Towards Delay Assurances in FIFO Networks

Satish Raghunath, Shivkumar Kalyanaraman,
Department of ECSE, Rensselaer Polytechnic Institute, Troy, NY,
Email: {raghus,kalyas}@rpi.edu

***Abstract*—The goal of this paper is to examine the gains of partial upgrades to existing FIFO networks, to support delay assurances. Specifically, we try to find the number of hops of FIFO multiplexing after which a latency target is violated. We first examine the effect of multiplexing two flows through successive FIFO schedulers, and for a simple scenario where cross-traffic is assumed to be absent, we derive a worst-case bound on the burstiness increase across n nodes. We use the result to obtain an effective service curve and a worst-case latency bound. We then examine the effect of having a priority scheduler at the entry of the network with FIFO nodes in the core. We provide a basis for determining the number of hops up to which a worst-case latency target is met.**

I. Introduction

Aggregate packet scheduling has attracted a lot of research attention lately. The purpose is to provide a scalable service with guaranteed rate and bounded delay. Diffserv [Bl98] has been proposed as an architecture to achieve service differentiation. Diffserv envisages guarantees to aggregates with certain pre-defined *Per-hop Behaviour* (PHB) at the individual routers. Accordingly, with expedited forwarding PHB (EF-PHB) [Da02], the EF aggregates must be guaranteed a particular minimum rate at each node of the network. FIFO packet scheduling has been proposed to support EF-PHB.

Recent work [ChBo01], [ZhDu01] has shown that delay bounds with a FIFO network depends on the utilization level and the number of hops. Consequently, it is seen that for number of hops being as low as 3, the utilization must be kept below 50% [ZhDu01]. Thus providing delay bounds with a FIFO network and aggregate scheduling needs more understanding. Simulation studies conducted in [SaGoVi99] also illustrate scenarios where a FIFO network can lead to very high end-to-end delays. A solution to the problem could be to extend the diffserv framework with some deadline information so that aggregates can be treated accordingly. One such solution is suggested in [ZhDu01]. Alternately, a simpler incremental strategy can be evolved wherein, the number of hops of FIFO multiplexing is reduced. A specialized scheduler and shaper can be inserted after a fixed number of FIFO nodes. We examine this approach by obtaining, for a simple network scenario, the number of hops of FIFO multiplexing before which a worst-case latency target is violated.

If the network edge is a specialized scheduler (say, a non-preemptive priority scheduler), it would be useful to examine the effect of a FIFO core network. In view of the discussion in the previous paragraph, we would be interested in quantifying the degradation in the worst-case latency with number of hops. To that end, we derive the worst-case burstiness of a flow passing through n hops. Here, we utilize the results from network calculus [Cr91], [Cr98], [BoTh02], while dealing with leaky-bucket constrained flows. We then utilize this result to obtain a characterization of the flow after it is multiplexed through n hops. The worst-case latency, in terms of the flow parameters and number of hops can then be calculated. The work in [ChBo01] gives a bound independent of the flow characteristics and in terms of the utilization level of the network. Our work does not need the utilization information. But [ChBo01] deals with arbitrary network topologies of FIFO networks, while the scenario considered here is very simple.

The contributions of this paper are two-fold - a) given the flow characteristics at the beginning of a simple network (without cross-traffic), we obtain the worst-case burstiness bound for the flow after it passes through n hops; b) we provide a way to find the number hops of FIFO multiplexing before which the worst-case latency is violated.

The rest of the paper is organized as follows. In Section II we detail some definitions of terms and notations used in the paper. In section III we obtain the burstiness bound and effective service curve for a simple network. In Section IV we discuss an incremental upgrade strategy using the results. Section V provides conclusions and points to future work.

II. Notation and background

We utilize results from deterministic network calculus [BoTh02]. Following are some basic definitions that will be useful in the succeeding sections. For a detailed introduction, the reader is referred to [BoTh02].

- **Wide-sense increasing functions**. A function f such that $f(s) \leq f(t)$ for all $s \leq t$ is wide-sense increasing. Define the set F to be the set of wide-sense increasing functions f such that $f(t) = 0$ for $t < 0$.
- **Data Flows**. A data flow, represented by a cumulative function $R(t) \in F$, is defined as the number of bits seen on the flow in the time interval $[0, t]$, and $R(0) = 0$.
- **Arrival Curve**. Given a function $\alpha \in F$, a flow R is constrained by α if and only if for all $s \leq t$, $R(t) - R(s) \leq \alpha(t-s)$. R is said to have α as an arrival curve and is said to be α-smooth.
- **Min-plus Convolution and De-Convolution**. For functions f and g from set F, min-plus convolution is defined as:

$$(f \otimes g)(t) = \inf_{0 \leq s \leq t} \{f(t-s) + g(s)\}$$

$[x]^+$	x if $x > 0$, zero else
$[x(t)]\ 1_{\{t>y\}}$	$x(t)$ if $t > y$, zero else
$\beta_{R,T}$	Rate (R) latency (T) curve
$\gamma_{r,b}$	Leaky bucket with rate r, bucket b
β_i^θ	Service curve family with param θ
β_i^n	Service curve at node n, flow i
$\beta_i^{m,n}$	Combined Service for nodes m to n, flow i
$b_i^{(n)}$	Burstiness of flow i after n nodes

TABLE I
NOTATIONS USED IN THE PAPER

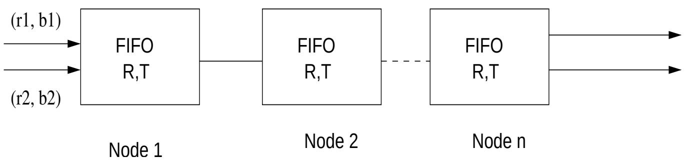

Fig. 1. Token-bucket constrained flows fed to a cascade of FIFO nodes

and de-convolution is defined as:

$$(f \oslash g)(t) = \sup_{u \geq 0}\{f(t+u) - g(u)\}$$

- **Service Curve**. Consider a flow going through a system S, with input and output functions R and R^*. S offers to the flow a service curve β if and only if $\beta \in F$ and $R^* \geq R \otimes \beta$.
- **Strict Service Curve**. A system S offers a strict service curve β to a flow if, during any backlogged period of duration u, the output of the flow is at least equal to $\beta(u)$.
- **Rate-latency Service**. A service of the form $R[t-T]^+$, where R denotes the rate and T the latency, is known as a rate-latency service, denoted as $\beta_{R,T}$.

The notations employed in the following sections has been summarized in Table I.

III. CASCADING FIFO NODES

In this section, we examine the effect of successive FIFO schedulers on the burstiness of two multiplexed flows. We first present an existing result regarding burstiness increase for one node. We extend this result for n nodes. We then obtain the effective service curve for two multiplexed flows that traverse a string of n nodes.

A. Burstiness increase due to FIFO nodes

The following theorem appears in [BoTh02].

Theorem III.1: Consider a node serving two flows, 1 and 2, in FIFO order. Assume that flow 1 is constrained by one leaky bucket with rate r_1 and burstiness b_1, and flow 2 is constrained by a sub-additive arrival curve α_2. Assume that the node guarantees to the aggregate of the two flows a rate latency service curve $\beta_{R,T}$. Call $r_2 := \inf_{t>0} \frac{1}{t}\alpha_2(t)$ the maximum sustainable rate for flow 2. If $r_1 + r_2 < R$, then at the output, flow 1 is constrained by one leaky bucket with rate r_1 and burstiness b_1^*

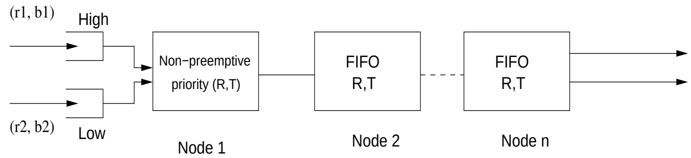

Fig. 2. A non-preemptive priority scheduler followed by n-1 FIFO nodes

with

$$b_1^* = b_1 + r_1\left(T + \frac{\hat{B}}{R}\right)$$

and

$$\hat{B} = \sup_{t \geq 0}[\alpha_2(t) + r_1 t - Rt]$$

Theorem III.1 can be specialized to obtain the burstiness increase when flow 2 is also leaky bucket constrained. We then have,

$$b_1^* = b_1 + r_1\left(T + \frac{b_2}{R}\right) \tag{1}$$

Armed with this result we consider a scenario depicted in figure 1. We first note that equation (1) can be used to obtain a leaky-bucket characterization of flow 1 as it enters node 2. If we apply theorem III.1 again, we obtain the bound for burstiness after passing through two nodes. Thus we obtain the following result for n FIFO nodes:

Theorem III.2: **Burstiness Increase due to n FIFO nodes.** *Consider n nodes serving two flows, 1 and 2, in FIFO order. Assume that flow i is constrained by the leaky bucket (r_i, b_i) when it enters node 1. If $r_1 + r_2 < R$, then, at the output of the n^{th} node, flow 1 is constrained by one leaky bucket with rate r_1 and burstiness $b_1^{(n)}$ with*

$$b_1^{(n)} = b_1 + (T + \frac{b_1}{R})\left(\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \frac{(r_1 r_2)^i}{R^{2i-1}} \begin{pmatrix} n \\ 2i \end{pmatrix}\right) + $$
$$+(T + \frac{b_2}{R})\left(\sum_{i=1}^{\lfloor \frac{n+1}{2} \rfloor} \begin{pmatrix} n \\ 2i-1 \end{pmatrix} \frac{r_1^i r_2^{(i-1)}}{R^{2i-2}}\right) \tag{2}$$

Proof: Please refer to the Appendix. ■

It is easy to see that equation (2) reduces to (1) for $n = 1$.

B. Effective service curve for n FIFO nodes

Consider now, n FIFO nodes that are each characterized by a minimum service curve $\beta_{R,T}$. In this section we are interested in the effective service curve a flow gets out of the n nodes when multiplexed with another flow. For this purpose we employ a theorem from [BoTh02] and exploit the fact that convolution of two service curves is equivalent to the service offered by two nodes with those service curves, in succession.

Theorem III.3 appears in [Cr98], [BoTh02].

Theorem III.3: Consider a lossless node serving two flows, 1 and 2, in FIFO order. Assume that packet arrivals are instantaneous. Assume that the node guarantees a minimum service curve β to the aggregate of the two flows. Assume that flow 2 is α_2-smooth. Define the family of functions β_1^θ by

$$\beta_1^\theta = [\beta(t) - \alpha_2(t-\theta)]^+ 1_{\{t>\theta\}} \tag{3}$$

Call $R_1(t), R'_1(t)$ the input and output for flow 1. Then for any $\theta \geq 0$

$$R'_1(t) \geq R_1 \otimes \beta_1^\theta$$

If β_1^θ is wide-sense increasing, flow 1 is guaranteed the service curve β_1^θ.

If flow i is constrained by a leaky bucket γ_{r_i,b_i}, equation 3 can be further specialized [BoTh02] and stated as

$$\begin{aligned}\beta_1^\theta &= [R(t-T) - \gamma_{r_2,b_2}(t-\theta)] \;\; with\, \theta = (T + \frac{b_2}{R}) \quad (4)\\ &= [(R-r_2)(t-\theta)] \;\; with\, \theta = (T + \frac{b_2}{R}) \quad (5)\end{aligned}$$

Then β_1^θ is a service curve guaranteed to γ_{r_1,b_1}. Let the service seen by flow i at node j be denoted by β_i^j. Let the effective service seen by flow i if it passes through n nodes be denoted as $\beta_i^{1,n}$. Then we have,

$$\beta_i^{1,n} = \beta_1^1 \otimes \beta_1^2 \otimes \ldots \otimes \beta_1^n$$

The following proposition then gives the effective service curve offered by n FIFO nodes in succession.

Proposition III.1: **An Effective Service Curve for N FIFO nodes**. *Consider n lossless nodes serving two flows, 1 and 2, in FIFO order. Assume that packet arrivals are instantaneous. Assume that each node guarantees a minimum service curve $\beta_{R,T}$ to the aggregate of the two flows. Assume that flow i is constrained by the leaky-bucket γ_{r_i,b_i}. Then the effective service curve for the n nodes, for flow 1, is given by:*

$$\beta_1^{1,n} = [(R-r_2)(t - \sum_{i=0}^{n-1} \theta_i)]^+ 1_{t > \sum_{i=0}^{n-1} \theta_i} \quad (6)$$

$$\theta_i = \begin{cases} T + \frac{b_2}{R} & i = 0 \\ T + \frac{b_2^{(i)}}{R} & else \end{cases} \quad (7)$$

and $b_2^{(n)}$ is given in equation (2).

Proof: Please refer to the Appendix for an inductive proof. Intuitively, since each FIFO node offers to flow 1, a service equal to rate $R - r_2$ with a latency θ_i (defined above), the effective service is again a rate-latency function with the latency being the sum of the latencies at each node. ∎

We have thus obtained the effective service curve for n FIFO nodes in equation (6) if the inputs are leaky bucket constrained.

C. Priority scheduling at the edge

We now have the tools to derive the effective service curve for a scenario that interests us, namely, a non-preemptive priority scheduler at the edge followed by a cascade of FIFO nodes, as depicted in figure (2). Let the effective service curve for flow 1 in figure (2) be $\beta_1^{1,n}$. The properties of the priority node are characterized by the following theorem from [BoTh02].

Theorem III.4: **Non preemptive priority node**. Consider a node serving two flows, H and L, with non-preemptive priority given to flow H. Assume that the node guarantees a strict service curve β to the aggregate of the two flows. Then the high priority node guarantees a strict service curve $\beta_H(t) = [\beta(t) - l_{max}^L]^+$ where l_{max}^L is the maximum packet size for the low priority flow. If in addition the high priority flow is α_H-smooth, then define β_L by $[\beta(t) - \alpha_H(t)]^+$. If β_L is wide-sense increasing, then it is a service curve for the low priority flow.

If flow 1 is the high priority flow, $\beta_1^{1,n}$ can be easily obtained as being a rate-latency curve with latency being the sum of $\frac{l_{max}^L}{R}$ and $\sum_{i=1}^{n-1} \theta_i$. Using the fact that, for an input with arrival curve α, the arrival curve at the output of a node with service β is obtained as $\alpha \oslash \beta$, we find that the flow 1 is bounded at the output of node 1 by $r_1 t + b_1 + r_1 \frac{l_{max}^1}{R}$. Similarly flow 2 is bounded by the curve $r_2 t + b_2 + r_2 \frac{b_1}{R - r_1}$. We can once again employ equation (2) to obtain the characterization of the flows after n hops.

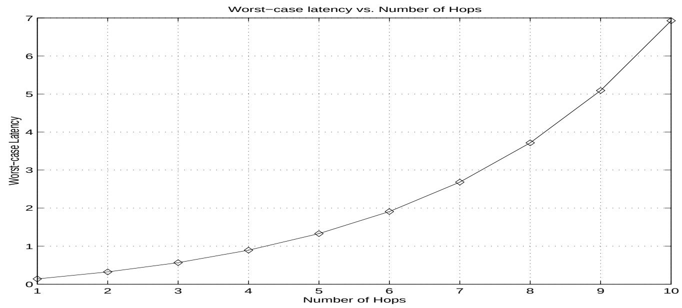

Fig. 3. Effect of number of hops on worst-case latency

IV. Incremental deployment of specialized schedulers

The motivation for the analysis of the previous section lies in finding a firm basis for incremental deployment of specialized schedulers. Given that a network-wide upgrade for a priority queue is prohibitive, the question we would like to answer is as follows. What is the degradation in service offered to a high priority flow if it has to be multiplexed across a fixed number of FIFO nodes? The result in equation (6) is a step towards finding a useful answer to this question. In this section we conduct some simple numerical studies using the results of the previous sections. We first examine the effect of number of hops on the burstiness of the flow, given the initial leaky-bucket constraints of the flow. We then consider the reverse case, we fix a target worst-case latency and the number of hops, and present directions for choosing bucket depths for the flows at the entry of the network.

A. Effect of number of hops

Consider a network with priority queues throughout. The high priority flow obviously faces a latency of only $\tau = N(T + l_{max}^L)$ if there are N nodes to be traversed. The worst-case latency faced in a network as pictured in figure (2) on the other hand is obtained by $\hat{\tau} = \sum_{i=0}^{n-1} \theta_i$. Comparing $\hat{\tau}$ and τ directly provides us with a measure of the degradation in worst-case latency suffered due to the series of FIFO nodes.

We numerically evaluate the value of $\hat{\tau}$ for 1 to 10 hops and present it in figure (3). While viewing this plot, it is important to note that these values are for the worst-case scenario, that is it only means that the latency will never exceed those values. To obtain figure (3) we use values of $(r_1 = 1000000, b_1 =$

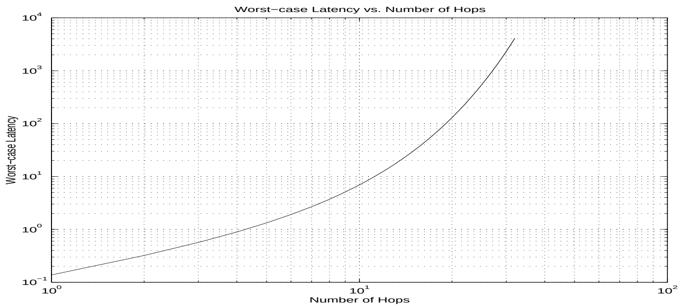

Fig. 4. Effect of number of hops on worst-case latency (plotted till 32 hops)

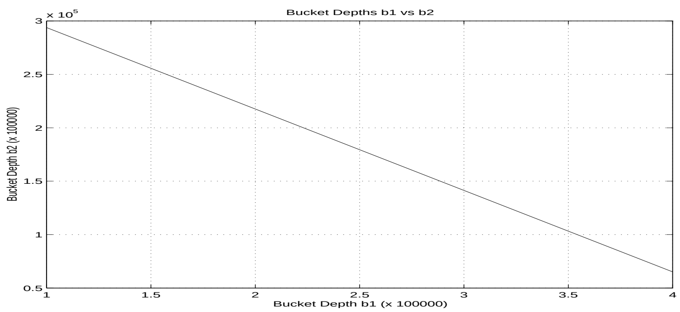

Fig. 5. Worst-case line relating bucket depths b_1 and b_2

$10000, r_2 = 1000000, b_2 = 10000, R = 3000000, T = 0.1)$. Thus given the leaky-bucket constraints for the flows, we could decide on the number of hops after which we place a specialized scheduler if we know the tolerable degradation in delay. From the plot, we see that for a worst-case target latency of 3, every 6^{th} hop must involve a shaping component.

The worst-case bound rapidly increases with the number of hops and is not really useful for larger n. This is illustrated in figure (4) where the number of hops is varied up to 32.

B. *Choosing bucket depths*

Using equations (6,7), we can find an expression for $\hat{\tau}$ in terms of b_1, b_2. It will be of the form

$$\hat{\tau} = c_0 + c_1 b_1 + c_2 b_2 \;\; c_i > 0$$

For each fixed $\hat{\tau}$ we then obtain a worst-case "operating line" along (or below) which b_1 and b_2 can be chosen. We can set the target $\hat{\tau}$ to a value greater than c_0 to obtain a line which can help us choose b_1 and b_2. We use the values of r_1, r_2, R, and T as $(r_1 = 1000000, r_2 = 1000000, R = 3000000, T = 0.1)$ and obtain the relation between the latency $\hat{\tau}$, b_1 and b_2, for 5 hops, as:

$$\hat{\tau} = 1.29 + 2.43 \times 10^{-6} b_1 + 1.85 \times 10^{-6} b_2$$

Setting the target latency as $\hat{\tau} = 2$, we obtain,

$$b_1 = 2.94 \times 10^5 - 0.76 b_2$$

This is plotted in figure (5). Thus given a target latency, we can find the worst-case operating line on or below which the bucket depths may be chosen.

V. Conclusions and Future Work

Incremental deployment of specialized schedulers in the network can achieve worst-case latency bounds. In order to find the number of hops after which the burstiness increase due to FIFO multiplexing violates the target latency, an effective service curve can be employed. For a simple network of FIFO nodes, we obtained a worst-case bound on the burstiness increase assuming leaky-bucket inputs. Utilizing this result, an effective service curve was obtained. An example was used to illustrate the use of the results to choose the number of hops between upgraded nodes.

The current deductions do not account for multiple classes of traffic, arbitrary topologies and cross-traffic. The arrival curves are simple leaky-bucket constraints. Further investigations will involve relaxing these constraints to gain insight into a more complex network scenario.

References

[Bl98] S. Blake, et al, "An Architecture for Differentiated Services," RFC 2745, December 98

[BoTh02] Jean-Yves Les Boudec, Patric Thiran, "Network Calculus", Parts I, II, III, Online version of the book, Springer Verlag LNCS 2050, Version Jan 16, 2002

[ChBo01] Anna Charny, J.-Y. Le Boudec, "Delay Bounds in a Network with Aggregate Scheduling," Proc. QoFIS, Berlin, Germany, October 2000 Online version of the book, Springer Verlag LNCS 2050, Version Jan 16, 2002

[Cr98] R.L. Cruz, "Sced+: Efficient management of quality of service guarantees," IEEE Infocom 98

[Cr91] R.L. Cruz, "A calculus for network delay, part I: Network elements in isolation," IEEE Transactions on Information Theory, Vol.37., No. 1, January 1991, pp.114-131

[Da02] B. Davie, et al, "An Expedited Forwarding PHB (Per-Hop Behavior)," RFC 3246, March 2002.

[SaGoVi99] J. Sahni, P. Goyal, H. M. Vin, "Scheduling CBR Flows: FIFO or Per-flow Queuing," NOSSDAV 99

[ZhDu01] Z. Zhang, Z. Duan, Y.T. Hou, "Fundamental Trade-offs in Aggregate Packet Scheduling," ICNP 2001

Appendix

Proof of theorem III.2

Proof: For n=1, the result clearly holds.

For n=2, consider a FIFO node whose inputs are $\gamma_{r_1, b_1^{(1)}}$ and $\gamma_{r_2, b_2^{(1)}}$. Using theorem III.1 we have,

$$\begin{aligned} b_1^{(2)} &= b_1^{(1)} + r_1 \left(T + \frac{b_2^{(1)}}{R} \right) && (8) \\ &= b_1 + \left(T + \frac{b_2}{R} \right) 2 r_1 T + \left(T + \frac{b_1}{R} \right) \frac{r_1 r_2}{R} && (9) \end{aligned}$$

which coincides with the expression obtained by substituting $n = 2$ in theorem III.2. Assume the result holds for an arbitrary natural number m. The following steps prove that the result must also hold for $m + 1$.

$$\begin{aligned} b_1^{(m+1)} &= b_1^{(m)} + r_1 \left(T + \frac{b_2^{(m)}}{R} \right) && (10) \\ &= b_1 + \left(T + \frac{b_2}{R} \right) \left(\sum_{i=1}^{\lfloor \frac{m+1}{2} \rfloor} \binom{m}{2i-1} \frac{r_1^i r_2^{(i-1)}}{R^{2i-2}} \right. \end{aligned}$$

$$+\left(T+\frac{b_2}{R}\right)\left(r_1+\sum_{i=1}^{\lfloor\frac{m}{2}\rfloor}\frac{r_1^{(i+1)}r_2^i}{R^{2i}}\left(\begin{array}{c}m\\2i\end{array}\right)\right)$$

$$+\left(T+\frac{b_1}{R}\right)\left(\sum_{i=1}^{\lfloor\frac{m}{2}\rfloor}\frac{(r_1r_2)^i}{R^{2i-1}}\left(\begin{array}{c}m\\2i\end{array}\right)\right)+$$

$$\left(T+\frac{b_1}{R}\right)\left(\sum_{i=1}^{\lfloor\frac{m+1}{2}\rfloor}\left(\begin{array}{c}m\\2i-1\end{array}\right)\frac{(r_1r_2)^i}{R^{2i-1}}\right) \quad (11)$$

Noting that,

$$\left(\begin{array}{c}n\\r\end{array}\right)+\left(\begin{array}{c}n\\r-1\end{array}\right)=\left(\begin{array}{c}n+1\\r\end{array}\right)$$

the appropriate terms can be combined in the above equation to yield,

$$b_1^{(m+1)}=b_1+(T+\frac{b_1}{R})\left(\sum_{i=1}^{\lfloor\frac{m+1}{2}\rfloor}\frac{(r_1r_2)^i}{R^{2i-1}}\left(\begin{array}{c}m+1\\2i\end{array}\right)\right)+$$

$$+(T+\frac{b_2}{R})\left(\sum_{i=1}^{\lfloor\frac{m+2}{2}\rfloor}\left(\begin{array}{c}m+1\\2i-1\end{array}\right)\frac{r_1^ir_2^{(i-1)}}{R^{2i-2}}\right) \quad (12)$$

which is the desired form for $m+1$. ■

PROOF OF PROPOSITION III.1

Proof: For $n=1$, we have equation (4).

In the following equations, θ_i is defined by equation (7). For $n=2$, consider,

$$\begin{aligned}\beta_1^{1,2} &= \beta_1^1\otimes\beta_1^2\\ &= \inf_{0\le s\le t}\{\beta_1^1(s)+\beta_1^2(t-s)\}\\ &= \inf_{0\le s\le\theta_0}\{\beta_1^2(t-s)\}\wedge\inf_{s>\theta_0}\{\beta_1^1(s)+\beta_1^2(t-s)\}\end{aligned}$$

Evaluating the above equation for different values of t we find the following. For $t\le(\theta_0+\theta_1)$, $\beta_1^{1,2}=0$. For $t>(\theta_0+\theta_1)$,

$$\begin{aligned}\beta_1^{1,2} &= \beta_1^2(t-\theta_0)\\ &\quad\wedge\inf_{\theta_0<s<t-\theta_1}\{\beta_1^1(s)+\beta_1^2(t-s)\}\\ &\quad\wedge\inf_{s\ge t-\theta_1}\{\beta_1^1(s)\} &(13)\\ &= (R-r_2)(t-\theta_0-\theta_1)\wedge(R-r_2)(t-\theta_0-\theta_1)\\ &\quad\wedge(R-r_2)(t-\theta_0-\theta_1) &(14)\\ &= (R-r_2)(t-\theta_0-\theta_1) &(15)\end{aligned}$$

which is the required form for n=2.

Now assume that the result holds for m. Consider, $n=m+1$. Tracing the steps in equations (13,14,15) with $\sum_{i=0}^{m-1}\theta_i$ and θ_m, we easily see the result for $n=m+1$. ■