

文章编号: 1001-0920(2011)12-1791-05

NFL: 一种基于活动流参数估计的自适应模糊 AQM 算法

陈伟杰^{a,b}, 王万良^a, 郑建炜^a, 蒋一波^a

(浙江工业大学 a. 计算机科学与技术学院, b. 之江学院, 杭州 310023)

摘要: 针对主动队列管理(AQM)机制面对动态突变的网络存在参数配置难问题, 提出一种将模糊 AQM 和活动流参数估计策略相结合的自适应 AQM 算法(NFL). 在综合权衡各性能指标的基础上, 设计了一组能适应一定网络变化的模糊规则, 并对算法进行了运算优化. 为捕获网络突发流, 引入了一种基于 Bloom 滤波器的无状态维护活动流参数估计策略, 并依此提出一个模糊 AQM 输出增益补偿器. 实验结果表明, NFL 能较好地适应网络变化, 相对其他算法, 具有更快的收敛速度和稳定的稳态队列控制性能.

关键词: 拥塞控制; 主动队列管理; 模糊逻辑; Bloom 滤波器

中图分类号: TP393

文献标识码: A

NFL: Adaptive fuzzy-logic-based AQM algorithm with active-flow parameter estimation

CHEN Wei-jie^{a,b}, WANG Wan-liang^a, ZHENG Jian-wei^a, JIANG Yi-bo^a

(a. School of Computer Science and Technology, b. Zhijiang College, Zhejiang University of Technology, Hangzhou 310023, China. Correspondent: WANG Wan-liang, E-mail: ww1@zjut.edu.cn)

Abstract: For the problem that the active queue managements(AQM's) parameters configuration is difficult, especially in the dynamic network, an adaptive AQM algorithm (called NFL) is proposed, which is composed of two main parts: the fuzzy AQM and the active-flow estimation strategy. Considering the tradeoff among each performance indicators, a set of fuzzy rules are built for NFL to adapt to the dynamic network situation. Furthermore, an optimization method is raised, which reduces the computational complexity of fuzzy AQM. Then, a stateless active-flow estimation strategy based on Bloom filter is introduced to capture network congestion status. According to this, an output gain compensator for fuzzy AQM in accordance with active-flow-number parameter is proposed. Simulation results show that NFL is adaptive to dynamic network with fast convergence rate and stable steady-state queue control performance, and the comprehensive performance of NFL is more excellent than other AQM algorithms.

Key words: congestion control; active queue management; fuzzy logic; Bloom filter

1 引言

自 RED(random early detection)算法被 IETF 推荐为主动队列管理(AQM)候选算法以来, 拥塞控制研究重心由端结点转向了中心路由结点^[1]. 然而, RED 算法性能依赖于其参数配置, 因此很多学者提出了各类 AQM 算法. Floyd^[2]提出了最大丢包概率自适应调整 ARED(adaptive-RED)算法, 然而固定的调整步长导致了算法鲁棒性差; 陆锦军等人^[3]利用控制论给出了 RED 参数的稳定性分析. 为分析 TCP 和 AQM 的动态行为, Misra 等人^[4]利用随机差分方程(SDEs)建立了 TCP(transmission control protocol)和 AQM 系统

非线性流模型, 这为基于控制论的 AQM 流派的诞生奠定了理论基础. Hollot 等人^[5]将 TCP/AQM 模型在稳定点附近线性化, 提出了 PI(proportion integration)算法, 并给出了基于幅相裕度稳定性准则的参数配置策略, 但过于保守的参数配置导致 PI 面对网络的突变收敛速度慢. Melchor 等人^[6]针对不确定时延 TCP/AQM 系统, 给出了非脆弱性 PI 设计方法, 虽然性能有所提高, 但并未根本解决参数配置问题.

现有的基于线性定常系统的网络模型难以准确地描述网络动态特性, 面对动态变化的网络场景, 此类 AQM 算法的参数配置相当困难. 模糊逻辑

收稿日期: 2010-07-07; 修回日期: 2010-10-08.

基金项目: 国家自然科学基金项目(61070043, 60573123); 浙江省自然科学基金项目(Y1100611).

作者简介: 陈伟杰(1985-), 男, 博士生, 从事网络控制与主动队列管理的研究; 王万良(1957-), 男, 教授, 博士生导师, 从事网络控制与调度等研究.

系统常被用于解决无法用数学模型准确描述的对象,运用自然语言描述系统控制准则,特别适用于复杂时变的网络系统.樊燕飞等人^[7]将模糊逻辑引入到 AQM 设计中,并研究了 ECN(explicit congestion notification)策略对算法的性能影响;Liu 等人^[8]使用队列长度和缓冲利用率作为模糊控制器输入;Hadjadj 等人^[9]将 PI 作为模糊控制输出补偿器,提出了 FAFC(fast and autonomic fuzzy controller)算法;Chrysostomou^[10]的 FEM(fuzzy explicit marking)算法,使用 AIMD(additive increase multiplicative decrease)机制自适应调整模糊 AQM 的丢包概率输出增益.上述的模糊 AQM 相对 RED 和 PI 算法,性能得到了较大的提高,但面对某些动态网络场景,算法依然存在队列控制性能差、响应速度慢等不足.

针对 AQM 算法存在参数配置困难等问题,在分析各类网络拥塞情况的基础上,结合模糊逻辑技术,提出一种基于活动流参数估计的自适应模糊 AQM 算法 NFL (adaptive fuzzy-logic algorithm with active-flow-number estimation).该算法通过模糊 AQM 动态更新路由丢包概率,使路由队列长度稳定在目标值附近,并对模糊 AQM 进行了运算优化,使算法时间复杂度降低;同时引入一种无状态维护的活动流在线估计算法,并依据该参数自适应补偿模糊 AQM 输出,以此来弥补固定模糊 AQM 规则的缺陷.实验表明, NFL 算法是可行有效的.

2 模糊 AQM

TCP/AQM 网络系统是一个复杂的分布式时变非线性系统,而模糊逻辑控制则是使用先验知识来启发控制对象,适合模型不确定的网络系统. NFL 由模糊 AQM 和活动流在线估计补偿策略组成,算法架构如图 1 所示.

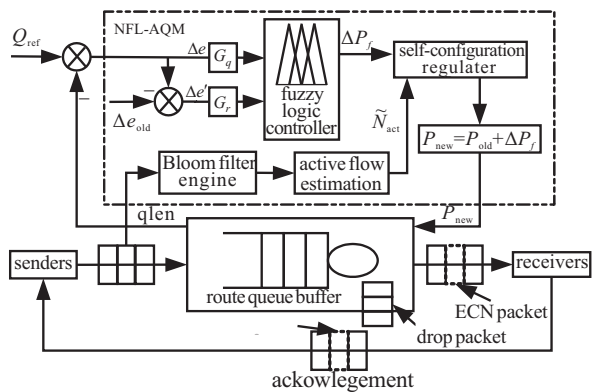


图 1 NFL 算法架构

2.1 输入输出接口设计

在设计中,模糊 AQM 采用双输入、单输出(MISO)的模糊模型.在综合分析网络拥塞各种情况的基础上,分别将瞬时队长 qlen 与目标队长 Q_{ref} 的

偏差 Δe , 以及两个采样时刻偏差的变化量 $\Delta e'$ 作为模糊 AQM 的两个输入. Δe 表征瞬时队长与目标值偏离程度, $\Delta e'$ 表征拥塞的发展趋势. Δe 和 $\Delta e'$ 通过下式计算得到:

$$\begin{cases} \Delta e = qlen - Q_{ref}, \\ \Delta e' = \Delta e - \Delta e_{old}. \end{cases} \quad (1)$$

在模糊逻辑中,隶属度函数的功能是将输入变量的定义域映射到论域.为使算法更具灵活性,首先通过缩放因子 G_q 和 G_r , 分别将 Δe 和 $\Delta e'$ 标准化到中间域 $[-1, 1]$, 然后再将标准化变量通过隶属度函数映射为语言变量.缩放因子表达式为

$$G_q = \begin{cases} \frac{1}{Q_{ref}}, & \Delta e \leq 0; \\ \frac{1}{Bufsize - Q_{ref}}, & \Delta e > 0; \end{cases} \quad (2)$$

$$G_r = \frac{1}{2Bufsize}. \quad (3)$$

这里的 Bufsize 为路由缓冲区大小.

由图 1 可知, NFL 周期性地采样早期拥塞信号, 依此更新丢包概率来控制流的汇聚速率和队列长度.文中选择丢包概率的步进增量 ΔP_f 作为 NFL 的控制输出, 更新公式如下:

$$P_{new} = P_{old} + \Delta P_f, \Delta P_f \in [-0.1, 0.1]. \quad (4)$$

为准确快速地检测并控制瓶颈链路潜在的拥塞, 通过反复试验法, 为每个输入设计了 9 个三角/梯形相邻型隶属度函数, 如图 2 所示. 相应的 Δe 和 $\Delta e'$ 论域

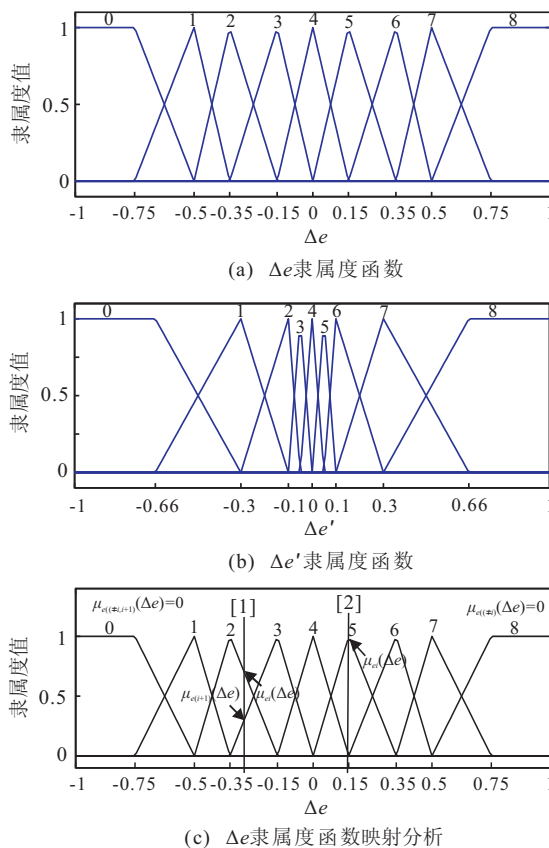


图 2 输入变量隶属度函数

集合定义为: {negative very big (0), negative big (1), negative medium (2), negative small (3), zero (4), positive small (5), positive medium (6), positive big (7), positive very big (8)}. NFL 算法选择丢包概率的步进增量 ΔP_f 作为模糊 AQM 的输出. 受 Ralf Wieland^[11] 的启发, 采用清晰值代替隶属度函数描述模糊规则的输出. 清晰值的优势在于解模糊时, 可使用加权求和运算来代替积分运算. 用 $\{(k) = \text{crisp value}\}$ 表示清晰值输出集合, k 表征模糊规则输出所对应的索引值, crisp value 表征 k 索引值下的清晰值. 通过反复试验, 最终获得模糊输出集合: $\{(0) = -0.002, (1) = -0.0005, (2) = -0.0008, (3) = -0.0005, (4) = 0, (5) = +0.0005, (6) = +0.0008, (7) = +0.0005, (8) = +0.002\}$. 负值代表丢包概率减小的步长, 反之亦然.

2.2 模糊规则设计与算法优化

模糊 AQM 通过模糊推理引擎 (FIE) 实现 ΔP_f 的动态更新, FIE 则是基于 If-Then 规则库完成模糊推理. 在队列管理规则设计过程中, 综合考虑算法的响应速度和稳定性, 依据 NFL 算法的输入输出定义, 设计了 9×9 条规则 (见表 1).

表 1 NFL 模糊规则

$\Delta e'$	Δe								
	0	1	2	3	4	5	6	7	8
0	0	0	1	2	3	3	4	5	5
1	0	1	1	3	4	4	4	5	5
2	0	2	2	4	4	4	5	5	5
3	0	2	3	4	4	4	5	5	6
4	0	2	3	4	4	4	5	6	7
5	1	2	3	4	4	4	5	6	7
6	1	3	3	4	4	4	5	7	7
7	2	3	3	4	4	5	6	7	8
8	3	3	4	5	5	5	7	8	8

为满足高吞吐量路由处理速度的需求, 本文对模糊推理进行了运算优化. 模糊 AQM 周期性地采样潜在的拥塞信号 Δe 和 $\Delta e'$, 然后通过隶属度函数映射为相应的语言值 $\mu_{ei}(\Delta e)$ 和 $\mu_{e'j}(\Delta e')$, $i, j \in [1, 3, \dots, 9]$. 由于采用相邻型隶属度函数, 对于任意输入变量 Δe 模糊化为语言值时, 有如下两种情况 (分别对应图 2(c) 左右半边):

- 1) 通常情况, $\exists \mu_{ei}(\Delta e) \neq 0, \mu_{e(i+1)}(\Delta e) \neq 0$, 并且其他 7 个语言值 $\mu_{e(\neq i, i+1)}(\Delta e) = 0, i \in [1, 2, \dots, 8]$;
- 2) 特殊情况, $\mu_{ei}(\Delta e)$ 刚好位于两隶属度函数交叉处, 此时 $\exists \mu_{ei}(\Delta e) \neq 0, \mu_{e(\neq i)}(\Delta e) = 0, i \in [2, \dots, 9]$.

$\Delta e'$ 情况类似. 这意味着在大部分情况下, 模糊 AQM 仅需对这 4 个有意义的语言值 $\mu_{ei}(\Delta e), \mu_{e(i+1)}(\Delta e), \mu_{e'j}(\Delta e'), \mu_{e'(j+1)}(\Delta e')$ 进行后续的模糊运算. 虽然设计了 9×9 条规则, 但实际上模糊推理仅需 4 条相应的模糊规则. 模糊推理采用 “min-max” 算子, 具体算法描述如下:

Step 1: 将 Δe 和 $\Delta e'$ 模糊化, 得到 4 个非 0 的语言值: $\mu_{ei}(\Delta e), \mu_{e(i+1)}(\Delta e), \mu_{e'j}(\Delta e'), \mu_{e'(j+1)}(\Delta e')$.

Step 2: 将上述 4 个语言值赋予相应的 4 条模糊规则, 得到规则结果集: $r_k^m = \min\{\mu_{e(i, i+1)}(\Delta e), \mu_{e'(j, j+1)}(\Delta e')\}$. r_k^m 代表第 m 条规则的输出为清晰值 k 的模糊规则结果, k 为清晰值输出集合索引值.

Step 3: 从规则结果集中, 为每个不同的清晰值筛选最优规则结果 $r_k = \max\{r_k^1, r_k^2, \dots\}$.

Step 4: $o = \frac{\sum_{k=1}^n r_k o_k}{\sum_{k=1}^n r_k}$. o_k 代表清晰值; o 代表最终的模糊输出 ΔP_f ; n 为 Step 3 最优规则结果集 $\{r_k\}$ 的大小, 其中 $n \leq 4$.

由上可知, 每次模糊推理, 最多需在 Step 2 进行 4 次 min 操作; 在 Step 3 进行 4 次 max 操作; 在 Step 4 进行 4 次乘运算、1 次除运算以及少量的加和赋值运算. 因此, 通过对模糊推理进行运算优化, 可使算法时间复杂度从 $O(n)$ 降为 $O(1)$, n 为模糊 AQM 规则数.

3 NFL 算法

研究表明, 时变的网络参数 (N, RTT, C) 是导致 AQM 设计困难的主要原因. 在网络参数中, C 链路带宽由链路本身决定, 在有线环境下通常是固定的. 由于数据流服从重尾分布, 意味着参数 N 具有突发性. N 的变化导致路由队长变化, 进而影响数据包的排队时延, 最终将导致往返时间 RTT 的变化, 而 RTT 的变化则进一步影响到 AQM 反馈信号被执行的时间. 由上分析可知, N 是三元组中最为关键的一个参数, 如能及时估计 N , 便能把握拥塞趋势. 本节首先引入了活动流参数估计策略, 然后依据稳态丢包方程, 提出一种模糊 AQM 自适应调整方法.

3.1 基于 BF 的活动流参数估计

对 N 的估计将面临如下两个问题: 1) 路由资源有限, 必须设计一种无状态维护的估计算法; 2) 数据流具有突发性, 如何快速地判断数据流的建立和消亡, 也是估计算法需解决的问题. 作为一种空间效率很高的随机数据结构, BF (Bloom filter) 以在极少情况下存在假命中的代价来换取高效存储空间利用率^[12]. 对于能容忍少量错误率的 N 估计场合, 可使用 BF 来极大地节省路由存储空间. 活动流参数估计算法 (AFE) 的核心思想如图 3 所示. 当一个数据

包 (Pkt) 到达路由, BF 引擎提取数据包头部的流标识字符串 F_{id} , 然后在 BF 集合中查找是否存在 F_{id} , 如果不存在则执行插入操作, 同时将 N_{act} (当前活动流参数测量值) 加 1; 否则不执行任何操作, 等待下一个数据包到达. 其中 $h_i(F_{id})$ 为哈希函数序列, AFE 算法共选择了 6 个哈希函数序列 {SAX, SDBM, JS, ELF, DJB, AP}, 即 $k = 6$. 虽然 MD5 和 SHA1 比上述 6 个函数分散性更均匀, 但计算复杂, 并不适合高速处理的路由.

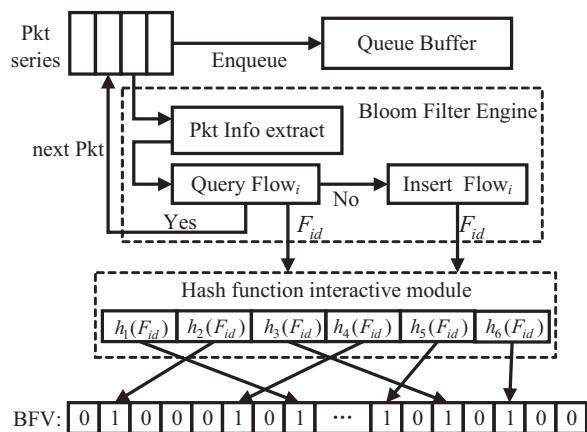


图 3 AFE 算法原理

因 BF 是采用压缩映射方式存储元素, 且 AFE 是无状态维护的算法, 所以 AFE 虽能快速地捕获到新加入的流, 却无法察觉已消亡的流. 为解决此问题, 为 AFE 增加了一个周期性刷新策略, 即每经过 ΔT 周期, AFE 获得当前的 N_{act} 统计; 然后使用指数加权平均 (EWMA) 方法估计当前活动流参数 \tilde{N}_{cur} ; 最后清除 BFV 数据, 并进入下个周期的 N_{act} 统计.

EWMA 方法如下:

$$\tilde{N}_{cur} = (1 - \omega)\tilde{N}_{last} + \omega \times N_{cur}. \quad (5)$$

其中: \tilde{N}_{cur} 为上周期估计值, ω 为指数加权值. 经过多次测试, 当 $\omega \in [0.7 \ 0.85]$ 时, 估计值与实际值误差较小. 依据最小“假命中率”原则^[12], AFE 算法在网络数据包平均大小 $MSS = 500 \text{ Byte}$, $m = 15$, $C = 15 \text{ Mbps}$ 条件下, 可得到最优更新时间 $\Delta T_{op} = 8.075 \text{ s}$, 此时假命中率 $f(k_{min}) = 0.015 \ 625$.

3.2 自适应模糊 AQM 补偿策略

目前, 大部分自适应模糊控制器采用动态调整模糊规则和隶属度函数的方式, 这一般需要较大的存储空间. 另一种则是对模糊输出进行补偿调整. 在文献 [10] 中, FAFC 使用 PI 补偿模糊 AQM 的输出, 但 PI 本身的参数配置却依赖于网络参数, 故 FAFC 性能反而受到了 PI 的牵制, 这可从实验部分得证. Ranjan 分析了 TCP/AQM 系统非线性特性, 并给出了如下稳态丢包率与三元组的近似等式关系:

$$T = \frac{C}{N} \approx \frac{MSS \times K}{\sqrt{p} \times RTT} = \frac{MSS \times K}{\sqrt{p} \times (\tau + qlen/C)}, \quad (6)$$

$$p \approx \left(\frac{MSS \times K \times N}{\tau \times C + Q_{ref} \times MSS \times 8} \right)^2. \quad (7)$$

如果 AQM 能将瞬时队长稳定在目标值 Q_{ref} 附近, 则 RTT 的传输时延将稳定在 $\tau + Q_{ref}/C$ 附近, 稳定的 RTT 保障了数据的稳定传输. 对式 (6) 进一步推导可得稳态丢包率近似方程 (7).

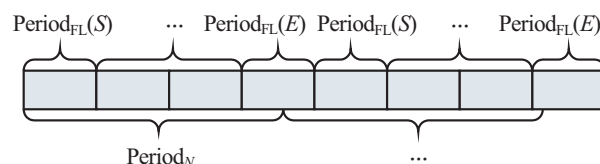


图 4 模糊 AQM 自适应调整策略

由式 (7) 可知, N 是影响稳态丢包率的主要因素, 可依据活动流参数对模糊 AQM 自适应调整. 图 4 中的 $Period_N$ 代表活动流参数估计周期, $Period_{FL}$ 代表模糊 AQM 丢包概率更新周期. $Period_{FL}(S)$ 代表模糊 AQM 在 $Period_N$ 周期内更新丢包概率的起始周期, $Period_{FL}(E)$ 为 $Period_N$ 的结束周期. 分别对 $Period_{FL}(S)$ 和非 $Period_{FL}(S)$ 的模糊 AQM 采用不同的补偿策略. $Period_{FL}(S)$ 采用如下策略进行粗调:

$$P_{new} = (P_{old} + \Delta P_f) \times \frac{p_{cur}}{p_{last}}; \quad (8)$$

对于非 $Period_{FL}(S)$ 采用下式进行微调:

$$G_p = \begin{cases} \frac{p_{last} - p_{cur}}{p_{last}}, & N_{last} > N_{cur}, \text{ qlen} < 0.8Q_{ref}; \\ \frac{p_{cur}}{p_{cur} - p_{last}}, & N_{last} < N_{cur}, \text{ qlen} > 1.2Q_{ref}; \\ 1, & \text{otherwise}; \end{cases} \quad (9)$$

$$P_{new} = P_{old} + G_p \times \Delta P_f. \quad (10)$$

其中: G_p 为微调增益因子, p_{cur} 为当前周期根据式 (7) 计算得到的丢包率, p_{last} 为上个周期丢包率计算值.

4 实验仿真

为进一步验证 NFL 算法的有效性和可行性, 在 NS2^[13] 仿真平台上进行了一系列的仿真实验, 同时与 ARED^[2], PI^[5], FAFC^[9], FEM^[10] 4 种 AQM 算法作了对比, 算法参数配置见表 2.

表 2 AQM 算法参数配置

参数配置	
ARED	$\min_{th} = \frac{1}{5} \max_{th} = 70$, interval = 0.1 s
PI	$\alpha = 1.822 \times 10^{-5}$, $\beta = 1.816 \times 10^{-5}$
FAFC	$f_a = 3$, $f_b = 1$, $Period_{FL} = 0.01 \text{ s}$
FEM	$Period_{FL} = 0.01 \text{ s}$
NFL	$Period_{FL} = 0.01 \text{ s}$

在仿真中, 研究了面对突发流场景时算法的鲁棒性, 并使用队列平均长度 (Avgq)、队列震荡方差

(STD), 链路利用率(Utiliz)和丢包率(LossR)作为算法性能衡量指标. 在实际网络中, 虽然数据传输需要经过多条链路, 但瓶颈链路一般只为其中的某一段, 因此可将拥塞的网络缩影为单瓶颈网络来研究 AQM 的性能. 仿真拓扑配置如图 5 所示, 瓶颈链路结点 n_1 和 n_2 采用 AQM 算法, 路由缓冲大小为 300 pkt, Q_{ref} 设置为 150 pkt, TCP 为允许 ECN 模式, 分组大小 MSS 为 500 B. 源端 S_i 与终端 D_i 存在数据连接流 $i = 1, 2, \dots, n$. 为表征网络负载动态变化, TCP 连接数 N 采用如下分段形式 $\{300, 600, 200, 700, 300, 500, 200\}$, 每个 N 维持 50 s, 仿真持续 350 s.

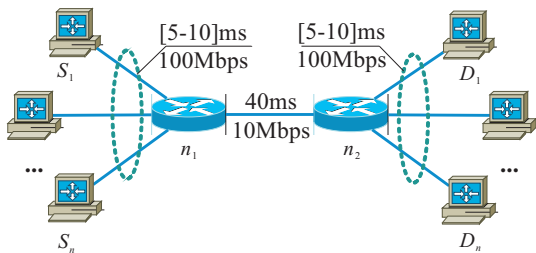


图 5 仿真网络拓扑图

图 6 描述了 5 种 AQM 算法随着流的加入或撤离网络, 瞬时队列演化情况. 结果表明, PI 的队列控制性能最差, 根本无法在 50 s 内将瞬时队长收敛到目标值

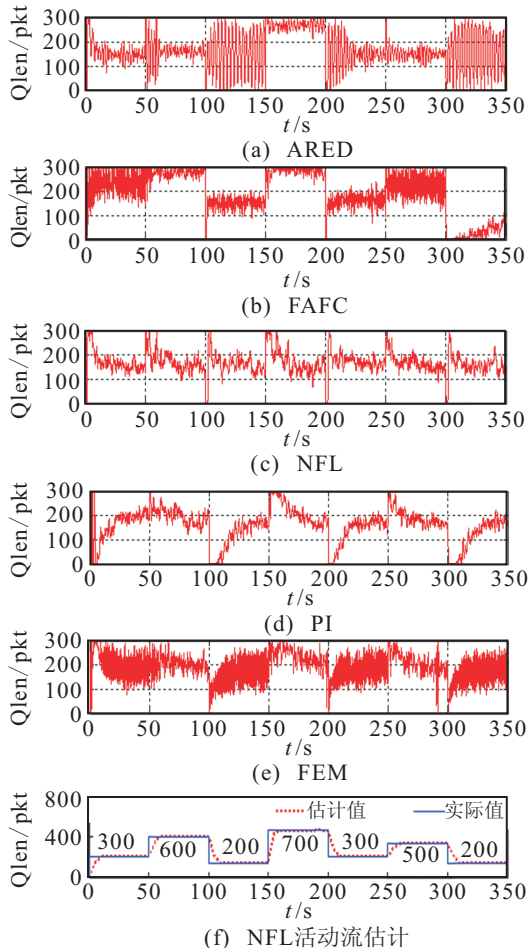


图 6 动态流量场景下队列的演化

(150 pkt); 其次是 FEM, 收敛速度较慢, 同时伴随着队列的严重震荡; FAFC 虽在某些负载场景下稳定在目标值附近 (比如 100-150 s 时), 但由于其采用 PI 作为模糊 AQM 输出补偿器, 明显影响了算法性能; ARED 在大部分负载下, 都能将队列保持在目标值附近, 但在重负载下出现队列无法收敛的现象; NFL 通过活动流在线估计策略, 能实时检测到网络动态, 有利于其更好地控制队列. 图 6(f) 给出了 NFL 活动流的实际值和估计值曲线, 可知 NFL 的活动流参数估计值较准确.

表 3 给出了 5 种算法的综合指标对比, 其中 NFL 的平均队长、队列震荡方差以及丢包率指标最好; 虽然 ARED 队列控制能力较强, 但其丢包率比 NFL 高两个数量级; PI 的队列震荡程度最大, 其次为 FAFC, 这表明基于模型的 PI 类算法参数配置依赖于网络参数, 网络场景变化后, 算法性能较差. FEM 不能将队列收敛于目标值, 其他指标与 NFL 相当. 综上所述, NFL 面对动态突发网络场景时, 算法鲁棒性最强.

表 3 ARED, PI, FAFC, FEM, NFL 性能对比

	Avgq/pkt	STD/pkt	Utiliz/%	LossR/%
ARED	167.1305	66.7443	99.719	14.789
PI	165.5259	118.3734	93.628	0.762
FAFC	202.6311	86.6338	99.762	1.137
FEM	201.5307	53.9719	98.838	0.638
NFL	157.2149	41.4390	99.504	0.609

5 结 论

目前, 基于线性定常系统的 TCP/AQM 数学模型使用固定的网络参数 (N, RTT, C) 来描述网络特性, 而实际网络则是一个分布式时变的非线性系统, 因此导致了基于模型的 AQM 算法鲁棒性差. 为此, 将模糊逻辑引入到 AQM 设计中, 通过分析、总结、提炼大量的网络拥塞控制规则, 提出了一种能适应一定网络场景的模糊 AQM 算法, 作为 NFL 的主要队列管理策略. 针对相邻型隶属度函数的特征, 对模糊 AQM 作了运算优化, 将算法的时间复杂度从 $O(n)$ 降为 $O(1)$, 同时也降低了推理过程中临时变量对存储空间占用的需求. 本优化策略可推广至其他采用邻近型隶属度函数的模糊控制系统. 文中分析了网络参数的性质以及参数间的相互关系, 指出 N 的突发性是拥塞发生的本质原因. 为此, 引入了一种基于 BF 的无状态维护的活动流在线估计策略, 并依此估计值自适应补偿模糊 AQM 输出, 作为 NFL 的辅助队列管理策略. 仿真实验表明, 与其他算法相比, NFL 算法提高了队列控制性能和链路利用率, 降低了丢包率. 下一步的工作是研究 NFL 算法在差分服务网中的性能.