_{gf}$, the number of links that must be traversed south to move from the row where $g$ is injected to the row where $f$ is injected. This quantity can be used an upper bound for $n_{def}(g, f)$, as a flit can only suffer up to one deflection for each link travelled south.




(a) Flow $f$ injected South




(b) Flow $f$ injected East

Fig. 1: Illustration of scenarios for calculating the in-flight jitter $J_{gf}$.

## III. Correction to Source Queuing Analysis

Before modifying the source queuing analysis, we first adopt a new definition for the arrival curve $\lambda_f(t)$:

$$\begin{aligned}\lambda_f(t) &= \min(t, \sigma_f + \rho_f \cdot (t-1)) \\ &\geq \min(t, \sigma_f + \lfloor \rho_f \cdot (t-1) \rfloor)\end{aligned} \tag{6}$$

This new curve is greater than or equal to the curve enforced by the token bucket regulator, so we may adopt it without being optimistic. Appendix A reproduces the demonstrations from [2] for this arrival curve, showing that it also admits Equation 2 and Equation 3 as bounds for the source queuing of flows.

The motivation for the change in the arrival curve formula is that it allows for a less pessimistic expression for source-queuing bounds once the in-flight jitter has been incorporated into the analysis.

To produce an arrival curve for $g \in \Gamma_f^C$ at the injection point of $f$, accounting for the in-flight jitter, we displace the injection arrival curve for $g$ in time:

$$\begin{aligned}\lambda_{gf}(t) &= \min(t, \sigma_j + \rho_g \cdot (J_{gf} + t - 1)) \\ &= \min(t, \sigma_j + \rho_g \cdot J_{gf} + \rho_g \cdot (t-1))\end{aligned} \tag{7}$$

We define $\sigma_{gf}$ as:

$$\sigma_{gf} = \sigma_g + J_{gf} \cdot \rho_g \tag{8}$$

to rewrite $\lambda_{gf}(t)$ as:

$$\lambda_{gf}(t) = \min(t, \sigma_{gf} + \rho_g \cdot (t-1)) \tag{9}$$

which is in the same format as (6), just with a higher burstiness constant, so the rest of the reasoning from Appendix A still applies. We now just need to calculate $\sigma\left(\Gamma_f^C\right)$ as:

$$\sigma(\Gamma_i^C) = \sum_{g \in \Gamma_f^C} \sigma_{gf} \tag{10}$$

Appendix C applies Equations 8-10 to the counter example to the original analysis presented in [3], to obtain a safe upper bound for the source queuing of the flow under consideration.

## IV. Evaluation

The plots in Figure 2 show a comparison between the original and the corrected bounds on the source queuing latency, using the RANDOM and ALL2ONE communication patterns and a burstiness of 1. The corrected source queuing upper bound for these test cases is $1.21\times$ to $1.75\times$ larger than the corresponding bound originally presented in [1].

We also consider the effect of increasing the burstiness parameter. The plots in Figure 3 shows a comparison between the original and the corrected bounds on a $4 \times 4$ HopliteRT NoC, for burstiness values ranging from 1 to 10. As the burstiness increases, the effect of the correction represents a diminishing fraction of the total worst-case queuing latency.

## Acknowledgments

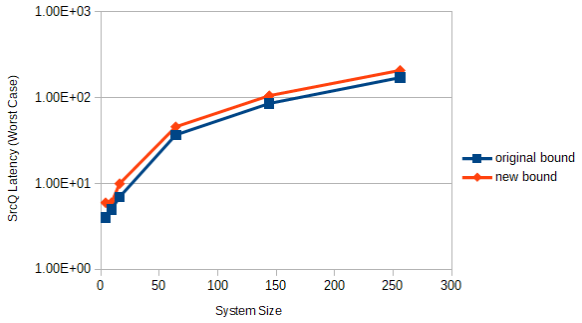
We would like to thanks Yilian Ribot and Geoffrey Nelissen for identifying the counter example to the original HopliteRT source-queuing analysis, which motivated this work.

## References

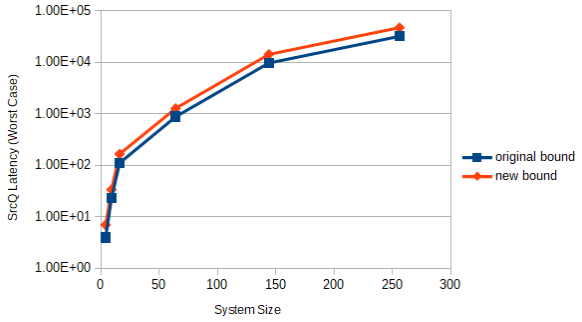
[1] S. Wasly, R. Pellizzoni, and N. Kapre, "Hoplitert: An efficient fpga noc for real-time applications," in *2017 International Conference on Field Programmable Technology (ICFPT)*, 2017, pp. 64–71.

[2] Wasly, Saud, Pellizzoni, Rodolfo, and Kapre, Nachiket, "Worst case latency analysis for hoplite fpga-based noc," 2017. [Online]. Available: http://hdl.handle.net/10012/12600

[3] Y. Gonzalez and G. Nelissen, "Hoplitert*: Real-time noc for fpga," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, pp. 1–1, 11 2020.

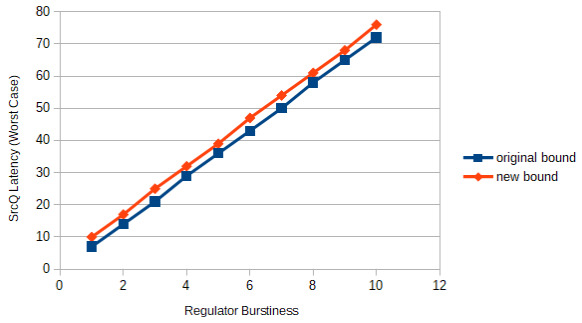



(a) RANDOM

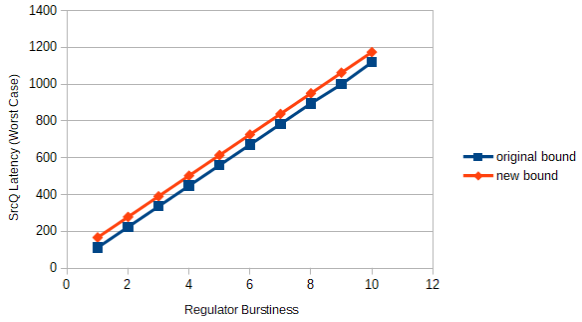



(b) ALL2ONE

Fig. 2: Comparison of original and corrected bound on worst-case source queuing latency in HopliteRT for various system sizes




(a) RANDOM




(b) ALL2ONE

Fig. 3: Comparison of original and corrected bound on worst-case source queuing latency in HopliteRT for various burstiness values.

## APPENDIX A
## DERIVATION OF SOURCE QUEUING BOUNDS

This appendix adapts the demonstrations in Appendix C - Derivation of Delay Bounds of [2] to the arrival curve formulation in Equation 6.

We first define the operation $\oplus$, which produces an arrival curve for a router output port in terms of the router's inputs:

**Lemma A.1.** *Let $\lambda^{\sigma_1,\rho_1}$ and $\lambda^{\sigma_2,\rho_2}$ bound the traffic on input ports (West, North or PE) directed to the same output port (East or South). Then the traffic on the output port is bounded by the following curve:*

$$(\lambda^{\sigma_1,\rho_1} \oplus \lambda^{\sigma_2,\rho_2})(t) = \min(t, \sigma_1+\sigma_2+(\rho_1+\rho_2)\cdot(t-1)) \quad (11)$$

*Proof.* In any time window of length t, the number of packets transmitted on an output port cannot be greater than the traffic produced by the input ports; hence it holds:

$$(\lambda^{\sigma_1,\rho_1} \oplus \lambda^{\sigma_2,\rho_2})(t) \leq \sigma_1+\sigma_2+\rho_1\cdot(t-1)+\rho_2\cdot(t-1) \quad (12)$$

Furthermore, since the number of packets cannot be larger than t, it also holds that $(\lambda^{\sigma_1,\rho_1} \oplus \lambda^{\sigma_2,\rho_2})(t) \leq t$. Equation 11 then immediately follows. □

The number of free cycles that $f$ has to inject packets into the network, over a time interval $t$, is then lower-bounded by $t - \oplus\Gamma_f^C(t)$. $\oplus\Gamma_f^C(t)$ is the combination of all other flows that enter the router where $f$ is injected and leave it through the same link as $f$.

**Lemma A.2.** *Flow $f$ cannot suffer starvation if $\rho\left(\Gamma_f^C\right) < 1$.*

*Proof.* By expanding the expression for the guaranteed number of free injection cycles $t - \oplus\Gamma_f^C(t)$ we obtain:

$$\begin{aligned} t - \oplus\Gamma_f^C(t) &= t - \min\left(t, \sigma\left(\Gamma_f^C\right) + \sum_{\forall\lambda^{\sigma,\rho}\in\Gamma_f^C} \rho\cdot(t-1)\right) \\ &= \max\left(0, t - \sigma\left(\Gamma_f^C\right) - \sum_{\forall\lambda^{\sigma,\rho}\in\Gamma_f^C} \rho\cdot(t-1)\right) \\ &= \max\left(0, t - \sigma\left(\Gamma_f^C\right) - \rho\left(\Gamma_f^C\right)\cdot(t-1)\right) \\ &= \max\left(0, t\cdot\left(1-\rho\left(\Gamma_f^C\right)\right) - \left(\sigma\left(\Gamma_f^C\right) - \rho\left(\Gamma_f^C\right)\right)\right) \end{aligned} \quad (13)$$

We now note that $\rho\left(\Gamma_f^C\right) < 1$ implies that $1-\rho\left(\Gamma_f^C\right) > 0$; thus, the number of guaranteed free slots increases with $t$, which means that the flow cannot be starved.

□

**Lemma A.3.** *If $\rho(\Gamma_f^C) < 1$, then:*

$$t - \oplus\Gamma_f^C(t) \geq \max(0, \lfloor (t-(T^s+1)) \cdot (1-\rho\left(\Gamma_f^C\right)) \rfloor + 1) \quad (14)$$

*with $T^s$ as defined in Equation 2.*

*Proof.* The lemma follows directly by algebraic manipulation, where the last equality is based on Equation 13.

$$\begin{aligned}
&\max(0, \lfloor (t-(T^s+1)) \cdot (1-\rho\left(\Gamma_f^C\right)) \rfloor + 1) \\
&\leq \max(0, (t-(T^s+1)) \cdot (1-\rho\left(\Gamma_f^C\right)) + 1) \\
&= \max(0, t \cdot (1-\rho\left(\Gamma_f^C\right)) - (T^s+1) \cdot (1-\rho\left(\Gamma_f^C\right)) + 1) \\
&= \max(0, t \cdot (1-\rho\left(\Gamma_f^C\right)) - \left( \left\lceil \frac{\sigma\left(\Gamma_f^C\right)}{1-\rho\left(\Gamma_f^C\right)} \right\rceil + 1 \right) \\
&\cdot (1-\rho\left(\Gamma_f^C\right)) + 1) \\
&\leq \max(0, t \cdot (1-\rho\left(\Gamma_f^C\right)) - \left( \frac{\sigma\left(\Gamma_f^C\right)}{1-\rho\left(\Gamma_f^C\right)} + 1 \right) \\
&\cdot (1-\rho\left(\Gamma_f^C\right)) + 1) \\
&= \max\left(0, t \cdot \left(1-\rho\left(\Gamma_f^C\right)\right) - \left(\sigma\left(\Gamma_f^C\right) - \rho\left(\Gamma_f^C\right)\right)\right) \\
&= t - \oplus\Gamma_f^C(t)
\end{aligned} \quad (15)$$

□

**Theorem A.4.** *Assume $\rho(\Gamma_f^C) < 1$ and the client wishes to inject a sequence of $k \leq \sigma_f$ packets for flow $f$. The delay to inject all packets in the sequence is then upper bounded by:*

$$\left\lceil \frac{1}{\rho_f} \right\rceil - 1 + T^s + \left\lceil (k-1) \cdot \max\left( \frac{1}{\rho_f}, \frac{1}{1-\rho\left(\Gamma_f^C\right)} \right) \right\rceil \quad (16)$$

*Proof.* In the worst case, the token bucket for $f$ can be initially empty for at most $\lceil 1/\rho_f \rceil - 1$ clock cycles. Afterwards, a new token is added to the bucket every $1/\rho_f$ cycles, at which point the next packet in the sequence becomes ready to be injected once the NoC port is free. Note that, since $k \leq \sigma_f$, the times at which the first $k$ tokens are added, and thus the packets in the sequence become ready at the regulator, do not depend on the time at which the packets themselves are sent; this is because the bucket does not become full until the $k$-th token is added.

Now consider the effects of conflicting NoC traffic. Let $\lambda^{free}(t) = \max(0, \lfloor (t-(T^s+1)) \cdot (1-\rho\left(\Gamma_f^C\right)) \rfloor + 1)$, and consider any subsequence of $i$ packets out of the sequence of $k$ packets under analysis which are being delayed by NoC traffic. Since the time at which the packets become ready is fixed, the delay suffered by the last packet in the subsequence cannot be larger than both $\lceil (i-1) \cdot (1/\rho_f) \rceil$ and $\bar{t}$, where $\bar{t}$ is the minimum window length for which $\lambda^{free}(\bar{t}) = i$ (that is, the time that it takes for the NoC to have $i$ free cycles based on Lemma A.3). Based on the expression for $\lambda^{free}$, it is then trivial to see that if $1/\rho_f \geq 1/1-\rho\left(\Gamma_f^C\right)$, the worst case delay for the sequence is found when the first $(k-1)$ packets are sent as soon as they become ready at the regulator, while the last packet suffers NoC delay of $T^s$; whereas if $1/\rho_f < 1/(1-\rho\left(\Gamma_f^C\right))$, the worst case is found where all $k$ packets are delayed by NoC traffic rather than regulation. Combining the two cases yields Equation 16. □

Theorem A.4 only holds for sequences of packets of length at most equal to the burst length of the flow. If the sequence is longer than the burst length, then the token buffer might become full during a window of length $T^s$ when the NoC port is blocked, at which point the time when further tokens are added is delayed based on when packets in the sequence are sent.

## APPENDIX B
## SUMMARY OF FORMULAS CHANGES

Table I shows a summary of formulas changes due to the source queuing analysis correction.

| | **original** | **corrected** |
|---|---|---|
| $T^s$ | $\left\lceil \frac{\sigma\left(\Gamma_f^C\right)}{1-\rho\left(\Gamma_f^C\right)} \right\rceil$ | unchanged |
| $\rho\left(\Gamma_f^C\right)$ | $\sum_{g \in \Gamma_f^C} \rho_g$ | unchanged |
| $\sigma\left(\Gamma_f^C\right)$ | $\sum_{g \in \Gamma_f^C} \sigma_g$ | $\sum_{g \in \Gamma_f^C} \sigma_{gf}$ |
| $\sigma_{gf}$ | N/A | $\sigma_g + J_{gf} \cdot \rho_g$ |
| $J_{gf}$ | N/A | Described in Section II |

TABLE I: Summary of formulas changes

## APPENDIX C
## IN-FLIGHT JITTER EXAMPLE

The counter-example to the original HopliteRT source-queuing analysis presented in [3] corresponds to the four flows depicted in Figure 4. We wish to find an upper bound $T_f^s$ on the time flow $f = f_4$ must wait to inject the first packet of a sequence into the network.

Flow $f_4$ may suffer interference from flow $g = f_1$, which has regulator parameters $\sigma_1 = 0.5, \rho_1 = 3$. The counter-example consists of the following sequence of events:

- $g = f_1$ injects three flits at times 0, 4 and 8 (which complies with its regulator rate of $0.25$).
- $f_2$ injects two flits at times 0 and 4, which causes the first two flits from $g$ to be deflected in router (1,1) - but not the third. These deflections cause the three flits from $g$ to arrive at router (1,2) at times 5, 9 and 10.
- $f_3$ injects one flit at time 5 which causes the the first flit from $g$ to be deflected again, now in router (1,3). This deflection causes the three flits from $g$ to arrive at router (1,5) at times 11, 12 and 13.
- $f_3$ injects one flit at time 5 which causes the first flit from $g$ to be deflected again, now in router (1,3). This deflection causes the three flits from $g$ to arrive at router (1,5) at times 11, 12 and 13.

The consequence of this sequence of events is that, at time 11, flow $f$ would need to wait for 3 time units in order to

inject a flit, due to interference from the flits from $g$. This means that a safe upper bound $T_f^S$ must be $\geq 3$.

This also happens to be the "tightest" upper bound possible for this case, as a $T_f^s > 3$ would imply that two flits from $g$ released at least 12 time units apart (three periods) arrive only 3 time units apart, which is not possible as the two deflections by $f_2$ and $f_3$ can only add a total of 6 time units to a flit's trajectory.

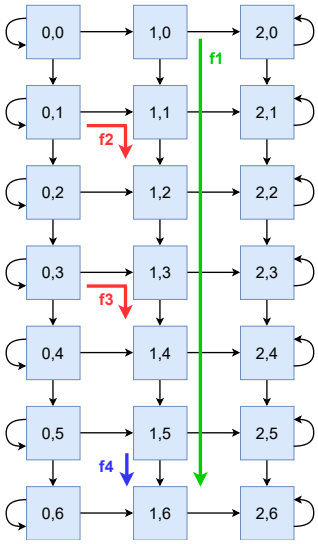



Fig. 4: Diagram for the counter-example to the original HopliteRT source-queuing analysis (adapted from [3])

### A. Original Source Queuing Bound

We can verify by inspection that $\Gamma_f^C = \{g\}$. Thus, according to Equations 4 and 5, we have that $\sigma\left(\Gamma_f^C\right) = \sigma_1 = 0.5$ and $\rho\left(\Gamma_f^C\right) = \rho_1 = 0.25$. Substituting into Equation 2:

$$T_f^s = \left\lceil \frac{\sigma\left(\Gamma_f^C\right)}{1 - \rho\left(\Gamma_f^C\right)} \right\rceil = \left\lceil \frac{1}{1 - 0.25} \right\rceil = 2 < 3 \qquad (17)$$

Which is not a safe upper bound, as we have seen above that it should be $\geq 3$.

### B. Corrected Source Queuing Bound

We begin by calculating the in-flight jitter $J_{gf}$. This example falls under the second case described in Section II, as the flow under analysis is injected South. We notice that the NoC is three switches-wide ($W = 3$), and that each flit of $g$ can be deflected up to twice before interfering with $f$: once by $f_2$ and once by $f_3$ (so $n_{def}(g, f) = 2$):

$$J_{gf} = n_{def}(g, f) \cdot W = 2 \cdot 3 = 6 \qquad (18)$$

We can now apply Equation 10 to determine the new value of $\sigma\left(\Gamma_f^C\right)$:

$$\sigma\left(\Gamma_f^C\right) = \sigma_{gf} = \sigma_g + J_{gf} \cdot \rho_g = 1 + 6 \cdot 0.25 = 2.5 \qquad (19)$$

We substitute $\sigma\left(\Gamma_f^C\right) = 2.5$ into Equation 2 to calculate the new upper bound $T_f^s$:

$$T_f^s = \left\lceil \frac{\sigma\left(\Gamma_f^C\right)}{1 - \rho\left(\Gamma_f^C\right)} \right\rceil = \left\lceil \frac{2.5}{1 - 0.25} \right\rceil = 4 > 3 \qquad (20)$$

Which is a safe upper bound for the source queuing of $f = f_4$, unlike the bound provided by the original approach.