A Chemical-Inspired Approach to Design Distributed Rate Controllers for Packet Networks

Massimo Monti*, Manolis Sifalakis*, Thomas Meyer*, Christian F. Tschudin*, Marco Luise†
*Department of Mathematics and Computer Science, University of Basel, Switzerland
E-mail: {massimo.monti, th.meyer, sifalakis.manos, christian.tschudin}@unibas.ch
†Department of Information Engineering, University of Pisa, Italy
E-mail: m.luise@iet.unipi.it

***Abstract*—In computer networks, a Distributed Rate Controller (DRC) must quickly propagate changes of inflow rates and let participating sites converge to their admissible rate. In this paper we introduce a family of DRCs where controllers can be easily customized, and their performance and dynamics are strictly predictable. Borrowing from engineering methods in Chemistry, we show how to derive the deterministic mathematical model of the network flow. We also report on simulation and native experimental results that validate our theoretical approach.**

I. Introduction

Until today the Internet is a resource-interconnecting and access-sharing facility. Access sharing is enabled through the statistical multiplexing of flows. In this process, rate control is the main intrinsic function both for regulating the relative network utilization by the admitted flows, and for guaranteeing admissibility for all flows that demand access. Capacity allocations and resource reservation, congestion control and avoidance, DoS mitigation, service differentiation, and traffic shaping are examples of embedded rate control functions serving as a common means to different ends.

Given the prominent role of this function in network operations, we have developed a systematic way of designing, analyzing, and deploying rate controlling functions. This methodology, which borrows a paradigm and a structural representation from chemical engineering, allows the direct derivation of the algorithm's flow model and thus an accessible, but accurate, analysis of the algorithm's dynamics. (In the related technical report [1], the flow-model-based analysis has proven to be accurate enough to alleviate the need for microscopic queue-theoretical modeling, which is cumbersome and largely bases on traffic-pattern assumptions.)

The family of rate controllers designed in this way can be easily parametrized, extended, and customized, for various purposes and in different operational environments: (a) at the client side to enable service differentiation among user's flows (e.g., management of DSL users' traffic), (b) Among clients for a distributed coordination between aggregate flows of nodes, (c) in the network to manage the allocation of capacities to admitted flows (e.g., distributed rate control for cloud environment), and (d) at the server to regulate the access to a resource (e.g., differentiation of user- from control- traffic in data center management).

A. Related work

In [2], Raghavan *et al.* proposed a distributed rate-limiter for regulating access to cloud resources. In their Flow Proportional Sharing (FPS) approach, distributed instances of token-bucket rate-allocators exchange statistics and "shift" capacity slices to different congestion-aware flows. The chemical rate controller has a similar characteristic behavior to FPS but it (a) is not confined to cloud settings and applications only, (b) is agnostic to intra-protocol rate estimators, and (c) is more predictable and parameterizable for flow cooperation or flow competition.

To control the sharing and virtualization of resources within the cloud, Abu-Libdeh *et al.* in [3] proposed a rate control scheme (Ajil) for traffic differentiation across multicast channels based on an out-of-band feedback channel and on rate-controlling reactors. Despite the conceptual similarity of the building blocks, Ajil applies coarse TCP-like rate-control policies, depends on synchronized clocks between resource users, and flow rate estimates are discretely computed at periodic intervals. Our chemical rate controller does not suffer from these artifacts and operates smoother in continuous time.

Aiming at a fair use of the network, several proposals [4]–[9] advocate the use of network assisted signaling towards the end user, instead of taking rate control actions in the network (non-work conserving scheduling). We are aligned with such a thinking and let actions for regulating the capacity sharing be made at the user-systems. However, we believe that our system is more suitable for this purpose than either of the above measures as in our system, the share-holders collectively regulate their relative capacity utilization rather than expecting from the network to act intrusively [7] or to compute and distribute individual capacity shares [6], [8].

Other nature-inspired approaches for flow-control have been proposed in the literature: e.g., the Lotka-Volterra competition model has been used in [10] to update the congestion window of TCP and in [11] as a hop-by-hop rate control mechanism for WSNs. By contrast to them, we do not borrow a specific model and shape the traffic flow by numerically approximating the equations (ODEs) that describe such a model. Rather, we adopt a metaphor (i.e., chemical reaction) and the whole underlying theory (i.e., dynamics and kinetics of reaction networks), which allow us to design, implement, analyze, and execute computational systems.

978-3-901882-50-0

Many works in the literature propose to use flow models and a control-theoretic approach to design networking algorithms, e.g. [12]–[16]. A central challenge in almost all of them is the derivation of a flow-model that approximates sufficiently well the proposed algorithm's dynamics. By contrast, a fundamental contribution of our approach is that no such derivation task is required! Instead, the execution model (state machine, automaton) *is* the state-space expression of the flow-model. Practically, this means that one can start with a desired dynamic behavior (flow model) and end up with the algorithm (that computes this model), rather than first devise the algorithm, find then the flow-model that approximates it, and finally examine if the resulting behavior is the desired one.

B. Paper outline

In the remainder of this paper we start slowly setting the stage by revising the basic operation of a token-bucket rate-control scheme in Sect. II, and establishing the analogy to chemical reactions in Sect. III. We then start extending this basic analogy to introduce dynamism and feedback control, arriving at our core design of an adaptive chemical rate controller (Sect. IV). We evaluate our design both in simulation and with a real deployment in Sect. V. Finally we conclude this paper with a discussion of the insights developed through this work in Sect. VI, and our future orientation in Sect. VII.

II. Token-Bucket Based Rate Control

A *token bucket* or *leaky bucket* scheme (see Fig. 1) has probably been the most common building block in the design of more or less sophisticated (and protocol-independent) rate control for various purposes both in reservation-based and reservation-less networks.

In the simplest case (rate limiter), a token bucket is used to regulate the departure process of packets from a queue; tokens essentially authorize the consumption of a respective number of packets or bytes. The bucket is filled with tokens at a constant rate, with excess tokens being tail-dropped. In this arrangement, the output of packets from the queue can never exceed a burst size equal to the size of the bucket, nor sustain a constant rate higher than the refill rate of tokens.

In a more dynamic arrangement (rate controller), the rate limit imposed by the token bucket may vary in time. This can be achieved by adapting the refill rate of tokens in response to feedback from the network, or to the departure process differentiation of two or more regulated queues. To analyze the behavior of such a system, one may regard the token bucket as yet another queue. In such an arrangement, the departure process would depend on the state of both queues: The process blocks if either the packet or the token queue is empty.

III. Rate Control using Chemical Reactions

Meyer introduced in [17] the chemical reaction as a metaphor for modeling information processing of packet streams and thereby designing network protocols. Leaving aside a complete exploration of this metaphor, in this section we concentrate on the aspects that enable us to describe through this model the operation of a simple fixed rate token-bucket controller. This will serve as an entry point to the design of our adaptive rate controller family in the next section.

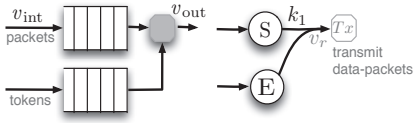


Fig. 1: Token-Bucket scheme and simple equivalent chemical reaction.

Consider the simple reaction $\mathrm{S} + \mathrm{E} \xrightarrow{k_1} \emptyset$ schematically illustrated in Fig. 1. An S-molecule reacts with an E-molecule producing a transmission. By associating packets with virtual molecules, we can refer to the fill-level of a queue as a molecular concentration. E.g., the number of queued packets and available tokens can be associated to the concentrations of species S and E respectively. Similarly, by associating a chemical reaction with the transmission of a packet, we can refer to the sending rate v_{out} as the reaction rate v_r (thereafter, we use symbol "v" to represent rates). If one of the two reactants is depleted, the reaction becomes *inert*. This condition captures the dependency between token and packet availability for scheduling transmissions out of a token-bucket regulated queue.

The reaction between the reactant species takes place at an average rate v_r. If we choose the inter-reaction times to be drawn from an exponential distribution, then this representation models the interaction of a $\cdot/M/1$ queue with the token bucket (subject to token availability).

Practically in this model, the reaction merely defines a rule (dependency) that relates the departure process of two (or more) queues. Also, the rate of the reaction describes the probability distribution of this rule being successfully applied.

A. Law of Mass Action Scheduling

At this point, to be consistent with the chemical model in [17], we need to introduce the laws that underpin the dynamic operation of a reaction network. The algorithm has to decide how the reactions in the network are scheduled. A work-conserving round-robin approach seems at first plausible for deterministic fairness or a random selection approach for statistical fairness. However, ideally for traffic shaping purposes, the regulation process should be a response towards balancing the pressures applied to the system, which can only be achieved by afflicting packets of certain queues with a small additional delay. (This will be understood more intuitively in the next section, where our design needs to make rate-controlling decisions that balance multiple flows.) We therefore propose a non-work-conserving, fill-level weighted scheduling algorithm. The algorithm that implicitly assigns weights to reactions bases on the chemical *Law of Mass Action* (LoMA):

L1: *The reaction rate is proportional to the concentration (quantity) c of all reactant molecules.*

LoMA essentially relates the reaction rates with the reactant

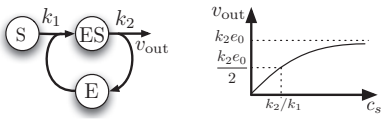


Fig. 2: A chemical fixed-rate token-bucket.

concentrations through the equation

$$v_r = k_r \times \prod_{i=1}^{|\mathscr{S}|} c_i^{\alpha_i}$$

where c_i is the concentration of the i-th species, k_r is a "speed" coefficient for reaction r, $|\mathscr{S}|$ is the set of all molecule species, and α_i is the number of molecules from species i that are consumed by reaction r. For example for reaction $\mathrm{S}+\mathrm{E} \xrightarrow{k_1} \mathrm{ES}$ in Fig. 1 its rate based on the LoMA is given by $v_{r_1} = k_1 \cdot c_{\mathrm{S}}^1 \cdot c_{\mathrm{E}}^1$

A direct consequence of this fill-level-proportional scheduling algorithm is the *flow conservation* principle:

L2: *At equilibrium state, for each molecular concentration, the total production rate equals the total consumption rate.*

Another consequence is that of mass conservation in a chemical reaction along closed loops (*moieties* in chemistry):

L3: *The total sum of molecule concentrations along a loop remains constant, if (a) the total number of molecules consumed by reactions along the loop is equal to the total number of molecules produced, and (b) all concentrations along the loop are altered only by reactions involved in this or another loop.*

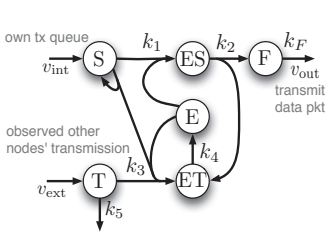
B. Enzymatic rate control

We now have to determine the rate at which the token bucket is filled in the chemical model. If we assume that the concentration of E-molecules is the one that represents the number of tokens in the bucket, we care that this concentration is refreshed at a fixed rate up to a maximum size. A simple possibility is to externally inject new E-molecules at a certain rate, and constantly monitor the concentration size such that it does not exceed the set limit (which equals the bucket length). However, an elegant design pattern (or *motif* according to [17]) comes directly from chemistry to capture this situation with a control loop instead. This enzymatic reaction pattern, illustrated in Fig. 2, introduces a second species ES and reaction $\mathrm{ES} \xrightarrow{k_2} \mathrm{E}$.

The combination of LoMA and the two conservation laws above is sufficient to provide us with a direct description of the system in Fig. 2 through the following equation, which relates the output rate v_{out} to the control parameters and to the fill level of the input queue (c_{S} concentration):

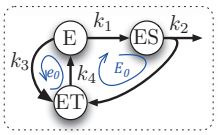
$$v_{\text{out}} = k_2 e_0 \frac{c_{\mathrm{S}}}{c_{\mathrm{S}} + k_1/k_2} \tag{1}$$

where $c_{\mathrm{E}} + c_{\mathrm{ES}} = const. = e_0$ according to the mass conservation law (**L3**). The product $k_2 e_0$ defines the upper rate limit $v_{\max}$ that can be achieved, whereas the ratio of control constants (k_2/k_1) specifies how aggressive the rate regulation is.

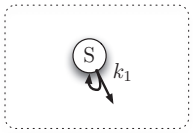


$$\begin{aligned}
r_{\text{int}}: &\quad \emptyset \xrightarrow{v_{\text{int}}} \mathrm{S} \\
r_{\text{ext}}: &\quad \emptyset \xrightarrow{v_{\text{ext}}} \mathrm{T} \\
r_1: &\quad \mathrm{S}+\mathrm{E} \xrightarrow{k_1} \mathrm{ES} \\
r_2: &\quad \mathrm{ES} \xrightarrow{k_2} \mathrm{ET}+\mathrm{F} \\
r_3: &\quad \mathrm{T}+\mathrm{E}+\mathrm{S} \xrightarrow{k_3} \mathrm{ET}+\mathrm{S} \\
r_4: &\quad \mathrm{ET} \xrightarrow{k_4} \mathrm{E} \\
r_5: &\quad \mathrm{T} \xrightarrow{k_5} \emptyset \\
r_{\text{out}}: &\quad \mathrm{F} \xrightarrow{k_F} \emptyset
\end{aligned}$$

Fig. 3: Adaptive chemical rate controller. The symbolism $\emptyset \xrightarrow{v} \mathrm{X}$ signifies the generation of X-molecules at rate v, whereas $\mathrm{X} \xrightarrow{k} \emptyset$ signifies the depletion of X-molecules with rate coefficient k.



(a) Adaptive-rate



(b) Concentration-to-rate

Fig. 4: Chemical motifs for the distributed, adaptive rate controller.

IV. Adaptive Chemical Rate Controller Design

We are now ready to describe the reaction network that provides the core design of a distributed adaptive rate controller. By parametrizing this design or extending its basic set of reactions, a family of rate controllers can be derived.

Figure 3 presents the complete reaction network implementing the functionality of the adaptive rate controller. The structure represents an extension of the fixed-rate chemical token bucket of Fig. 2, through simple chemical design patterns.

Exploring part-by-part this reaction network, we first identify two inter-weaved reaction loops, one of which is encapsulated in (part of) the other. This pattern is highlighted in Fig. 4(a). We will denote the sum of concentrations along the larger loop as $E_0 = c_{\mathrm{ES}} + c_{\mathrm{E}} + c_{\mathrm{ET}} = \text{const.}$ according to **L3**, and the sum of concentrations along the smaller loop as $e_0 = c_{\mathrm{E}} + c_{\mathrm{ET}}$. Note that according to **L3**, e_0 is not constant because E is consumed by two reactions (r_1 and r_3) but only produced by one (r_4); also ET is consumed by only one reaction (r_4) but produced by two (r_2 and r_3). The difference of the two concentrations can be expressed as a function of the ratio v_3/v_1 of reaction rates consuming E-molecules, where $v_3 = k_3 c_{\mathrm{E}}$ and $v_1 = k_1 c_{\mathrm{E}}$. The difference of quantities along the two loops, $E_0 - e_0 = c_{\mathrm{ES}}$, determines the number of tokens available for dispatching packets from the input queue S.

Thus, by replacing the single conservation loop of Fig. 2 with the extended two-loop pattern of Fig. 4(a) (and replacing e_0 with $E_0 - e_0$ in (1)), the output rate v_{out} becomes dependent on the ratio of outflow rates from E-species. By adjusting this ratio we can adapt the maximum rate limit $v'_{\max} \le v_{\max}$, where $v_{\max} = k_2 E_0$ and $v'_{\max} = k_2(E_0 - e_0)$.

In order to regulate rate v_3 that controls the "speed" of the small reaction loop, reaction r_3 depends on two additional inputs. One of them is the backlog of S-molecules, which builds up when reaction r_1 cannot process them fast enough.

This represents the amount of locally queued packets. The other is the concentration c_T, which increases at a rate that reflects the "sensed" aggregate utilization of the network of other transmitters. (How the sensing function is implemented is beyond the scope of this design. Depending on the application environment different solutions may be adopted from the literature (e.g., carrier sensing, multicast, gossip, explicit out-of-band signaling, and others).)

More explicitly, concentration c_S participates in reaction r_3 with the so-called *concentration-to-rate* pattern [17] (Fig. 4(b)). The looping arrow signifies that for every molecule consumed from species S, a new S-molecule is produced. Thus c_S remains constant, and because of LoMA, the reaction rate is directly proportional to c_S.

The effect of this pattern is that when the arrival rate v_{int} at S is in flow-balance with rate v_1 of reaction r_1, the input queue is drained fast enough and there is no increasing backlog of S-molecules to "fuel" reaction r_3. The small loop is blocked and the system behaves like a fixed-rate token-bucket.

As soon as the enqueuing rate v_{int} increases too much and there are other flows claiming part of the network capacity, there will be an increasing backlog of S-molecules created that activate reaction r_3, with a rate proportional to the backlog. As there are contributor species to reaction r_3, the rate increase will be further regulated according to the utilization of the network by other users through the concentration c_T. The result will be an adaptive drop in the maximum allowed rate v'_{max}.

The last component of the design is the F-species, which implements (in a simple queue) a linear low-pass output filter. When the system is in equilibrium, the instantaneous variation of c_F will follow the variations of the influx rate v_2, and because of the LoMA the transmission rate will be: $v_{\text{out}} = k_F c_F$. This means that k_F determines the cut-off frequency for smoothing (dampening) the oscillations of the transmission rate acceleration. This is important for several reasons: (a) it reduces the burstiness of transmissions and their effects on the sensing function of other rate controllers, (b) in case of long delay networks, it "smoothens" the traffic variations over time such that rate controllers respond to them as they develop and not after they have occurred.

A. Flow-model Extraction

One of the key advantages of using the chemical metaphor to design algorithms is the direct connection of the execution process (actual algorithm) to an exact analytic flow-model that represents its dynamics. For example a computer actually instantiates and executes an exact image of the reaction network of Fig. 3, by continually computing rates and concentrations. At the same time, the direct application of the three laws (L1, L2, L3) on the model in Fig. 3 gives us the set of ODEs that connect the reaction rates and the changes of molecular concentrations over time:

$$\dot{c}_S = v_{\text{int}} - k_1 c_S c_E \qquad \dot{c}_{ES} = k_1 c_S c_E - k_2 c_{ES}$$
$$\dot{c}_E = k_4 c_{ET} - k_3 c_S c_E c_T - k_1 c_S c_E \qquad \dot{c}_T = v_{\text{ext}} - k_3 c_S c_E c_T$$
$$\dot{c}_{ET} = k_3 c_S c_E c_T + k_2 c_{ES} - k_4 c_{ET} \qquad \dot{c}_F = k_2 c_{ES} - k_F c_F$$

$c_S(0\text{s}) = 0$	$c_E(0\text{s}) = E_0$	$k_1 = 1\,{}^1/_{\text{s}}$	$k_4 = 1\,{}^1/_{\text{s}}$
$c_T(0\text{s}) = 0$	$c_{ET}(0\text{s}) = 0$	$k_2 = 100\,{}^1/_{\text{s}}$	$E_0 = 1000$
$c_{ES}(0\text{s}) = 0$	$c_F(0\text{s}) = 0$	$k_3 = 1\,{}^1/_{\text{s}}$	$k_F = 1\,{}^1/_{\text{s}}$

TABLE I: Configuration parameters for simulation tests

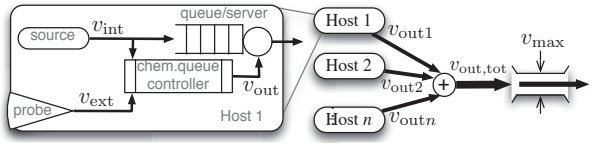


Fig. 5: System deployment in an experimental scenario of N hosts sharing a bandwidth-delay product v_{max}.

This close one-to-one relationship allows us to start from either the behavior (model) or the algorithm, and then arrive to the other. Then, thanks to the direct mapping between model-parameters and system variables, it is straight-forward to employ standard control-theory or chemical engineering tools to study and analyze the algorithm's behavior, select initial conditions, identify key parameters, etc. Such an analysis (previously generalized in [18]) for the DRC can be found in [1] and has been left out from this paper for space limitations.

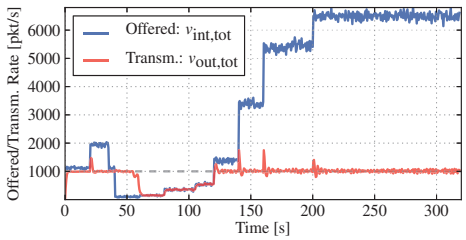
V. Evaluation of the Chemical Rate Controller

In this section we present an evaluation of the chemical rate controller both in simulations as well as in a real deployment. For the simulation-based tests we have based the system operation on packet rates and used the system settings shown in Tab. I (please refer to [1] for details on how to derive these settings analytically). The total shared rate limit v_{out} was set through the E_0 mass at $1000\,{}^{\text{pkts}}/_{\text{s}}$. For the real deployment tests we have implemented a custom queueing discipline for the Linux kernel, which operates on byte rates to account for varying size packets. Instead of replacing the existing queues in Linux' traffic control system, our software implements the chemical reaction network as an external manager module that controls the service process of the egress link queue or a certain class of traffic (see Fig. 5); concentrations in the chemical controller are mere byte counters. The intent of this test was to confirm our simulation results with more realistic capacities/speeds, and also to get a first impression of how the adaptive behavior of the chemical controller interacts with TCP's control loop.

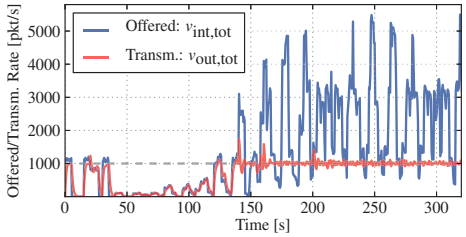
A. Multiple transmitters, multiple delay feedback

We have put into our simulation environment ,11 hosts that started generating data packets with different rates at different times and according to a Poisson model. The nodes shared a maximum capacity of $v_{\text{max}} = 1000\,{}^{\text{pkts}}/_{\text{s}}$. Additionally, each host sensed the utilization of the shared capacity $v_{\text{ext},j}$ with a variable delay from a uniform distribution of $(0, 60)$ ms.

Figure 6 shows that the *capacity allocation is efficient*: It depicts the cumulative input and resulting output of the system, i.e., the total offered load $v_{\text{int,tot}} = \sum_i v_{\text{int},i}$ and the total utilization of the capacity $v_{\text{out,tot}} = \sum_i v_{\text{out},i}$. We observe that *on average* the total utilization respects the configured limit:

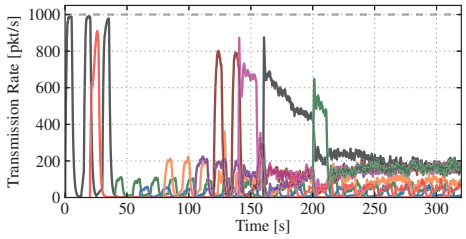


(a) Continuous traffic generation

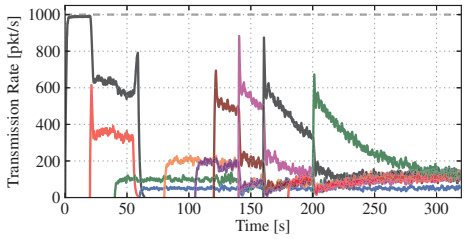


(b) Bursty traffic generation

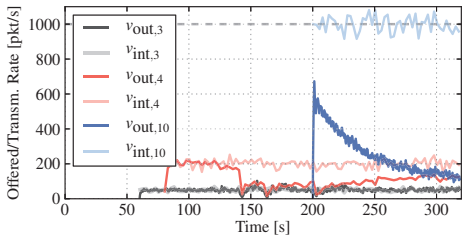
Fig. 6: Total offered load of 11 nodes and total network utilization. The rate limit of $v_{\max} = 1000\,\text{pkt}/\text{s}$ is fully respected on average.



(a) Bursty traffic



(b) Continuous traffic



(c) Details on continuous traffic tests

Fig. 7: (a,b): Individual transmission rates $v_{\text{out},j}$ of the 11 nodes responding to each others transmission rates. (c): Offered load per node $v_{\text{int},j}$ and effective transmission rate $v_{\text{out},j}$ for continuous-traffic tests: Node 3 starts at $t = 60\,\text{s}$, node 4 at $t = 80\,\text{s}$, and node 10 at $t = 200\,\text{s}$.

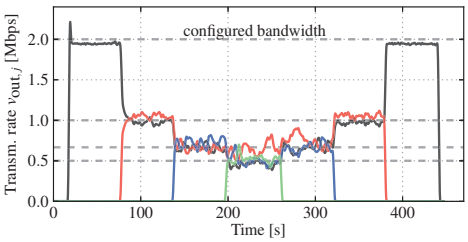


Fig. 8: Transmission rates $v_{\text{out},j}$ of four nodes in a real deployment. The chemical rate controller triggers the egress device queue in the Linux traffic control system of the corresponding nodes. Traffic is generated by TCP ssh/scp sessions. The aggregate rate limit for all rate-controlled traffic was set to $v_{\max} = 2\,\text{Mbps}$ (via E_0), the filter coefficient to $k_{\text{F}} = 100\,1/\text{s}$, and the feedback channel delay was 20 ms.

$v_{\text{out,tot}} \leq v_{\max}$ (even though *instantaneous* rates can assume any value).

Figure 7 shows that the *capacity allocation is fair*: The transmission rate of each node $v_{\text{out},i}$ converges to its share in response to the total offered load. The low-pass filter stage at the output of each controller effectively suppressed bursts in the offered load in order to prevent oscillations. Further experimentation with the parameter k_{F} showed its critical role. When the sensing delay is high, variations of k_{F} in the range of $(0,1]\,1/\text{s}$ can make the system too sensitive leading to high oscillations around the rate limit, or too slow in its adaptation. It is likely that in some cases a small statistical variation of k_{F} in response to path variability might yield best results.

Finally, Fig. 7(c) details the relation between transmission rate and system load for a few individual nodes for continuous-traffic tests: Host 3, who was not overloaded (as it generated packets at a very low rate), did not experience rate limitation. Host 4, who generated data slightly above its permissible transmission rate, experienced a slight rate regulation. Host 10, who was significantly overloaded, experienced major rate regulation (it even did not match its packet generation rate as it entered the network).

B. Real world deployment

The setup involved four hosts, one on a wired network (1000baseT) and the other three on a wireless network (802.11 n). The feedback was provided by an out-of-band UDP channel, and the rate controller was configured to regulate the node's ssh/scp traffic class (destination port 22). Each node used the ssh service for remote login and transferred a large file at different time intervals and for various time durations.

Figure 8 reports the results and confirms the correct operation of the system. Every time a new node accesses the service, the total capacity is divided fairly among the nodes on average: 1 node at 2 Mpbs, 2 nodes at 1 Mpbs per node, 3 nodes at 0.65 Mbps per node, and 4 nodes at 0.5 Mbps per node. This was also validated by transfer rates reported at each scp session.

One observation is that as the number of nodes accessing the service increases, the stochasticity did so too (although on average it converged to the per-node share). A possible explanation could come from how the TCP exits from the fast-

recovery phase when the rate limit is lowered (i.e., whether it succeeds to stay in congestion avoidance or if it drops to slow start and then re-enters congestion avoidance). Further future exploration is needed to establish a conclusive assessment of the situation and refinement of the controller.

A second observation is that the allocated aggregate capacity was asymptotically matched but never 100 % utilized, even in the single node periods. In fact, this observation also manifested in the simulation scenario. This is probably related to small variances of the feedback signals (expressed through c_T) and c_S that keep reaction r_3 always active even at a very low speed.

VI. Discussion

Chemical distributed rate controllers collaborate to optimally meet a globally-predefined rate limit. The reinforcement of this limit is *soft*: the controllers do not synchronize the transmissions (i.e., the *instantaneous* aggregate rate can assume any value). Instead over time, they cooperatively tend to converge to a rate value, proportional to their load and to the time they have been active, which ensures that the *average* aggregate rate respects the limit.

Each controller actively monitors and infers the current network load from the aggregate transmission rate, without requiring knowledge of the number or identity of other transmitters. The inference process does not need any explicit or sophisticated estimation model: the LoMA-based scheduling effectively bridges workloads and rates through a simple predictable relationship that underpins the operation of the reaction network. As concentrations of molecules are a direct linear expression of workload, the granularity at which the system operates can be easily adjusted from the byte level to the packet level (or anything in-between).

Sporadic problems in the sensing of the network utilization have marginal and instantaneous-only effects: Thanks to how the intrinsic regulation process works, a fractional variation of a molecular concentration, due to errors, has little effect on the operative concentration level.

With our algorithm, the node's *instantaneous* transmission rate is guaranteed to respect the predefined maximum threshold. This capability stems from the properties of the enzymatic reaction pattern in Fig. 2. Note that, an intuitive chemical pattern that captures exactly the typical transient behavior of the classic token-bucket with feedback-control is the one represented in Fig. 9. However, in contrast to the traditional token-bucket dynamics, the basic scheme we used (Fig. 2) has smoother transient characteristic and is free from overshooting-problems. As a result, it is suitable to operate also in bursty-traffic conditions, which are more common in Internet traffic.

The core design we have presented is easily extendable (see [19]) with additional chemical reaction patterns, in order to refine the sensing inputs (total utilization, partial utilization), as well as how the system distributes rates (equally, proportionally to traffic class, etc.). This extensibility alongside the parameterizability of the core design can lead to a family of rate controllers suitable for different applications and environments.

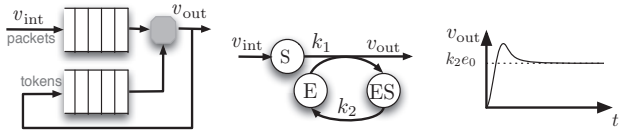


Fig. 9: Classical token-bucket scheme with feedback-control. (e_0 represents the fixed amount of molecules in the loop: $c_E(t)+c_{ES}(t),\ \forall t$.)

At the microscopic level a chemical reaction network as well as a traditional queueing network is described by a stochastic process – a continuous time discrete space Markov jump process. At the same time there is a subtle difference between the two approaches in the design of network systems. Using queuing theory, one starts by engineering the interaction of independent arrival and departure processes of a queue, and observe the resulting effects when queues interact. In chemical design, the departure process links directly to the queue's fill level and thus, we can directly focus on engineering the interactions (reactions) of queues because the fill-level-proportional scheduling promotes the system's self-stabilization.

At the macroscopic level, control theory and LTI analysis, which are common-ground in the design of many networking algorithms, can be used also to analyze the system behavior and leverage the design process. Two unique and very useful properties of our chemical metaphor in this process are: (a) the direct connection of an accurate flow-model to an algorithm that allows any of the two to be extracted from the other in a straight-forward manner, and (b) the exact one-to-one mapping of model parameters to algorithm variables.

VII. Conclusions and Future Outlook

We have presented an adaptive rate controller that can be used for different purposes and environments where distributed rate control is needed. We have used a unique design methodology that derives from chemical engineering, and we have successfully validated both the performance of the proposed system as well as the credibility of the methodology, through simulations and real world tests.

In the future, a performance-comparison with other existing traffic shaping mechanisms is planned, as well as a more extensive real-world test to reveal the nature of interactions with existing congestion control schemes and traffic profiles of different services. Additionally, we plan to extend the here presented distributed rate controller to enable the weighting of admitted flows and thus an adjustable (proportional) rate allocation.

References

[1] M. Monti and M. Sifalakis, "Design, analysis and performances of chemical-inspired rate controllers in packet networks," Univ. of Basel, http://cn.cs.unibas.ch/pub/doc/cs-2013-001.pdf, Tech. Rep., Jan. 2013.

[2] B. Raghavan, K. V. Vishwanath, S. Ramabhadran, K. Yocum, and A. C. Snoeren, "Cloud-control with distributed rate limiting," in *Proc. of ACM SIGCOM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Aug. 2007, pp. 337–348.

[3] H. Abu-Libdeh, Y. Vigfusson, K. Birman, and M. Balakrishnan, "Ajil: Distributed rate-limiting for multicast networks," Cornell Univ., Tech. Rep., 2008.

[4] B. Briscoe, "Flow rate fairness: dismantling a religion," *ACM SIGCOM Computer Comm. Review (CCR)*, vol. 37, no. 2, pp. 63–74, Mar. 2007.
[5] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.
[6] A. Charny, D. Clark, and R. Jain, "Congestion control with explicit rate indication," in *Proc. of IEEE Int'l Conf. on Communications (ICC)*, vol. 3, Seattle, Jun. 1995, pp. 1954–1963.
[7] S. S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. on Networking*, vol. 1, no. 4, pp. 397–413, Aug 1993.
[8] S. Floyd, K. K. Ramakrishnan, and D. Black, "The addition of explicit congestion notification (ecn) to ip," RFC 3168, IETF, 2001.
[9] L. Massoulie, "Stability of distributed congestion control with heterogeneous feedback delays," *IEEE Trans. on Automatic Control*, vol. 47, no. 6, pp. 895–902, 2002.
[10] G. Hasegawa and M. Murata, "Tcp symbiosis: Congestion control mechanisms of tcp based on lotka-volterra competition model," in *Proc. of ACM Workshop Inter-Perf*, 2006.
[11] P. Antoniou and A. Pitsillides, "A bio-inspired approach for streaming applications in wireless sensor networks based on the lotka-volterra competition model," *Computer Comm.*, vol. 33, pp. 2039 – 2047, 2010.
[12] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, "A control theoretic analysis of RED," in *Proc. of IEEE INFOCOM*, Apr. 2001, pp. 1510–1519.
[13] C. V. Hollot, V. Misra, D. Towsley, and W.-B. Gong, "On designing improved controllers for aqm routers supporting tcp flows," in *Proc. of IEEE INFOCOM*, Apr. 2001, pp. 1726–1734.
[14] N. Xiong, L. T. Yang, Y. Zhang, Y. Zhou, and Y. Li, "Design and analysis of a stable queue control scheme for the internet," in *Proc. of IEEE Conf. on Embedded and Ubiquitous Computing (EUC)*, 2008, pp. 378–384.
[15] K. Jacobsson, L. L. H. Andrew, A. Tang, K. H. Johansson, H. Hjalmarsson, and S. H. Low, "Ack-clocking dynamics: Modelling the interaction between windows and the network," in *Proc. of IEEE INFOCOM*, 2008, pp. 2146–2152.
[16] A. Tang, L. L. H. Andrew, K. Jacobsson, K. H. Johansson, H. Hjalmarsson, and S. H. Low, "Queue dynamics with window flow control," *IEEE/ACM Trans. on Networking*, vol. 18, no. 5, pp. 1422–1435, 2010.
[17] T. Meyer, "On chemical and self-healing networking protocols," Ph.D. Thesis, Univ. of Basel, Switzerland, 2010.
[18] M. Monti, T. Meyer, C. F. Tschudin, and M. Luise, "Signal processing applied to chemically inspired communication protocols," in *Proc. of IEEE Int'l Conf. on Communications (ICC)*, Ottawa, Jun. 2012.
[19] M. Monti and M. Sifalakis, "Extending the artificial chemistry to design networking algorithms with controllable dynamics," Univ. of Basel, http://cn.cs.unibas.ch/pub/doc/cs-2012-003.pdf, Tech. Rep., Jul. 2012.