



工业互联网产业联盟  
Alliance of Industrial Internet

# 基于 TSN 的端到端网络 演算技术研究 (2022 年)

工业互联网产业联盟  
Alliance of Industrial Internet

工业互联网产业联盟 (AII)

2023 年 6 月



# 声 明

本报告所载的材料和信息，包括但不限于文本、图片、数据、观点、建议，不构成法律建议，也不应替代律师意见。本报告所有材料或内容的知识产权归工业互联网产业联盟所有（注明是引自其他方的内容除外），并受法律保护。如需转载，需联系本联盟并获得授权许可。未经授权许可，任何人不得将报告的全部或部分内容以发布、转载、汇编、转让、出售等方式使用，不得将报告的全部或部分内容通过网络方式传播，不得在任何公开场合使用报告内相关描述及相关数据图表。违反上述声明者，本联盟将追究其相关法律责任。

工业互联网产业联盟  
Alliance of Industrial Internet

工业互联网产业联盟

联系电话：010-62305887

邮箱：aia@caict.ac.cn



# 前 言

TSN技术发展多年，在标准上也定义了多项TSN子技术。对于构建或更好地构建端到端TSN网络方案，或者在更广义的角度，构建提供确定性服务能力的网络方案，还涉及到TSN标准之外的技术。例如网络演算就是其中的一项关键技术。网络演算面向网络的确定性时延上界、缓存上界的分析，可用于TSN的有界时延保证的理论基础。

本研究项目着力于基于TSN的端到端网络演算技术研究，考虑网络演算与TSN数据面调度机制的配合关系，以及网络演算在TSN控制面方案中的应用。同时，将TSN应用时工程问题考虑在内，结合TSN设备和生态，以期望加快TSN产业化发展与应用落地。

牵头编写单位：

华为技术有限公司

参与编写单位：

深圳市三旺通信股份有限公司

中国联合网络通信有限公司

中国移动通信有限公司研究院

网络通信与安全紫金山实验室

西安云维智联科技有限公司

贵州大学

工业互联网产业  
Alliance of Industrial In

编写组成员（排名不分先后）：

陈李昊、张嘉怡、高涛、王童童、熊伟、邓顺义、高一颠、贾雪琴、韩政鑫、刘鹏、何斌、王红春、邹赛



工业互联网产业联盟公众号



# 目 录

<b>1 概述</b> .....	<b>1</b>
<b>2 网络演算介绍</b> .....	<b>1</b>
2.1 网络演算基本概念 .....	1
2.2 确定性网络演算 .....	8
2.3 随机网络演算 .....	13
2.4 网络演算在 AFDX 网络中的应用 .....	15
<b>3 TSN 调度机制与网络演算的关系</b> .....	<b>17</b>
3.1 基于经验的做法 .....	17
3.2 SP 的时延计算 .....	18
3.3 CBS 的时延计算 .....	19
3.4 ATS 的时延计算.....	21
3.5 关于时间门控 .....	23
<b>4 应用网络演算的 TSN 方案</b> .....	<b>23</b>
4.1 服务与流量建模 .....	23
4.2 网络演算与 TSN 配置模型、TSN 方案的关系 .....	25
<b>附录 A 确定性方案的本质探讨</b> .....	<b>27</b>
对确定性方案的分类 .....	27
运用网络演算进行性能分析 .....	30
整形的影响 .....	32
<b>参考文献</b> .....	<b>35</b>





## 1 概述

TSN 制定了多种调度机制，简单来讲可以分为同步调度（TAS）和异步调度（SP, CBS, ATS）两大类。基于这些调度机制，配合合理的资源预留，则可以实现对流量的有界时延等保障。为了进行资源预留，TSN 定义了三种配置模型，其中两种都需要一个集中的网络控制器功能实体。在网络控制器上，网络演算是属于算法模块的重要功能。网络演算以当前的网络信息和流量信息（包括对端到端时延上界的保障需求）作为输入，基于 TSN 异步调度类机制，计算出网络配置参数，从而帮助实现整体的 TSN 方案。

就网络演算与 TSN 的关系，尤其是与 TSN 调度机制的关系，具体而言，就是 TSN 调度机制及其配置参数，会对设备的服务能力——即处理报文排队转发时的特征与约束——产生显著的影响，而这就是网络演算在对网络设备进行建模时需要重点考虑的。特别是当使用 TSN 异步调度类机制时，网络在不同条件下对流量所能提供的时延保障，不像门控类方案那样可以轻松获得。基于经验的计算方案会带来精度上的误差和使用上的约束，而网络演算基于严谨的数学分析过程，可以成为 TSN 时延分析和网络配置指导的理想工具。

本文研究、总结了网络演算的基本概念和方法，TSN 调度机制与网络演算的具体关系，以及基于网络演算的 TSN 方案应用案例。

## 2 网络演算介绍

### 2.1 网络演算基本概念

Network calculus 直译过来叫网络微积分，又称网络演算（下文均称网络演算）。

网络演算是用于分析如通信网络、数字电路、并行程序等人造系统的一种数学方法，主要分析目标是时延等“服务保障（performance guarantees）”。网络演算最初由 R. L. Cruz 在 1990 年提出<sup>[1][2]</sup>，此后迅速发展<sup>[3][4]</sup>。

网络演算主要由以下关键部件组成：

- **流量建模**：使用到达曲线（Arrival Curve）描述流量到达网络设备时的行为特征。

- **网络服务建模**：使用服务曲线（Service Curve）描述网络设备的服务能力，重点是对调度器行为的描述。

- **计算方法**：考虑网络拓扑、流量间的相互干扰等，计算任意一个流量在网络中传输的端到端时延上界。

通过建立到达曲线和服务曲线，网络演算可以把复杂且非线性的网络系统转变为易于分析的线性系统，进而计算求解。

目前网络演算的主要技术分支有：

- **确定性网络演算**（Deterministic Network Calculus, DNC），基于到达曲线和服务曲线，计算 100% 保障的时延上界，如端到端最坏时延不会超过 10ms。<sup>[4][5]</sup>

- **随机网络演算**（Stochastic Network Calculus, SNC），建模时考虑流量到达和网络服务的随机特性，引入了概率论，计算满足概率的时延上界，如端到端时延 99.99% 不超过 10ms。<sup>[6][8]</sup>

学术界对于网络演算研究的应用场景从最初 ATM 网络，发展到 IP 网络，再逐渐扩展到机载网络、工业网络、车载网络，以及无线网络等。

目前网络演算最成功的商业应用案例是在机载网络。航空电子全双工交换式以太网（AFDX, Avionics Full Duplex switched Ethernet）技术需配合网络演算的理论证明，验证相关流量被保障获得的时延上界满足其需求<sup>[9][10]</sup>。AFDX 已应用于空客 A380、波音 B787 等大型客机上。西门子的西门子网络规划工具（SINETPLAN）也使用了网络演算来辅助 PROFINET 网络的设计<sup>[11]</sup>。

### 2.1.1 到达曲线——描述流量特征

网络演算理论使用到达曲线  $\alpha(t)$  来描述流量特征。到达曲线是流量到达行为的上包络，即对于任意时刻  $t$  以及  $0 \leq s \leq t$ ，满足： $R(t) - R(s) \leq \alpha(t-s)$ 。其中， $R(t)$  是流量的累积到达函数。如图 1 所示，黑色曲线是一个流量的累积到达，随时间单调递增（非严格）；红线是到达曲线。将到达曲线的起点从原点开始沿着黑色曲线任意移动，黑色曲线始终位于红线之下。

另外，图 1 中的到达曲线是一种经典的到达曲线的形式，可以描述为  $\alpha(t) = rt + b$ 。直观的理解是： $r$  描述了流量的长期平均速率，即图中红线的斜率； $b$  描述了流量的最大突发量，即图中过原点的红线与纵轴的交点。

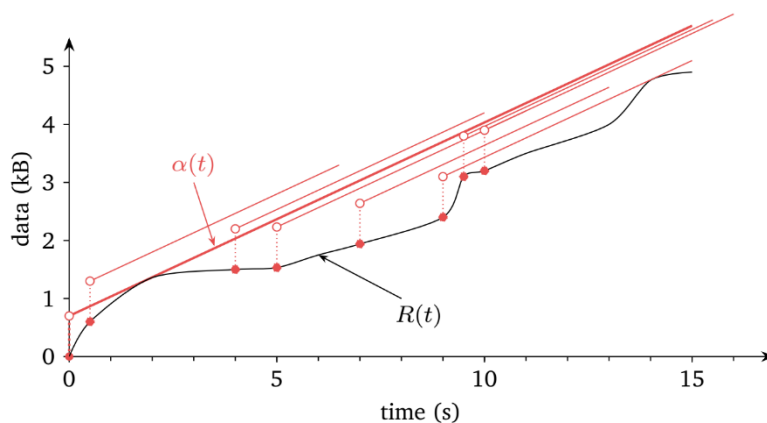


图 1 到达曲线  $\alpha(t)$  与流量累积到达  $R(t)$  的关系<sup>[7]</sup>

到达曲线建模真实流量的举例：

设有一个流量，从一个 1Gbps 的链路到达网络设备，每间隔 1ms 会有 5 个连续报文的突发到达，报文长度均为 1500Byte。如图 2 中，黑线表示了这个流量的累积到达量。最简单地，我们可以用该流量的平均速率 60Mbps 和突发量 60Kbit 作为到达曲线  $\alpha(t)=rt+b$  中的  $r$  和  $b$ ，此时刻画的到达曲线如图 2 中红线所示。如果考虑到流量到达受链路速率的约束，则可以用紫线来描述。紫线也是到达曲线，而且显然，紫线对于流量到达行为的刻画更为准确。

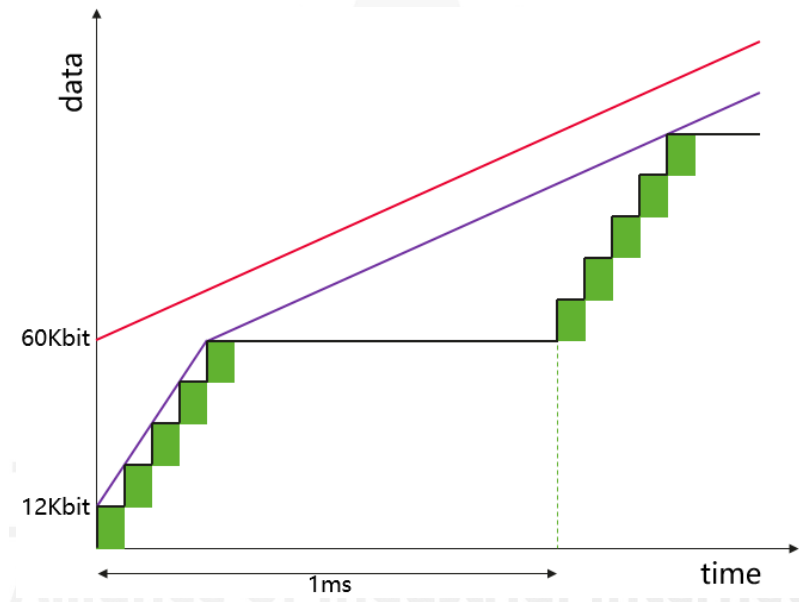


图 2 到达曲线实例

到达曲线可以描述单流，也可以描述聚合流。例如，聚合流由两个单流组成，则  $\alpha=\alpha_1+\alpha_2$ ，如图 3。

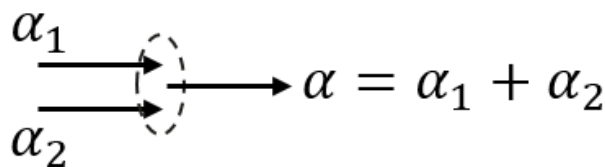


图 3 到达曲线的聚合

### 2.1.2 服务曲线——描述调度器行为

网络演算理论使用**服务曲线**  $\beta(t)$  来描述网络设备的服务能力，且可以具体到描述设备对于其调度器中一个特定队列的服务能力。服务曲线是服务能力的下包络，满足： $R^*(t) \geq (R \otimes \beta)(t)$ 。其中， $R(t)$  和  $R^*(t)$  分别是流量的累积到达和累积输出。如图 4 所示，黑色曲线是一个流量的累积到达，随时间单调递增（非严格）；由于蓝色的服务曲线是对于该流量服务能力的下包络，所以流量的实际累积输出（即红色曲线）一定总在流量的累积到达  $R$  与服务曲线  $\beta$  的卷积（即绿色曲线）之上。该处卷积指最小加卷积（min-plus convolution）： $f \otimes g(t) = \inf_{0 \leq s \leq t} \{f(s) + g(t-s)\}$ ，用以表征两个函数之间的加和下界。换言之，流量的实际累积输出同时受到流量的累积到达量和设备调度器服务能力的限制，卷积运算描述了这两个限制同时作用后的结果。

另外，图 4 中蓝色的服务曲线是一种经典的服务曲线形式，即  $\beta(t) = R(t-T)$ 。这种服务曲线也被称作“rate-latency”服务曲线。直观的理解是： $T$  描述了流量抵达后，最长需要等待  $T$  时间，才能得到调度服务； $R$  描述了流量可以从调度器上获取的长期平均最小调度速率。

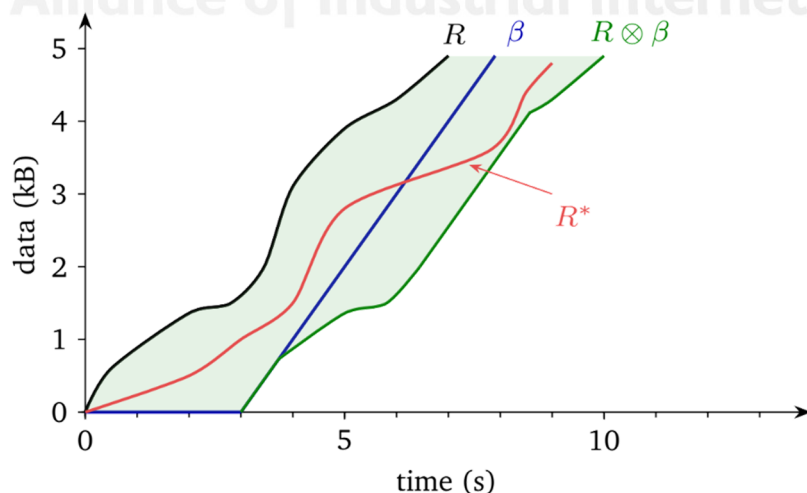


图 4 服务曲线  $\beta(t)$  与流量累积输出  $R^*(t)$  的关系<sup>[7]</sup>

服务曲线建模 DRR 调度器的举例：

Deficit Round-Robin (DRR, 亦称 Deficit Weighted Round-Robin, DWRR) 是一种经典的加权轮询调度器 [12]，很多网络设备上都使用了 DRR 或类似于 DRR 的调度机制。DRR 的基本原理示意如图 5：三个队列分别配置了权重值  $Q_i$ ，每一轮轮询过程中，每个队列根据权重值来调度报文出队。

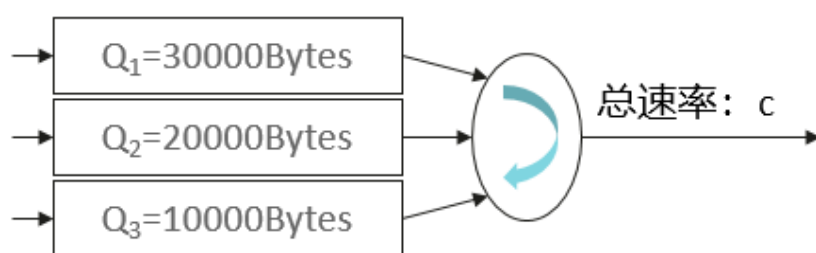


图 5 DRR 原理示意图

[13] 给出了 DRR 的服务曲线推导，这里直接列出结论，任意一个 DRR 队列  $i$  的服务曲线为：

$$\beta_i^{DRR} = \underbrace{\frac{Q_i}{Q} c}_{R} \left( t - \underbrace{\frac{Q_i \sum_{j \neq i} L_j + (Q - Q_i)(Q_i + L_i)}{Q_i c}}_T \right)$$

其中， $Q$  是所有 DRR 队列的总权重值， $L_i$ 、 $L_j$  是各 DRR 队列中的最大报文长。该服务曲线符合  $\beta(t)=R(t-T)$  的形式。具体地，对于“ $R$ ”的理解，即该 DRR 队列  $i$  中的流量所能获得的长期平均最小调度速率就是该队列的权重份额的比例乘以总服务速率；而“ $T$ ”则是由 DRR 调度机制造成的最大轮询等待时间。

多个服务曲线级联时，可以进行卷积运算。一个流量先通过服务曲线为  $\beta_1$  的调度器，再通过服务曲线为  $\beta_2$  的调度器，相当于只通过一个服务曲线为  $\beta=\beta_1 \otimes \beta_2$  的调度器，如图 6。

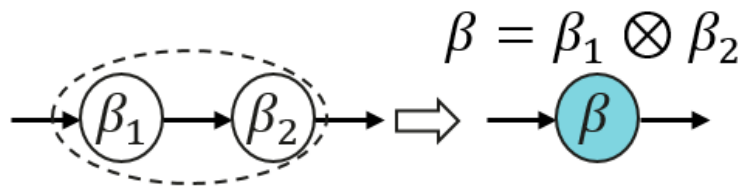


图6 服务曲线的级联

### 2.1.3 时延上界

到达曲线  $\alpha$  描述了流量到达行为的上包络，亦即流量到达行为的上界；服务曲线  $\beta$  描述了设备调度器服务能力的下包络，亦即服务能力的下界。如图 7 所示，将两个曲线放在一起，则两个曲线的最大垂直距离为该流量在该设备上所能占用的缓存量的上限，即缓存上界；而两个曲线的最大水平距离为该流量中任意报文在该设备调度器中所能经历的最大时延，即时延上界  $D=h(\alpha, \beta)$ 。

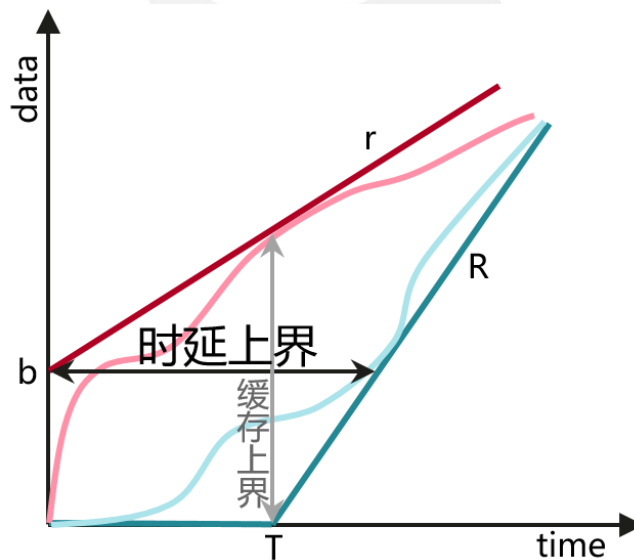


图7 时延上界与缓存上界

时延的上界 (delay/latency upper-bound)，或者说时延界 (delay/latency bound)，是网络演算最主要的研究对象。由于时延

界是基于流量到达行为的上界和设备服务能力的下界所给出的，所以时延界也被称作为最差情况下的时延（worst-case delay/latency）。用通俗的话来说，当一个流量的最大突发刚好赶上一个调度器最忙碌的状态时，这个流量中的某个报文可能会在调度器里经历这个最差情况下的时延。

下面举一个具体的例子：选取 2.1.1 节中的流量进行分析，该流量的到达曲线为： $\alpha(t) = rt+b = 6*10^7*t+6*10^4$ ，设为最高优先级；设经过一个总服务速率为 1Gbps 的绝对优先级（Strict Priority, SP）调度器，所有低优先级流量的最大报文长为 1500Byte=12000bit，即等待一个低优先级最长报文完成传输需要 12 微秒，则调度器对该流量的服务曲线为： $\beta(t)=R(t-T)=1*10^9(t - 1.2*10^{-5})$ 。至此，由于  $R>r$ ，很容易求得时延上界： $D = T + b/R=7.2*10^{-5}s$ 。

以上是一个流量经过一个调度器（一个设备）的情况，得到的时延上界是单点的时延上界。当然，网络演算也适用于多节点级联网络的端到端时延上界的计算，也适用于网络中有大量不同特征的流量互相交汇、干扰的情况。在下面两节，会给出进一步的介绍。

## 2.2 确定性网络演算

确定性网络演算（DNC, Deterministic Network Calculus）基于到达曲线和服务曲线，计算单网络节点或网络端到端 100%保障的时延上界。常用的端到端计算方法有：

- 最小加（Min-plus）代数类算法（后面称代数类算法）：TFA、SFA、PM00A、TMA 等。其中，TFA 是最基础的计算方法，沿路径逐跳计算时延上界，然后相加得到端到端时延；SFA、PM00A、TMA 是通过卷积运算考虑设备多跳级联时的服务情况的算法。代数类算法将在 2.2.1 节



中介绍。

- 优化类算法：把到达曲线、服务曲线等转化为约束条件，构建优化问题，求解时延上界。优化类算法将在 2.2.2 节中介绍。

由于到达曲线或服务曲线建模的精度不足、算法中存在近似处理等原因，DNC 的结果会产生一定的悲观性。即如图 8 所示，DNC 算出的某流量的时延上界（红线）会大于等于其真实的时延上界（黑线）。确定性网络演算的主要研究目标，就是通过选取合适的算法，并进行设计优化，使得时延上界的计算结果更准确（即提高紧性、降低悲观性），同时也关注计算精度与计算复杂度之间的权衡取舍。

另外，图 8 中流量真实的时延上界（黑线）也会大于等于通过仿真测量所得到的最差时延，即测试不一定能探测到最差时延。这种测试类误差不是网络演算的研究对象。

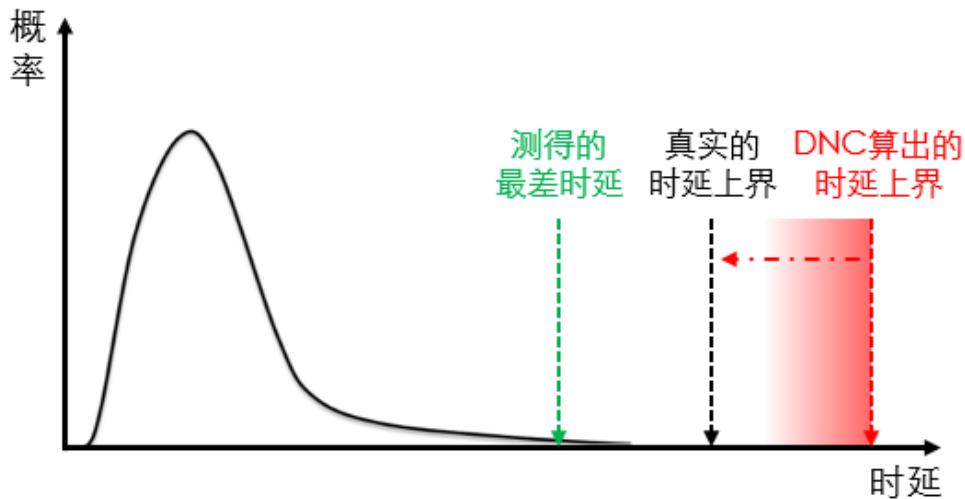


图 8 三种时延上界之间的关系

### 2.2.1 代数类算法

表 1 列出了四种代数类算法：

算法	计算方法	特点
TFA	$D_{E2E} = \sum_i D_i$	分别计算单节点时延，然后相加
SFA	$\beta_i^{l.o.} = \beta_i \ominus \alpha_{cross}, \beta = \beta_1^{l.o.} \dots \otimes \beta_n^{l.o.}$	先求剩余服务曲线，再求卷积
PM00A	$\beta = \beta_1 \dots \otimes \beta_n, \beta_i^{l.o.} = \beta \ominus \alpha_{cross}$	先将服务曲线求卷积，再求剩余服务曲线
TMA	网络分段+SFA/PM00	选取合适的切分点，分别用 SFA 和 PM00A

表 1 网络演算代数类算法

TFA (Total Flow Analysis) 算法<sup>[1][2]</sup>，也可叫做元件分解法，是确定性网络演算中最基本的一种端到端时延的计算方法。该方法的核心思路，就是沿端到端路径逐跳计算时延上界、并相加，从而得到端到端时延上界。

需要注意的是，流量在经过一个调度器后，其形态会发生变化，故需求解流量经过一个节点后的输出界 (output bound)，并以此作为下一个节点的到达曲线。还有一种解决方法是使用“逐跳整形”。即在每个节点，对流量都施加一个整形器。如流量在源端的特性可以用到达曲线  $\alpha(t)=rt+b$  描述，则整形器可使用单令牌桶整形器，分配配置承诺信息速率  $cir=r$ 、允许突发尺寸  $cbs=b$  即可。<sup>[14]</sup>证明了这种逐跳整形不会使端到端时延上界变差。

TFA 适用于任意拓扑形态，算法简单，扩展性好。但因未考虑多跳调度级联的性质等原因，计算结果的悲观性有时较大。

SFA (Separated Flow Analysis)<sup>[4]</sup>的核心思想是，对于需要求端到端时延的观察流 (foi, flow-of-interest)，先求其路径上每个节点的剩余服务曲线 (left-over service curve)、再将这些剩余服务曲线卷积得到观察流端到端的总剩余服务曲线。其中，求解一个节点的剩余服务曲线，就是考虑在该节点上干扰流对于观察流产生的调度竞争

后，观察流能得到的服务曲线。

PMOOA (Pay Multiplexing Only Once Analysis) <sup>[15]</sup> 的核心思想是，对于需要求端到端时延的观察流，先求一段路径上几个服务节点的服务曲线的卷积，再将同样通过这段路径的流量视作干扰流而在卷积后的服务曲线上求剩余服务曲线。

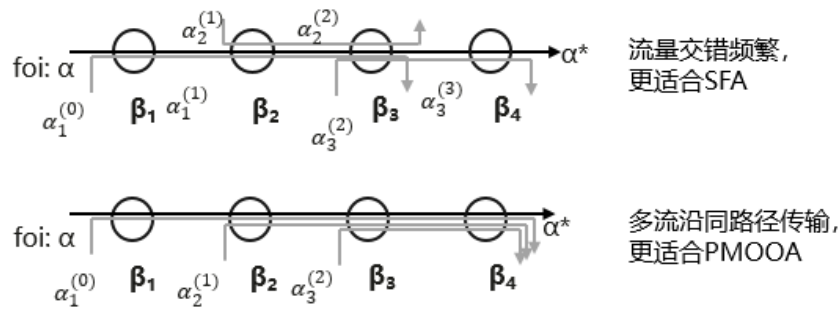


图9 SFA 与 PMOOA 的适用场景示例

如图 9 所示，概括的说，SFA 更适用于流量交错频繁的场景，但由于将干扰流的突发多次计算，损失了一定的精度；而 PMOOA 更适用于多流沿同一路径传输的场景，不过卷积丢失了不同节点的顺序细节，也会损失一定的精度。同时，SFA 与 PMOOA 也可以配合使用，即分析观察流端到端时延时，部分节点上使用 SFA、另一部分使用 PMOOA。在一些场景下，使用 SFA/PMOOA 算法可以获得比 TFA 更好的时延结果。

TMA (Tandem Matching Analysis) <sup>[16]</sup> 算法的本质就是穷举所有 SFA 与 PMOOA 组合使用的方式，并在得到的所有结果中取最好的结果。因此，TMA 理论上是所有代数类算法中，精度最高的。当然，代价是显著提高了计算复杂度。

DeepTMA <sup>[17]</sup> 是一种使用了深度神经网络来预测网络切点的优化算法，其计算精度接近于 TMA，但计算复杂度较 TMA 下降很多。DeepTMA 经过离线训练后，对于给定的观察流及拓扑信息可以快速输出建议的网络切

点，通过合理设定阈值，形成最终的网络切点序列再进行时延计算，实现计算一次、高精度输出，使得 TMA 指数级的复杂度降低到多项式级的复杂度。

### 2.2.2 优化类算法

优化类算法，如线性规划<sup>[18]</sup> (LP, Linear Program)，使用多重约束条件来描述节点行为、流量特性以及多流汇聚等行为，同时以最大时延作为目标函数，求解这个线性规划问题。

以图 10 为例，单节点上有两个流量，A 和 D 分别表示流量累积到达节点的量和累积离开节点的量。要构造线性规划问题，需要构建目标函数和约束：

- 目标函数： $\max\{t_1-t\}$ 。
- 节点服务曲线约束： $D_1(t_1)+D_2(t_1) \geq A_1(t)+A_2(t)+\beta(t_1-t)$ 。该约束的含义是，存在一个  $t$ ，使得从  $t_1$  时节点的累积输出量大于等于  $t$  时的累积到达量加  $t$  到  $t_1$  这段时间内服务曲线的取值。该约束就是对于服务曲线定义的描述，即满足不等式  $D_1(t)+D_2(t) \geq (A_1(t)+A_2(t)) \otimes \beta(t)$ 。
- 流量到达曲线约束： $A_i(t_1)-A_i(t) \leq \alpha_i(t_1-t)$ ,  $i=1,2$ 。该约束的含义是，对于任意  $t$ ，从  $t$  到  $t_1$  这段时间内，流量  $i$  的累积到达小于等于到达曲线的取值。该约束就是对于到达曲线定义的描述。
- 其它约束，如队列上报文先入先出 (FIFO) 的约束，累积函数单调性的约束等，此处不展开。

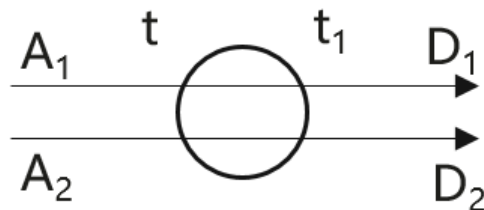


图 10 用于构造线性规划问题的单节点模型

在分析端到端时延上界时，优化类算法构建好所有约束后，直接在约束域内求解。一般来讲，优化类算法相对于代数类算法，具有更好的精度（结果的紧性更好）。同时，优化类算法的复杂度也更高。

## 2.3 随机网络演算

确定性网络演算（DNC）能够为流量提供 100% 的 worst-case 性能上界保证，同时也会一定程度上造成过度的资源预留。实际网络中的应用一般能够容许一定概率的丢包或时延性能违背。例如<sup>[19]</sup>，语音业务要求 99% 的情况下端到端时延不大于 100ms；5G URLLC 定义的离散自动化业务场景要求网络端到端时延上界 10ms，可靠性 99.99%。同时，网络中流量存在一定的随机特征，若流量之间相互独立，则大规模流量同时突发的概率很低。对此，希望能够给出“一定违背概率”下的网络时延性能上界分析。

随机网络演算（Stochastic Network Calculus, SNC）将随机过程和传统网络演算结合，刻画流量、服务的随机特征，给出概率时延上界、概率缓存上界的理论分析。如统计复用的 IP 网络中对 SLA 要求概率保证，则 SNC 能够对时延分析和资源规划提供一定的理论指导。针对 SNC 的理论探索和算法开发仍在学术界合作开发中，后续将以专门的技术文稿展开讨论，本节只给出 SNC 的一些基本概念和经典算法。

### 2.3.1 概率时延上界

SNC 可以计算一定概率下的时延上界。其中，“概率”的引入具体体现在：

- **对于流量的到达过程引入概率描述：**考虑流量的到达和突发具有随机特性，往往可以允许流量小概率违背流量上包络。这时的流量上包络，即为“概率的流量上界”。如图 11 所示，取不同的上包络（黑色

实线、点划线和虚线），则真实流量的累积到达违背流量上包络（即彩色实线到流量上包络之上）的概率明显不同。

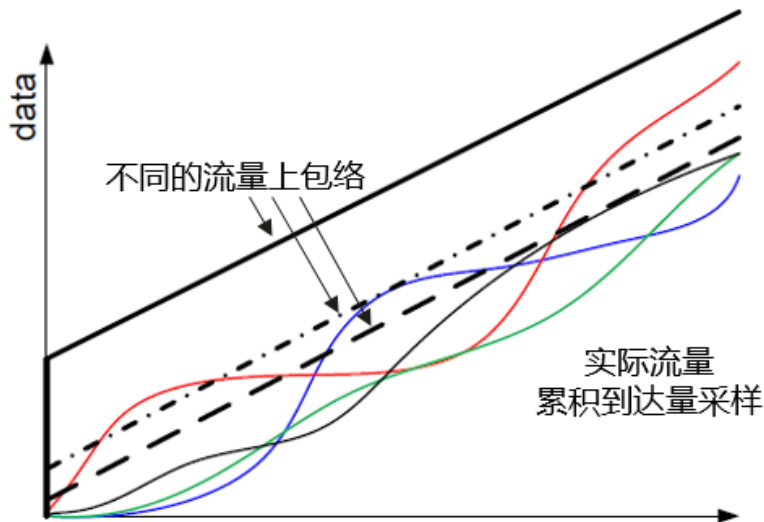


图 11 概率的流量上界

- **对于网络服务过程引入概率描述：**对于有线网络场景，即使认为服务节点具有恒定的转发速率，但由于具有随机特性的干扰流量的存在，可造成剩余的服务能力具有随机特性；对于无线通信场景，其信道具有随机衰落特性，对其服务过程建模时引用概率描述。

基于概率描述的到达过程和概率描述的服务过程进行分析计算，得到概率下的性能界，如时延和缓存。下一节介绍一种基本的概率时延界的分析方法。

### 2.3.2 基于 MGF 的 SNC 分析方法

在 SNC 中，对流量到达、服务过程可采用矩生成函数（MGF, Moment Generating Function）表示。以到达流量  $A(t)$  为例，根据其概率分布，可以求得 MGF:

$$\phi_A(\theta) \triangleq E(e^{\theta A(s,t)})$$

进而，利用不等式缩放：

$$\phi_A(\theta) \leq e^{\theta\rho(\theta)(t-s)+\theta\sigma(\theta)}$$

该不等式的含义为 MGF 在指数上具有线性的上界，称为 MGF bound 或  $\sigma$ - $\rho$  bound。例如，泊松分布、Markov Modulated on-off 分布、伯努利分布等，都具有 MGF 线性上界。

类似的，将服务描述映射到 MGF 域。如前所述，即使服务节点具有恒定的转发速率，但在网络中存在流量汇聚时，由于干扰流具有随机分布，则观察流获得的剩余服务也具有随机特征。因此，将服务节点的随机性能变换到 MGF 域，并求得其线性下界。

基于 MGF 域流量上界和服务下界，可进行流量汇聚、服务级联等运算，最终求得端到端的时延上界与违背概率有关。据此，在流量特征和网络拓扑与服务能力确定时，SNC 可以求得不同保障概率下的时延上界，例如时延 99.9% 不超过 1ms、99.99% 不超过 0.1ms 等。类似的，在一定的流量特征下，给定一个保障概率下的时延上界，则可以通过 SNC 推导求出需要配置的网络服务能力，如所需为该流量分配的网络带宽。

## 2.4 网络演算在 AFDX 网络中的应用

航空电子全双工交换式以太网（AFDX, Avionics Full Duplex switched Ethernet）是目前大中型飞机的机载骨干互连技术，如空客 A380、A350，波音 B787 等客机都使用了 AFDX。AFDX 在传统以太网的基础上进行升级改造，网络演算被用于 AFDX 网络中数据传输排队的最大时延评估，从而为机载网络应用的性能保证提供具体的设计参考。

AFDX 使用了虚拟链路（Virtual Link, VL）的概念，一个流量使用一个 VL，多路 VL 可以共用一条物理链路。对每条 VL，定义了带宽分配间隔 BAG（Bandwidth Allocation Gap）和最大帧长度  $S_{\max}$ 。AFDX 帧在

进入 AFDX 网络时，受到整形，从而使得进入网络的帧间距不小于 BAG。由此，一个 AFDX 流量的到达曲线就可以很容易地构建出来，如图 12 所示。

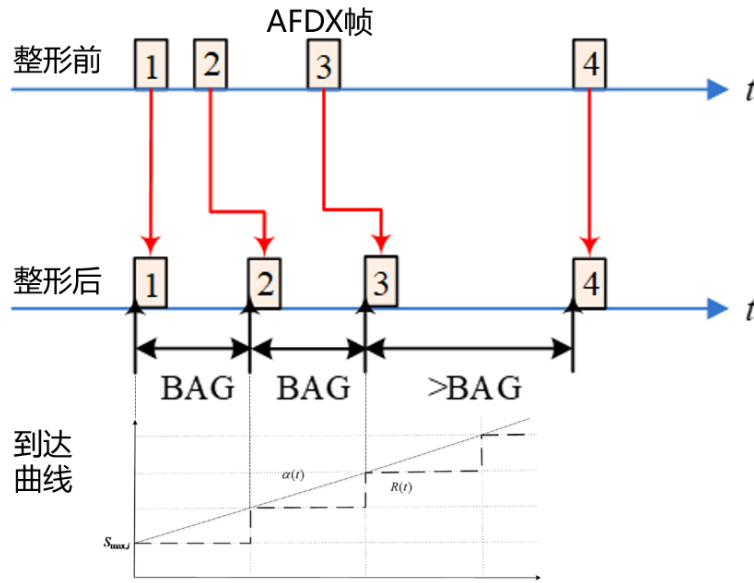


图 12 AFDX 中流量到达曲线与 BAG、最大帧长的关系

用公式表示到达曲线：

$$\alpha_i(t) = \sigma_i + \rho_i t = \frac{S_{max,i}}{T_i} \times t + S_{max,i}$$

同时，AFDX 网络中的交换机上使能了高、低两个优先级的调度。对需要时延保障的流量分配高优先级，其服务曲线的推导过程在此不展开说明。<sup>[20]</sup>

由此，网络演算可以计算 AFDX 中流量的端到端时延上界。AFDX 方案必须通过适航验证，使用 AFDX 的飞机才被允许起飞。适航验证中关于网络演算的具体过程，就是由飞机的设计方使用网络演算理论，向适航验证当局（如美国联邦航空管理局、欧洲航空安全局）证明各流量在当前 AFDX 设计下的端到端传输时延上界均不大于其传输需求。



在实际的 AFDX 应用中，网络规模为 100+端系统、8 个左右的交换机、千余条 VL。具体使用的是基于 TFA/元件分解法的网络演算算法。

AFDX 网络的关键设计在于，使用了边缘的准入控制（或者说是边缘整形），使得流量的到达上界可知可控。从而网络演算可以得到一个合理的结果。

### 3 TSN 调度机制与网络演算的关系

#### 3.1 基于经验的做法

TSN 最早的一个应用标准，是 IEEE 802.1BA，面向音视频桥接网络的应用场景。背后使用的具体的 TSN 调度整形技术，是由 IEEE 802.1Qav 定义的，一般称其为 CBS (Credit-based Shaping)。在 IEEE 802.1BA 中，有如下这样一个表格：

Table 6-2—Latency targets for SR classes A and B

SR class	Max end-to-end latency
A	2 ms
B	50 ms

图 13 IEEE 802.1BA 中对于时延目标的定义

SR class A 指在最高优先级下使用 CBS 整形机制的那一类流量；SR class B 指在第二高优先级下使用 CBS 整形机制的那一类流量。而达成表格中时延保障目标的预设条件是：1) 7 跳；2) 100Mb/s 的以太网带宽；3) 最大帧长是 1522 字节。并且标准也明确说明，这两个目标时延，

反映了典型场景下的需求，但不应被认为是一种强制的端到端保障性能。

IEEE 802.1BA 中同时给出了一种时延计算公式，通过四则运算就能获得最大时延。简单理解是通过已经资源预留了的带宽占总带宽的比例，估算流量之间竞争的影响，然后加上设备的固定时延、等待低优先级一个最长报文传输的时延等等，得到最大时延的结果。这种技术方法在后面和网络演算一起来展开说明。

TSN 第二个应用标准，就是 IEEE 802.1CM，用于移动前传网络。该标准中制定了两套方案。方案 A 中，使用严格优先级（SP，Strict Priority）调度，为高优先级前传（HPF，High Priority Fronthaul）数据、中优先级前传（MPH，Medium Priority Fronthaul）数据、低优先级前传（LPF，Low Priority Fronthaul）数据分别提供不超过 100 微秒、1 毫秒、100 毫秒的端到端单向时延。在方案 B 中，引入了帧抢占（Frame Preemption，由 IEEE 802.1Qbu 定义）后，为三类数据提供的服务不变，但取消了对非前传流的帧长的限制。值得注意的是，上述的 100 微秒中，包含了传输距离 10 公里下的大约 50 微秒的链路传输时延。同 IEEE 802.1BA 类似，IEEE 802.1CM 里的时延指标，本质上也是一种经验值，或者说是参考值。

IEEE 802.1CM 中也给出了一种最高优先级的时延计算公式，通过四则运算就能获得最大时延。这种技术方法在后面和网络演算一起来展开说明。

### 3.2 SP 的时延计算

SP，Strict Priority，绝对优先级调度。

以下是 IEEE 802.1CM 中给出的计算公式。

$$t_{MaxBridge} = t_{SF} + t_{SelfQueueing} + t_{Queueing} + t_{MaxGoldFrameSize+Pre+SFD+IPG} \quad (7-1)$$

where

$t_{SF}$  is the store-and-forward delay (item e) in 7.1) of the bridge

$t_{SelfQueueing}$  is the self-queueing delay (item g) in 7.1)

$t_{Queueing}$  is the queueing delay (item f) in 7.1)

$t_{MaxGoldFrameSize+Pre+SFD+IPG}$  is the transmission time for a maximum size frame (MaxGoldFrameSize) of a gold flow with Preamble (Pre), Start Frame Delimiter (SFD), and the following Inter Packet Gap (IPG)

上面 7-1 公式，是一种直观而朴素的时延计算方法。对比图 7 来看，网络演算的单节点时延  $D=T+b/R$ 。在简单场景下，两者是很相似的。

$t_{SelfQueueing}$  这一项，相当于  $b/R$ 。而其它项之和，相当于  $T$ 。感兴趣的读者，可以对照标准原文进行理解。

运用网络演算，可以将对于使用 SP 调度的时延上界计算，扩展到非最高优先级、以及多节点端到端的级联计算（而非逐跳相加，这样计算会有较大误差）。这里就需要具体参考第 2 章中，网络演算服务曲线的构建和几种确定性网络演算算法。

### 3.3 CBS 的时延计算

CBS, Credit-based Shaping, 基于信用的整形。

以下是 IEEE 802.1BA 中给出的计算公式。

$$\text{Max Latency} = t_{Device} + t_{MaxPacketSize+IPG} + (t_{AllStreams} - t_{StreamPacket+IPG}) \times \text{Rate}/\text{MaxAllocBand} + t_{StreamPacket}$$

where

$t_{Device}$  = the internal delay of the device (in increments of 512 bit times)

NOTE 3— $t_{Device}$  is an integral multiple of 512 bit times so that it scales with the speed of the media.

$t_{MaxPacketSize+IPG}$  = the transmission time for a maximum size interfering frame (1522 octets to 2000 octets) plus its preamble and start of frame delimiter (SFD) (8 octets), and the following inter-packet gap (IPG) (12 octets)

$t_{StreamPacket}$  = the transmission time for the maximum frame size of the stream that is being reserved, plus its preamble and SFD (8 octets)

$t_{StreamPacket+IPG}$  = the transmission time for the maximum frame size of the stream that is being reserved, plus its preamble and SFD (8 octets) and the following IPG (12 octets)

**Rate** = the transmission rate of the port

**MaxAllocBand** = maximum allocatable bandwidth, the maximum amount of bandwidth the AVB system is able to allocate for Class A streams on the port

$t_{Interval}$  = the Class A observation interval or 125  $\mu$ s

$t_{AllStreams}$  =  $(\text{MaxAllocBand} \times t_{Interval}) / \text{Rate}$  = the sum of the transmission times of all Class A stream frames the AVB System is able to allocate in an observation interval (125  $\mu$ s) on a port

同样的，这也是一种直观而朴素的时延计算方法。那么我们来看网络演算是如何建模分析 CBS 的时延上界的。

在 AVB 网络中，主要设置两类优先级流量进行 CBS (Credit Based Shaper) 整形，即 Class A 和 Class B 流量 (A 的优先级高于 B)；Class A 和 Class B 流量在传输排队时依赖两个带宽保障参数  $idleSlopeA/B$  和  $sendSlopeA/B$ ，其中， $idleSlopeA/B$  是 Class A/B 流量在节点处等待传输时信用量的增加速率， $sendSlopeA/B$  是 Class A/B 流量正在传输时信用量的减小速率， $C$  为链路的总带宽传输速率。

流量在 Talker 处以周期  $T$  时间产生并发送进入网络的流量可建模为  $(r, b)$  漏桶模型，以及考虑到网络链路总带宽  $C$  的输出约束能力；流量到达使用到达曲线描述，周期性流量到达曲线定义为，

$$\alpha(t) = \min\{Ct, b + rt\}$$

其中， $b$  为流量的最大突发量，一般可认为是流量的最大帧长  $L_{A/B}$ ； $r$  为流量的最大可持续稳定速率，由此得出到达曲线。

下面给出服务曲线描述。具体而言，对 A 类流量，

$$\beta_A(t) = I_A \left[ t - W_u(t) - \frac{L'_A}{C} \right]_{\uparrow}^+ \quad (4.13)$$

对 B 类流量，

$$\beta_B = I_B \left[ t - W_u(t) - \frac{\left( L_{BE} + L_A - \frac{L'_A I_A}{S_A} \right)}{C} \right]_{\uparrow}^+ \quad (4.14)$$

其中， $I_A$ 、 $I_B$  分别为  $idleSlopeA$ 、 $idleSlopeB$ ， $S_A$  为  $sendSlopeA$ ； $L_A$ 、 $L_B$ 、 $L_{BE}$  分别代表 A、B、BE 类最大帧长， $L'_A$  代表优先级低于 A 类流量的最大帧长，也即  $L'_A = \max\{L_B, L_{BE}\}$ ； $W_u(t)$  代表为 CDT 类分配的时间

窗口上界。

### 3.4 ATS 的时延计算

ATS (Asynchronous Traffic Shaper) 是一个异步数据流整形器，它旨在通过每跳重塑 TSN 数据流，ATS 并不要求桥和终端节点同步，对于高实时要求和非实时业务混合业务模式下，ATS 也能保持带宽的最大利用率。ATS 采用基于令牌桶的核心算法将合格时间分配给属于特定流的帧，然后由 ATS 传输选择算法用于流量调节。

ATS 采用的核心算法 ATS 算法是基于令牌桶算法进行流量整形，该整形器为一个贪婪整形器，对于每一类流量，令牌桶中令牌量和增长速率是相对独立的，所以对于各类型流量来说，ATS 对其提供的整形曲线（服务曲线）为：

$$\beta_{ATS}^x(t) = shaper(t) = CBS^x + CIR^x * t$$

ATS 所提供的整形曲线受  $CBS^x$  和  $CIR^x$  两个参数影响，参数的配置决定了 ATS 的服务能力。但如果 ATS 所提供的整形曲线小于流量的到达曲线，也就是整形器的服务能力要小于流量的到达，ATS 也会按整形曲线对流量整形，但需要付出延迟的代价，此时流量经过 ATS 所产生的延迟为：

$$D_{ATS}^x = h(\alpha^x, \beta_{ATS}^x)$$

流量离开的突发由整形器限制，即： $\alpha^*(t) = \min\{\alpha(t), shaper(t)\}$ ，作为下一跳的到达曲线。据此可以计算端到端时延上界。

IEEE 802.1Qcr-2020 中给出了另一种时延计算方法：

A delay bound  $d_{BU,max}(k, f)$  on the buffering delay for a single hop in absence of the effects covered in subsequent clauses is known (see Specht and Samii [B87]). For  $k < n$  (i.e., all but the last hop to the Listener), the delay bound  $d_{BU,max}(k, f)$  is given by Equation (V-2).

$$d_{BU,max}(k, f) = \max_{h \in F_S(k, f)} \left\{ \frac{\sum_{g \in F_H(k, h) \cup F_S(k, h)} b_{max}(k, g) - l_{min}(h) + l_{LP,max}(k, h)}{R(k) - \sum_{g \in F_H(k, h)} r_{max}(k, g)} + \frac{l_{min}(h)}{R(k)} \right\} \quad (V-2)$$

For  $k = n$  (i.e., the last hop to the Listener), this bound is given by Equation (V-3).

$$d_{BU,max}(n, f) = \frac{\sum_{g \in F_H(n, f) \cup F_S(n, f)} b_{max}(n, g) - l_{min}(f) + l_{LP,max}(n, f)}{R(n) - \sum_{g \in F_H(n, f)} r_{max}(n, g)} + \frac{l_{min}(f)}{R(n)} \quad (V-3)$$

where

$F_H(k, f)$ and $F_H(k, h)$	denote the set of streams transmitted in a numerically higher traffic class (8.6.8) than stream $f$ and a stream $h$ , respectively, at the upstream transmission Port of the $k$ th hop
$F_S(k, f)$ and $F_S(k, h)$	denote the set of streams transmitted in the same traffic class as stream $f$ , including stream $f$ and stream $h$ , respectively, at the upstream transmission Port of the $k$ th hop
$l_{LP,max}(k, f)$ and $l_{LP,max}(k, h)$	denote the maximum interference length, in bits, by any numerically lower traffic class than the class of stream $f$ and a stream $h$ , respectively, at the upstream transmission Port of the $k$ th hop
$l_{min}(f)$ and $l_{min}(h)$	denote the minimum frame length of stream $f$ and a stream $h$ , respectively, in bits, including all media-dependent overhead (8.6.11.3.11, 12.4.2.2)
$b_{max}(k, g)$	is the maximum burst size associated with a stream $g$ at the $k$ th hop, in bits
$r_{max}(k, g)$	is the committed information rate of stream $g$ at in the upstream device of the $k$ th hop, in bits per second
$R(k)$	is the transmission rate, in bits per second, that the underlying MAC Service that supports transmission through the upstream transmission Port of the $k$ th hop provides

看起来很复杂，但是别害怕，很好理解。看公式 (V-2)，主要是第一个分式，分子第一项是所有流的突发 cbs，第三项是低优先级长帧的影响，而这两项需要被总速率减去所有更高优先级流量的总 cir（即分母）所“消化”，（分子第二项是一个短报文，这里减掉，第二个分式又加回来，只是在修正消化的速率）而这个消化的最长时间，就是单跳上最大的时延。对这种计算方法的详细介绍，可以参考<sup>[24]</sup>。

这种计算方法可以成立的重要原因，在于 ATS 机制对流量进行了逐跳重整形，因此流量的突发 cbs 和速率 cir 在每一跳上都是已知量。在计算的本质上，与网络演算中的 TFA 算法一致（见 2.2.1），但是如前所述，TFA 有它的缺陷。

### 3.5 关于时间门控

一般意义上，使用时间门控（scheduling, TAS），端到端时延就直接受门控的规划而给出。当然，时间门控可以和上述几种机制混合使用。例如，每个大的门控周期中，划分一部分时隙给周期性的的重要流量使用，而剩余时隙不进行精细化编排，而是对其它流量使用 QoS、异步流量整形机制等。这样，这些流量的时延上界也可以通过网络演算进行分析。简单的理解是，需要在服务曲线的速率上，减去关键周期流量的门控所占用的比例，而和横坐标的交点  $T$ ，可以加上一次关键周期流量的门控打开时长即可（可对照图 7）。受篇幅限制，这里不再展开。

## 4 应用网络演算的 TSN 方案

### 4.1 服务与流量建模

为方便读者阅读，把图 7 复制一份在这里：

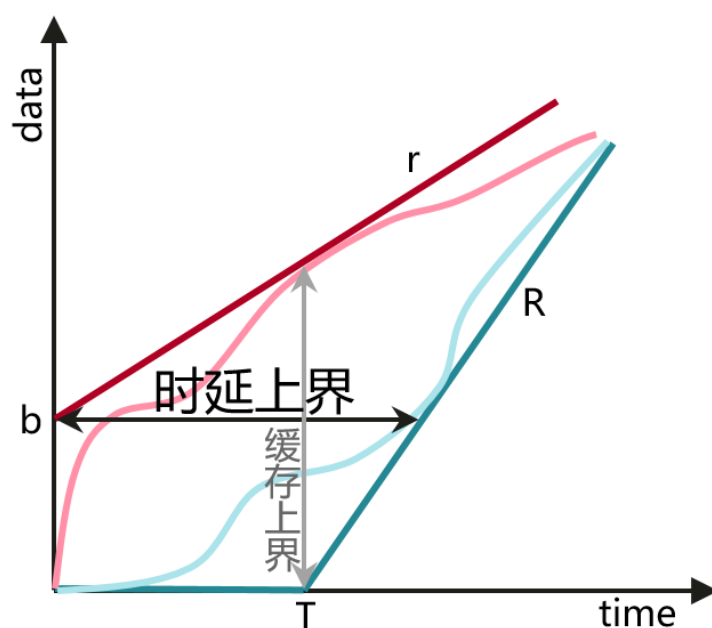


图 14 时延上界与缓存上界

由于具体的网络设备在工程上的差异，进行服务曲线的建模时，不仅需要考虑到调度器、整形器的理论特征，还要考虑设备的工程差异。例如，设备处理报文转发，一般会引入  $\mu\text{s}$  量级的处理时延，并且在不同工况下，处理时延可能有较大的差异。一种简单的方法，是多次对设备进行测量，尽量覆盖所有工作场景，包括轻载、重载等，使用测得的最大处理时延，然后加到服务曲线原有的  $T$  上；再例如，设备实际进行整形器操作、轮询调度操作时，因为芯片处理能力的差异，或者是调度过程中的工程实现和理想调度器模型存在差异，也会造成实际的服务曲线和理论中的产生偏差，一般也可以通过工程手段进行修正。

对于流量的到达曲线建模，2.1.1 中已经给出一种案例，并且这种案例对于绝大多数周期性特征显著的流量都适用。对于一般的流量，我们也可以通过突发和平均速率的方式来描述（二参数），或者用突发、最大报文长、峰值速率、平均速率的方式来描述（四参数）。下图可以



看出，参数越多，描述可以越准确，时延计算结果的悲观性也就越小（越好）。

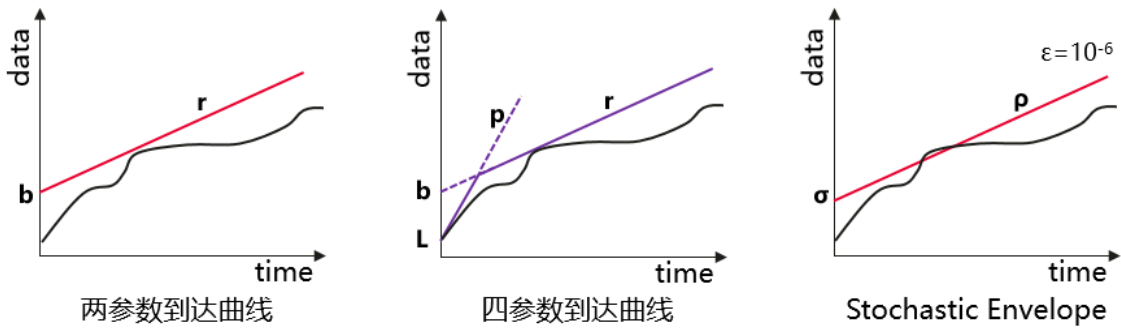


图 15 不同的到达曲线

## 4.2 网络演算与 TSN 配置模型、TSN 方案的关系

下图以工业网络场景作为一种案例，使用集中式配置模型（纯集中式，或集中式网络分布式用户），进行说明。

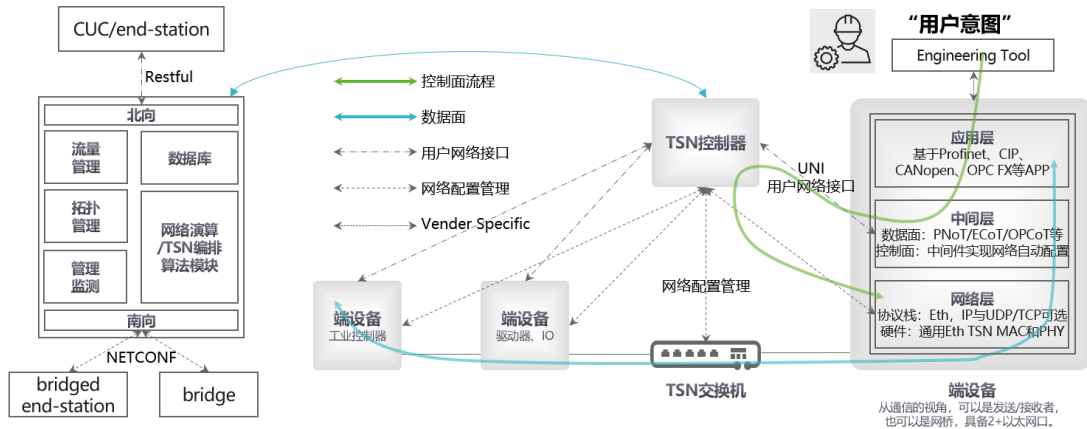


图 16 TSN 方案示例

TSN 集中式配置模型的核心是网络控制器，即图中 TSN 控制器（从功能的角度也叫 CNC，Centralized Network Configuration）。而 TSN 控制器除了南北向接口以及相关功能单元外，比较重要的一个模块就是

算法模块。网络演算可以是 TSN 控制器算法模块的重要组成部分（当然，算法模块还可以包含 TSN 门控编排算法等）。

算法模块输出的结果，例如门控配置、整形参数配置等，通过 TSN 控制器的南向接口，可以配置到 TSN 网络设备上，从而实现对流量的服务保障——这一点很好理解。但是算法需要哪些输入呢？

具体以网络演算的算法为例，如前所述，需要网络的信息和流量的信息。

- 网络信息：一方面是网络拓扑，获取网络拓扑是一般网络控制器所具备的能力，具体会依赖于LLDP等协议；另一方面是设备的服务能力的建模，可以参考4.1中的内容，提前对设备完成建模，在TSN控制器中维护一个设备模型库，在有需要进行更新，就可以了。

- 流量信息：按照TSN标准的规定，需要由用户方，通过用户网络接口（UNI），把流量的特征信息、需求等告知网络控制器。这里所谓的用户方，可以是具体的每个端设备，也可以是端设备的代表，从功能的角度叫CUC（Centralized User Configuration）。如果现有的端设备，不具备这种能力，则可以通过构建、升级一个软件中间件（中间层），将来自用户不同应用的行为和需求，转换为标准UNI的数据模型，与TSN控制器进行交互。

由此，一个完整的 TSN 方案构建流程如下：

用户将意图通告给 TSN 控制器（其中包括流量的特征、对服务的要求等，具体可经由 TSN 中间件），TSN 控制器基于当前所有用户的流量信息和网络状态，运行网络演算、门控编排等算法，得到的结果一方面配置给网络设备，一方面告知端设备。此后，端设备发出的流量，可以得到 TSN 网络的服务保障。

## 附录 A 确定性方案的本质探讨

### 对确定性方案的分类

网络发展的历史中，QoS 需求一直存在。最早有 ATM 方案，后来 IP 流行后，有 IntServ 等 QoS 方案。目前，IP 网络中主流还是 diffServ，虽然也可以区分优先级，但本质是 BE，即使是高优先级流量，也没有可量化的 QoS 保障，具体而言就是没有可量化的确定性保障。新技术如 TSN 中的 CBS、TAS，5G 中提出的网络切片概念等，带来了新的确定性的讨论。TSN 关注的确定性，主要指时延上界的保障，其次是平均时延和时延抖动。

网络中的一部分流量需要时延上界的保障。我们总结了为此而设计的 3 大类确定性方案：QoS, Scheduling, Slicing。不论哪个方案，在应用时，需要通过仿真或网络演算等理论分析的方法确定方案具体为特定流量能提供怎样的确定性服务。这里所讨论的确定性，或 QoS 保障，如不经特殊声明，则指流量端到端传输的时延上界，即最差情况下的端到端时延。

#### ● QoS（也可称为异步调度类机制）

IntServ 是一种经典的 QoS 机制，由 IETF 提出，如通过 RSVP 为特定的流量在传输路径的网络设备上进行资源预留，预留后的带宽由特定流量独享，从而网络为该特定流实现 QoS 保障。IEEE 802.1Qav 定义了一种基于信用的整形机制 CBS，IEEE 802.1Qcr 正在定义一种异步流量整形机制。这两种机制的本质思想和 IETF IntServ 相同，都需要在进行资源预留后，网络为特定流量实现 QoS 保障。具体体现在：转发设备上，识别报文，归入相应 class、进入相应队列，由于存在资源预留，

相应队列中的调度竞争存在约束，故而报文的转发存在保障。

CBS 在 AVB 网络中有着成功的应用。另外，AFDX 是一种典型的使用 none time based QoS 对流量提供确定性时延保障的技术，且在航空机载网络上有着成功的商业应用。

当前网络设备中普遍适用轮询调度，只要确定规则，亦可通过轮询调度得到一定的确定性保障。

● Scheduling（也可称为同步调度类机制、时间门控调度、时间感知整形等）

IEEE 802.1Qbv 定义了一种基于时间片对队列进行开关门操作的机制。这种机制可以用在二层桥接转发设备或三次路由转发设备上，实现一种基于时间片的调度方法：在转发设备上，识别报文，归入相应 class、将报文放入相应队列；队列上的开关门状态经过设计，换言之，资源预留。具体的设计，可以是端到端为流量都规划好具体的时间片，也可以仅仅将时间门控作为一种整形机制。（理论上，将基于时间门控的调度用到极致，则在性能上可以向 ATM 逼近，其差异在于基于时间门控的调度的最细粒度是逐报文进行的。例如以太帧，报文长度为 64-1522 字节。）

TTEthernet 中对于 TT 流量的 QoS 保障、Profinet IRT 中对于 IRT 流量的确定性保障，均是 Scheduling 类机制。

● Slicing（这里特指逻辑隔离网络，也可称为硬网络切片）

其思想是，将一个物理网络分成多个逻辑网络，具体到一条链路上，就是把一条物理链路切分成多条互不干扰的逻辑链路，多条子链路的带宽总和约等于物理链路的带宽。

Slicing 类方案保障确定性的核心，就是多个切片上流量传输互不

干扰。例如，在一条 10G 链路上，切出一条 1G 的切片，专门用于传输某种业务流。实现网络切片可以使用多种具体的方法，如信道化子接口、FlexE 等。

以上三种方法都可以为特定流量提供时延上界的保障，但各自有优缺点：QoS 由于统计复用特性好，故效率高，但抖动高（确定性差），且对于关键流量需要理论计算时延上界，这对于车载、机载网络等局域网场景是可行的，但对于超大规模的运营商网络，会造成一定程度上的易用性下降；Scheduling 由于可以基于时间精确设计，故确定性高，一般对于网络硬件有特殊要求，且需要求解门控表，比较适合如以周期性流量为主的工业现场网络，而在许多其它场景上会存在一定的易用性问题，部分门控流量不能统计复用（效率较低）；slicing 由于逻辑子链路间不冲突，无干扰流量，故易用性好，但“VIP”独享大量带宽（效率低），且并没有保障低抖动的机制（确定性较差）。

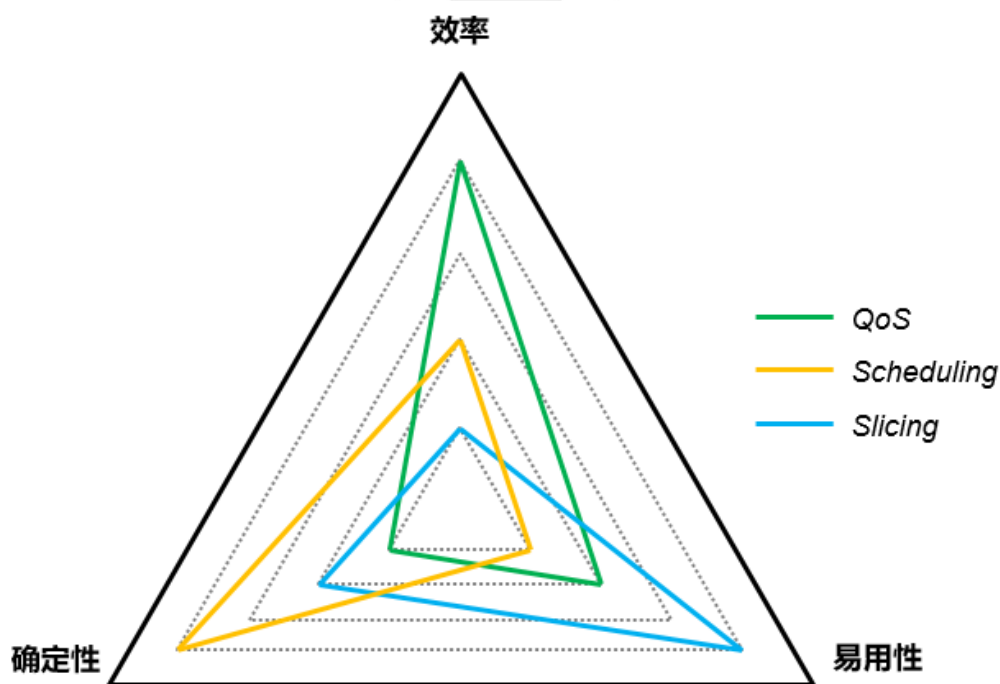


图 17 确定性方案性能三角

上图中，我们将效率、确定性、易用性列为确定性方案性能三角的三个重要性能指标。对于其评分，只能做定性分析，因为随实际场景不同，这些方案的适用性也会有差异。提出确定性方案性能三角的主要目的，是希望强调，没有哪种 QoS 方法是“神”。

### 运用网络演算进行性能分析

下面我们运用网络演算的概念，对几种 QoS 方法的性能进行比较。

观察流（需要时延保障） $\alpha = rt + b$

$r$  为平均速率， $b$  为突发量

干扰（其它）流（需要时延保障） $\alpha_H = r_H t + b_H$

$r_H$  为总平均速率， $b_H$  为总突发量

干扰（其他）流（不需时延保障） $\alpha_L = r_L t + b_L$

$r_L$  为总平均速率， $b_L$  为总突发量

服务曲线 $\beta = R(t - T)$

$R$  为端口速率/带宽

$L_L$  为低优先级最长包。

现网中流量众多，取任意一个流作为观察流，则：

$$r \ll r', b \ll b'$$

下面所求的均为最差排队时延，即不包含处理时延等。

➤ A. 同优先级混流（FIFO）： $D_{WC-FIFO} = \frac{b+b_H+b_L}{R} \approx \frac{b_H+b_L}{R}$

观察流的 worst case 时延约正比于干扰流总突发量，与干扰流总流量无关（只要总流量不超过带宽）。

➤ B1. QoS 下，观察流为 SP 高优先级： $D_{WC-SP-H} = \frac{b+b_H+L_L}{R} \approx \frac{b_H}{R}$

观察流的 worst case 时延约正比于同为高优先级的干扰流总突发量。

➤ B2. QoS 下，观察流为 SP 高优先级+DRR:  $D_{WC-SP-H} = \frac{b_X}{R} \div \frac{Q_X}{\Sigma Q} +$

$$T \approx \frac{b_X}{R} \div \frac{Q_X}{\Sigma Q}$$

BX 为和观察流使用同一个高优先级 DRR 队列的其它干扰流量的总突发。QX 是本队列轮询权重。（这里忽略掉一项，当一个 DRR 队列里的总突发）观察流的 worst case 时延正比于同 DRR 队列中干扰流总突发量。

[13] [22]

➤ C. Slicing 下，观察流使用带宽为  $R_H$  的切片:  $D_{WC-SLICE-H} = \frac{b+b_H}{R_H} \approx \frac{b_H}{R_H}$

观察流的 worst case 时延正比于同样使用该切片的干扰流总突发量。但受切片带宽变小影响而整体变大。

➤ D1. Scheduling 下，观察流享受时间片 (CQF):  $D_{WC-CQF-H} = (2+2) \times T_c \approx \frac{4 \times b_H}{R}$

设 3 缓存轮转，其中一个缓存是分配给高优先级流量的，那么最差情况有可能需要额外多等 2 次缓存轮转。[21]

➤ D2. Scheduling 下，观察流享受时间片 CQF+SP:  $D_{WC-CQF-H} = 2 \times T_c \approx \frac{2 \times b_H}{R}$

设 3 缓存轮转，其中高优先级流量可能进入任意缓存。

举例：设观察流  $b=1$ 、 $r=1$ ，高优先级干扰流  $b_H=99$ 、 $r_H=99$ ，低优先级干扰流  $B_L=500$ ，低优先级最长报文  $L_L=0.1$ ；使用 SP+DRR 时，高优先级流量平分为 4 分、分别进入 4 个等权重的 DRR 轮询队列；使用 Slicing 方案时，高优先级享受 1/3 总带宽的切片（下图 LSN），或高优先级享受 1/2 总带宽的切片（下图 LSN+）；使用 CQF 时，三缓存轮转，其中一个缓存是给高优先级使用的。如此设置，保障公平对比。得到如下结果：

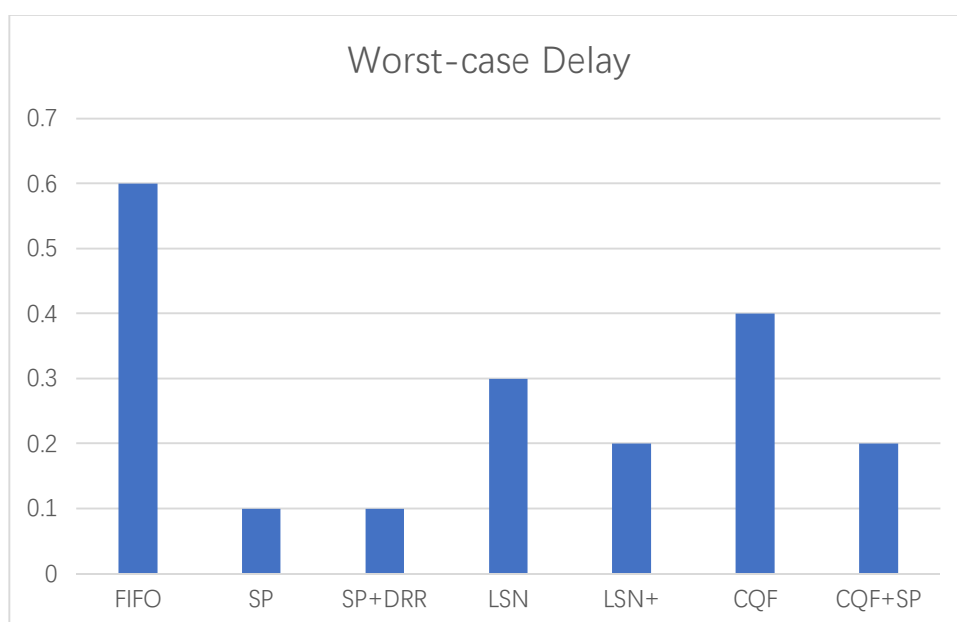


图 18 几种具体确定性方案在特定条件下的最差时延

几种方案中，高优先级流量的最差时延相比于 FIFO 都有明显改善。QoS 优于 Slicing，也印证了性能三角中切片方案在 efficiency 上的不足（想要相同的时延服务，则切片方案需要更多的总带宽）。SP+DRR 的效果和 SP 一致，因为当前是向 DRR 队列里平均分配权重和高优先级总突发的，则 DRR 可以更灵活（因为不同 DRR 队列的突发影响被隔离）。虽然切片也有类似隔离突发的能力，但 DRR 的优点在于，可以让高优先级占用到 100% 的带宽。

### 整形的影响

在 TSN 中，不论是基于信用的整形（CBS）、异步流量整形（ATS），还是时间感知整形（TAS），我们都注意到这里有一个关键词：整形。

其实，对于一个流量，整形是会引入时延的。例如将一个流量的突发  $b$  整形为  $b/N$ ，则该流量需要额外承受的最大整形时延：

$$D_{shaping\_max} = \frac{b - \frac{b}{N}}{r}$$



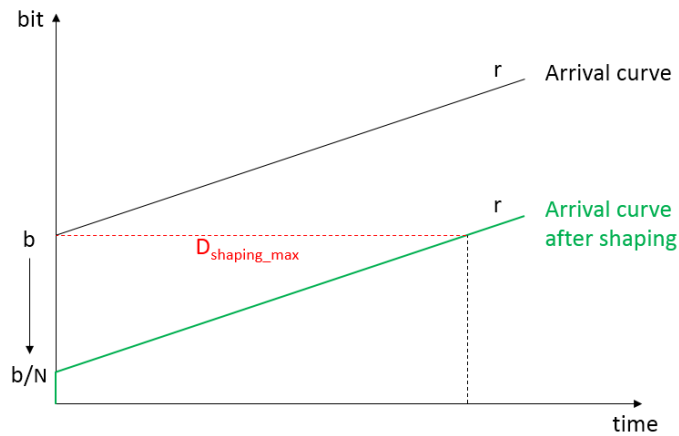


图 19 从网络演算的角度分析整形引入的时延

[23] 中对于整形的影响给出了具体分析。虽然整形本身引入时延，但是对于网络整体的时延性能可以起到改善的作用。如下图，统一对流量进行整形，把突发整形为原来的  $1/10$ 。对于一个突发和速率均占总流量  $1/100$  的流，在 1 跳上，整形所带来的 WCD（最差时延）的改进，就抵消了整形本身引入的时延，即图中点  $(100, 1)$ 。而随着跳数的增加，WCD 改进的效果会越来越明显。

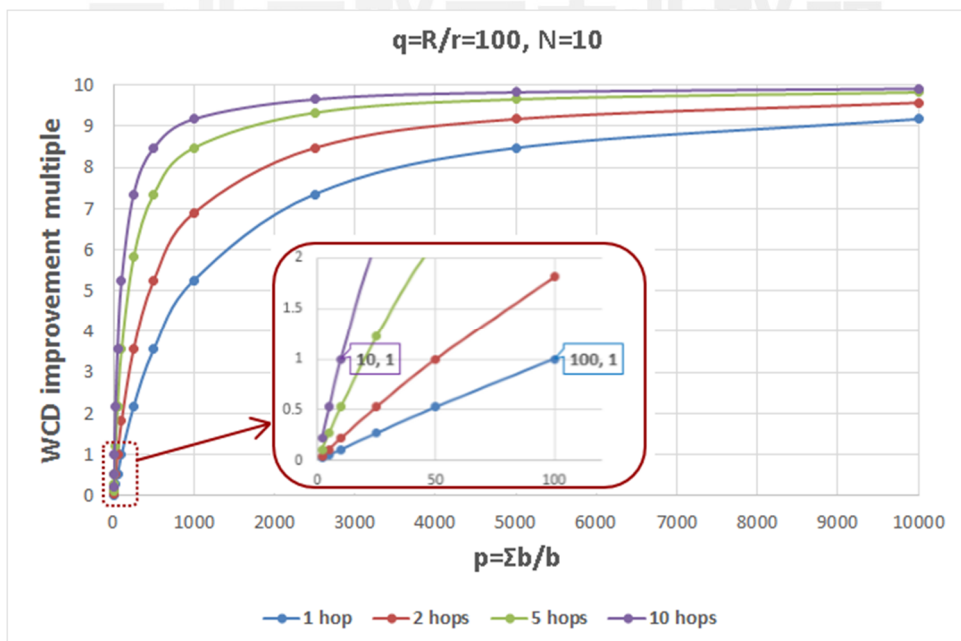


图 20 整形对于整体网络性能的改善

如上所述，TSN 中提供的几种整形机制，各有各的优势，并且可以通过网络演算进行量化评估。而通过网络演算进一步的分析可以发现，不论应用哪种整形机制，都对网络的整体性能有进一步改善的效果。



**工业互联网产业联盟**  
Alliance of Industrial Internet

## 参考文献

[1] R. L. Cruz, "A calculus for network delay. I. Network elements in isolation," in IEEE Transactions on Information Theory, vol. 37, no. 1, pp. 114–131, Jan. 1991, doi: 10.1109/18.61109.

[2] R. L. Cruz, "A calculus for network delay. II. Network analysis," in IEEE Transactions on Information Theory, vol. 37, no. 1, pp. 132–141, Jan. 1991, doi: 10.1109/18.61110.

[3] C.-S. Chang: "Performance Guarantees in Communications Networks," Springer, 2000.

[4] J.-Y. Le Boudec and P. Thiran, "Network calculus: a theory of deterministic queuing systems for the internet," Vol. 2050, Springer Science I&Business Media, 2001.

[5] A. Bouillard, M. Boyer, and E. Le Corronc, "Deterministic Network Calculus: From Theory to Practical Implementation," Wiley-ISTE, 2018

[6] Y. Jiang and Y. Liu, "Stochastic Network Calculus," Springer, 2008.

[7] A. Van Bemten and W. Kellerer, "Network calculus: A comprehensive guide." 2016.

[8] M. Fidler and A. Rizk, "A Guide to the Stochastic Network Calculus," in IEEE Communications Surveys & Tutorials, vol. 17, no. 1, pp. 92–105, Firstquarter 2015, doi:

10.1109/COMST.2014.2337060.

[9] F. Frances, C. Fraboul, and J. Grieu, "Using Network Calculus to optimize AFDX network," in Proceeding of the 3thd European congress on Embedded Real Time Software (ERTS06), January 2006, Toulouse.

[10] M. Boyer and C. Fraboul, "Tightening end to end delay upper bound for AFDX network calculus with rate latency FIFO servers using network calculus," 2008 IEEE International Workshop on Factory Communication Systems, Dresden, 2008, pp. 11–20, doi: 10.1109/WFCS.2008.4638728.

[11] S. Kerschbaum, K. Hielscher and R. German, "The need for shaping non-time-critical data in PROFINET networks," 2016 IEEE 14th International Conference on Industrial Informatics (INDIN), Poitiers, 2016, pp. 160–165, doi: 10.1109/INDIN.2016.7819151.

[12] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round-robin," in IEEE/ACM Transactions on Networking, vol. 4, no. 3, pp. 375–385, June 1996, doi: 10.1109/90.502236.

[13] M. Boyer, G. Stea and W. M. Sofack, "Deficit Round Robin with network calculus," 6th International ICST Conference on Performance Evaluation Methodologies and Tools, Cargese, 2012, pp. 138–147, doi: 10.4108/valuetools.2012.250202.

[14] J. Le Boudec, "A Theory of Traffic Regulators for

Deterministic Networks With Application to Interleaved Regulators," in IEEE/ACM Transactions on Networking, vol. 26, no. 6, pp. 2721–2733, Dec. 2018, doi: 10.1109/TNET.2018.2875191.

[15] M. Fidler, "Extending the Network Calculus Pay Bursts Only Once Principle to Aggregate Scheduling," In: Marsan M. A., Corazza G., Listanti M., Roveri A. (eds) Quality of Service in Multiservice IP Networks. QoS-IP 2003. Lecture Notes in Computer Science, vol 2601. Springer, Berlin, Heidelberg

[16] S. Bondorf, "Quality and Cost of Deterministic Network Calculus - Design and Evaluation of an Accurate and Fast Analysis, Measurement and Analysis of Computing Systems," No.16, 2017

[17] F. Geyer and S. Bondorf, "DeepTMA: Predicting Effective Contention Models for Network Calculus using Graph Neural Networks," IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, Paris, France, 2019, pp. 1009–1017, doi: 10.1109/INFOCOM.2019.8737496.

[18] A. Bouillard, L. Jouhet and E. Thierry, "Tight Performance Bounds in the Worst-Case Analysis of Feed-Forward Networks," 2010 Proceedings IEEE INFOCOM, San Diego, CA, 2010, pp. 1–9, doi: 10.1109/INFCOM.2010.5461912.

[19] 3GPP TS 23.501 version 15.3.0 Release 15, Table 5.7.4-1: Standardized 5QI to QoS characteristics mapping.

[20] 李峭, 等. 机载网络AFDX/TTE有界时延技术研究, 研究报告,

2019.10.

[21] N. Finn, Multiple Cyclic Queuing and Forwarding, IEEE 802.1 public files, 2019.

[22] M. Boyer, Combining static priority and weighted round-robin like packet scheduling in AFDX for incremental certification and mixed-criticality support, 5th European Conference for aeronautics and space sciences, 2013

[23] Chen, L., Zhang, J., Gao, T., Wang, T. (2021). Analysis of QoS Schemes and Shaping Strategies for Large Scale IP Networks Based on Network Calculus. In: Gao, H., J. Durán Barroso, R., Shanchen, P., Li, R. (eds) Broadband Communications, Networks, and Systems. BROADNETS 2020. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 355. Springer, Cham.

[24] Specht, J., and S. Samii, "Urgency-Based Scheduler for Time-Sensitive Switched Ethernet Networks," 28th Euromicro Conference on Real-Time Systems (ECRTS), pp. 75–85, 2016.