EXPLOITING CHEMISTRY FOR BETTER PACKET FLOW MANAGEMENT

R. TRAYLOR, PH. D.

ABSTRACT. We review a 2011 technical report that discusses novel applications of chemical kinetics and artificial chemistries to queuing and flow models in computer networking. Basics of chemical kinetics are covered, and additional analysis and commentary is given. Our overall recommendation is that this body of work deserves further study and implementation.

Perhaps the phrase "don't reinvent the wheel" is overused. However, many newer disciplines, particularly in the technology sector, seem to insist on it. One thing physical engineers learned long ago was to study the world around them, work with it, and emulate it in their designs. Network engineering should be no different. In a technical report from 2011, authors Thomas Meyer and Christian Tschudin from the University of Basel describe a highly elegant natural flow management method [11] that exploits much of the hard work done in the chemistry sector in chemical kinetics. They describe a scheduling approach that creates an artificial chemistry as an analogue of a queueing network, and uses the Law of Mass Action to schedule events naturally. The analysis of such networks utilizing their implementation is simplified regardless of the depth one wishes to go into. In addition, they show a congestion control algorithm based on this framework is TCP fair and give evidence of actual implementation (a relief to practitioners).

This report will discuss their paper at length with a goal of covering not only their work, but the underlying ideas. Since the paper requires knowledge of chemical kinetics, probability, queueing theory, and networking, it should be of interest to specialists in these disciplines, but a comprehensive discussion of the fundamentals glossed over in the paper would make dissemination more likely.

§1. Overall Review of the Paper

The paper is well written, readable, and quite clear despite the breadth of scope it contains. It's a major benefit to the authors and readers that their theoretical model has been tested in a proof-of-concept network. The work shows much promise for the networking space as a whole.

§2. Overview of the Paper and Chemistry Basics

Just as in chemistry and physics, packet flow in a network has microscopic behavior controlled by various protocols, and macro-level dynamics. We see this in queueing theory as well–we can study (typically in steady-state to help us out, but in transient state as well) the stochastic behavior of a queue, but find in many cases that even simple attempts to scale the analysis up to networks (such as retaining memorylessness) can become overwhelming. What ends up happening in many applied cases is a shift to an expression of the macro-level properties of the network in terms of average flow. The cost of such smoothing is an unpreparedness to model and thus deal effectively with erratic behavior. This leads to over-provisioning and other undesirable and costly design choices to mitigate those risks.

Date: January 21, 2019.

Meyer and Tschudin have adapted decades of work in the chemical and physical literature to take advantage of the Law of Mass Action, designing an artifical chemistry that takes an unconventional non-work-conserving approach to scheduling. Non-work-conserving queues add a delay to packets and have tended to be avoided for various reasons, typically efficiency. Put simply, they guarantee a constant wait time of a packet regardless of the number of packets in a queue by varying the processing rate with fill level of the queue. The more packets in queue, the faster the server processes those packets.

2.1. LAW OF MASS ACTION IN CHEMISTRY. If we have some chemical reaction with reactants $A_1, A_2, ..., A_n$ and products $B_1, B_2, ..., B_m$, and the reaction is only forward (obviously we can deal with both directions in chemistry to model chemical equilibrium, but only the forward part makes sense for our networking analogue), then we may express the reaction as

$$A_1 + A_2 + \ldots + A_n \longrightarrow B_1 + B_2 + \ldots + B_n,$$

where *k* is the rate constant. In a simple reaction $A \rightarrow P$, with *P* as the product, we can see the rate expressed nicely in a very basic differential equation form [9]:

$$-\frac{\mathrm{d}c_A}{\mathrm{d}t} = k \cdot c_A$$

This should actually look somewhat similar to problems seem in basic calculus courses as well. The rate of change of draining the reactant is a direct function of the current concentration.

The reaction rate r_f of a forward reaction is proportional to the concentrations of the reactants:

$$r_f = k_f c_{A_1} c_{A_1} \cdots c_{A_N}$$

for a set of reactants A_i .

2.2. THE QUEUING ANALOGUE AND ASSUMPTIONS. Meyer and Tschudin [II] express the networking version of these chemical reactions in a very natural way. Packets are molecules. A molecular species is a queue, so molecules of species X go into queue X. The molecular species is a temporary buffer that stores particular packets types until they are consumed by a reaction (processed by some server in the queueing space). FIFO (first-in-first-out) discipline is assumed.





FIGURE 1. Reproduced from Meyer and Tschudin

Figure 1 from the technical report shows how a small system of reactions looks in the chemical space and the queuing space. Where analysis and scheduling can get complicated is in the coupled nature of the two reactions. The servers both drain packets from queue Y, so they are required to coordinate their actions in some way. It's important to note here that this equivalence rests

on treating the queuing system as M/M/I queues with a slightly modified birth-death process representation.



FIGURE 2. A Birth-Death Process with state-dependent service rate

Typically, in an M/M/1 queue, the mean service rate is constant. That is, the service rate is independent of the state the birth-death process is in. However, if we model the Law of Mass Action using a birth-death process, we'd see that the rate of service (analogously, the reaction rate) changes depending on the fill-level of the queue (or concentration of the reactant). We'll investigate this further in the next sections, discussing their formal analysis.

§3. Related Work and precedent

The authors noted that adding packet delay is not unheard of in the networking space. Delay Frame Queuing [12] utilizes non-work-conserving transmission at edge nodes in an ATM network in order to guarantee upper bounds on delay and jitter for virtual circuits. Three other researchers in 2008 (Kamimura et al) proposed a Constant Delay Queuing policy that assigns a constant delay to each packet of a particular priority stream and forward other best-effort packets during the delay [8].

§4. Formal Model of Artificial Packet Chemistry

Note: for those more interested in the application and implementation review and discussion, this section can be skipped.

4.1. ARTIFICIAL CHEMISTRY. The notion of an formal artificial chemistry has been around for some time. There is an excellent paper by Dittrich, Ziegler, and Banzhaf that gives a survey of the work in this area [1]. Put simply, an artificial chemistry is a tuple $(S, \mathcal{R}, \mathcal{A})$ where $S = \{s_1, s_2, \ldots, s_n\}$ is a set of all valid molecules, \mathcal{R} is a set of rules r that describe interactions between molecules, and \mathcal{A} is the reactor algorithm that determines how the set of rules in \mathcal{R} is applied to a collection of molecules termed \mathcal{P} . \mathcal{P} may be a reaction vessel, reactor, or "soup" (as Dittrich et al call it). It's also notable that \mathcal{P} cannot be identical to S. (The reason given in the paper is that some molecules might be present in many exemplars, but not all.)

To expand a bit more, we'll note that the rules $r \in \mathcal{R}$, all take the form

$$s_1 + s_2 + \ldots + s_n \longrightarrow s'_1 + s'_2 + \ldots + s'_m$$

where all $s \in S$. These rules are fairly abstract, and don't explicitly seem to describe just a reactant to product type reaction. These can be collisions or other types of interactions. The set *S* of valid molecules presumably can be partitioned into disjoint subsets of different species of molecule as well, though the representation in [1] is more general.

Regarding the reactor algorithm \mathcal{A} , Dittrich et al [1] give several different descriptions/approaches by which it can be defined, depending on whether each molecule is treated explicitly, or all molecules of a type are represented by a single number (frequency or concentration):

- (1) Stochastic molecular collisions: Every single molecule is worked with, where a sample of molecules from the reaction vessel \mathcal{P} is drawn and the algorithm checks to see if a particular rule $r \in \mathcal{R}$, applies.
- (2) Differential Rate Equations: This approach seeks to describe the dynamics of a chemical system using concentrations of molecular species. The rules under this algorithm take a species approach:

$$r: a_1s_1 + a_2s_2 + \ldots + a_Ns_N \longrightarrow b_1s_1 + b_2s_2 + \ldots + b_Ns_N$$

Here, the s_i 's are species, not individual molecules. The coefficients are stoichiometric factors of the reaction. They are simply indicator functions to denote whether species s_i is a reactant or product. That is $a_i = 1$ if and only if s_i is a reactant in the rule r, and $b_i = 1$ if and only of s_i is a product in the rule r. It is this form of \mathcal{A} that Meyer and Tschudin [II] utilize in their packet chemistry.

The change of overall concentration (concentration denoted c_{s_i}) is given by a system of differential equations

$$\frac{\mathrm{d}c_{s_i}}{\mathrm{d}t} = (b_i - a_i) \prod_{j=1}^N c_{s_j}^{a_j}, \quad i = 1, \dots, N$$

according to the Law of Mass Action discussed earlier. There may be multiple rules/reactions $r \in \mathcal{R}$, that affect the concentration of species s_i , so

$$\frac{\mathrm{d}c_{s_i}}{\mathrm{d}t} = \sum_{r \in \mathcal{R}} \left[(b_i^r - a_i^r) \prod_{j=1}^N c_{s_j}^{a_j^r} \right], \quad i = 1, \dots, N$$

(3) Others: There are other options, such as metadynamics (where the number of species and thus differential equations may change over time), mixed approaches, or symbolic analysis of the differential equations.

According to Dittrich et al [1], the reactor algorithm \mathcal{A} depends on the representation of the elements of s_i and thus the population. Meyer and Tschudin utilize the second approach, though they do not explicitly state this.

4.2. ARTIFICIAL PACKET CHEMISTRY. Meyer and Tschudin adapt the artificial chemistry in the previous section to suit their queuing networks in a given computer network. They add an element G to the artificial chemistry tuple to get an artificial packet chemistry $PC = (G, S, \mathcal{R}, \mathcal{A})$, where G is the digraph that gives the topology of the computer network. G consists of a set of nodes V_G which represent the chemical reaction vessels. (These were the \mathcal{P} in the previous section.), and E_G is the set of directed arcs that represent network links connecting adjacent nodes. (That is, if flow between nodes can happen in both directions, they will be represented by two arcs. One going in one direction, one in the other.)

Here, since a molecular species is the analogue of a queue, $S = \bigcup_{i \in V} \{S_i\}$. Here, $\{S_i\}$ is a set of all queue instances in a particular node *i*. At this point, some discussion on clarity is warranted. It is possible to have more than one queueing instance (more than one molecular species) inside each reaction vessel (here the nodes of the network). I don't think this is meant to be a disjoint union, since a reaction species can show up in more than one reaction vessel, so there may be repeats of certain species in this representation of *S* when written this way. Perhaps it's just a nitpick, but it's worth mentioning.

 $\mathcal{R}_{i} = \bigcup_{i \in V} \{R_i\}$ gives all the flow relations among the queues. Here the rules *r* take form 2 of \mathcal{A} :

$$r \in R_i: \sum_{s \in S_i} a_{s,r}s \longrightarrow \sum_{s \in S_i \cup \{S_j: j \in N_i\}} b_{s,r}s$$

The reaction rules basically describe what's going on in a particular reaction vessel. We can send packets to neighboring nodes/vessels (N_i is the notation for the neighborhood of node *i*, or the set of adjacent nodes), or we can keep packets in the same node after the reaction is done. The reactions that send packets to neighboring nodes are transmissions.

The mean reaction rate v_r of each reaction is given by the Law of Mass Action as applied to forward reactions:

$$\nu_r = k_r \prod_{s \in S} c_s^{a_{s,r}},$$

just as described in the previous section.



Figure 3: Example Network for the Disperser Protocol – A distributed LoMA-scheduled queueing network that globally averages the fill levels of the queues.



Figure 3 from Meyer and Tschudin gives an explicit example to help solidify these abstract ideas. The network consists of 4 nodes, so $V = \{n_1, n_2, n_3, n_4\}$. Each node has a bidirectional link with its neighbors, so $E = \{n_1n_2, n_2n_1, n_2n_3, n_3n_2, n_2n_4, n_4n_2, n_3n_4, n_4n_3\}$. In this case, we only have one species of molecule (one queue) per node, so $S = \{X_1, X_2, X_3, X_4\}$. The set of reactions is simply a first-order reaction per arc: $\mathcal{R} = \{r_{a,b} : X_a \to X_b : ab \in E\}$

From a review standpoint, I would have liked to see a less trivial example, such as one with multiple queues in a node, and rules that may keep packets in a node instead of just transmitting. These types of scenarios would be interesting to model this way, and demonstrate better the power of this approach.

§5. Formal Analyses of Artificial Packet Chemistry

This section can also be skipped for those less interested in the formal mathematics. Suffice it to say that there are a multitude of already created methods now available for the elegant analysis of computer networks when modeled by an artificial packet chemistry.

By representing packet flow in a computer network as an artificial chemistry, a multitude of analyses are available, from high to low granularity. The authors give a heavily brief survey (and a good bibliography) of works that can be utilized to analyze these networks pulled from the physics and chemistry literature. A particular advantage of this method is the ability to study the transient states of the network rather than just steady states. The authors also claim the ability to determine the stability of the network flow based only on topology, a heavy advantage in design.

5.1. STOCHASTIC ANALYSIS AT THE MICROSCOPIC LEVEL. The stochastic behavior of chemical reaction networks is described by the chemical master equation [10] which takes the form

$$\frac{\mathrm{dP}}{\mathrm{d}t} = \mathrm{AP},$$

which is a differential equation describing the evolution of state probabilities for a system. Here the states are discrete, and time is continuous. The matrix A describes the transition rates (which can also be kinetic or reaction rates), and the stochastic process described is a Markov jump-process Since we're on a network, the Markov jump process exists in an *S*-dimensional integer lattice. Some work has been done to analyze several classes of chemical reaction networks to find the steady-state probability distribution of the state space. For example, if the total number of packets in the network has a bound, and the network contains only first order (unimolecular to unimolecular) reactions, the steady state probability distribution for the lengths of the queues in the network is a multinomial distribution [3]. On the other hand, if the network is open (we allow packets to exit the network completely), then the steady state probability distribution of the lengths of the queues follows a product of Poisson distributions (which is also Poisson) [3]. (This is an extremely desirable property, called a product-form.)

5.2. DETERMINISTIC APPROXIMATIONS. This is the most common approach utilized in computer network analysis today, simply because networks are so large and complex that stochastic modeling becomes too cumbersome. Here, the average trajectory is represented by a system of ordinary differential equations, building a fluid model. One downside to this in the networking space is that the analysis of protocols by this method requires manual extraction from source code and accuracy is uncertain.

In the chemistry sector (and now in the packet chemistry model), obtaining a fluid approximation is not only easier, but shown to be accurate. There are links between the stochastic master equation to several approximations [5, 6] including a deterministic ODE model. Gillespie [6] showed that the ODE model accurately predicts the network flow trajectory in many cases.

One thing the authors note here is that the ODE model can be directly and automatically generated from the network topology. For example, a single server with a single queue (M/M/I) is simply modeled as one chemical species X. The arrival rate (inflow) is λ , and the service rate is proportional to the queue length, so $\mu = kx$, where x is the queue length. Then we get a simple differential equation

$$\dot{x} = \lambda - kx,$$

describing the change in queue length as the difference of inflow and outflow. In the steady state, $\dot{x} = 0$, which lets us look for a fixed point $\hat{x} = \frac{\lambda}{k}$. This is the steady-state queue length, which allows us to derive the expected waiting time $T = \frac{1}{k}$, showing that the latency of a packet under this model is independent of the arrival rate and fill level. This model when implemented automatically adjusts the service rate such that in the steady state, every packet sees the same latency.

It's also important to determine just how stable this steady state is by analyzing the sensitivity of the network and states to perturbations. The authors list several citations to show that no new approaches are needed to do this; one can look to signal and control theory literature. In particular, a network designer would desire to predict the stability of a complex network by studying the topology as opposed to an analysis of the system of ODEs. Fortunately, modeling a network this way allows for the use of the Deficiency Zero Theorem for complex chemical networks that gives conditions for stability of steady-state [2, 7]. The authors give a formal convergence proof that the example network of Figure 3 converges to a stable fixed point and is asymptotically stable, comparing it to the proof of a similar protocol *Push-Sum* (a gossip protocol in computer networks).

. Implementation of a Scheduler Based on the Law of Mass Action

Likely, this section will be of greatest interest to network engineers. The authors have indeed designed and implemented a scheduler that utilizes this approach in an elegant fashion. In addition, they discuss a "chemical control plane" that can automatically be compiled from the abstract model. In another application, they relax the static nature of the network to allow an active networking approach that reshapes the queuing network at run-time. The authors do discuss specifics of implementation, though this article will only briefly touch on it.

6.1. SCHEDULER. Each network node/reaction vessel has its own scheduler. The scheduler computes the next occurrence time of each rule $r \in R_i$ in its local node (this is equivalent to "serving" or processing a packet or set of packets for bimolecular reactions) according to the Law of Mass Action. It then will sort the events into a priority queue, wait until the first event occurs, then execute. The main difficulty for a scheduler is to dynamically react and reschedule events properly as packets are added to or drained from its queues. The authors note that an efficient mass action scheduler can be implemented that requires only $O(\log(|\mathcal{R}_{\downarrow}|))$ time to enqueue or dequeue packets. This is based on the Next Reaction Method [4] of Gibson and Bruck.

Here we'll recount an explicit example that illustrates the concept. If we return to Figure 1 reproduced below, we can walk through Meyer and Tschudin's scheduler implementation.



Figure 1: Bimolecular reaction in two different settings – A bimolecular reaction extracts an instance of two chemical species (= packet queues), X and Y. At the same time, the queue Y is drained by a second server (= monomolecular reaction).

FIGURE 4. Reproduced from Meyer and Tschudin

There are two queues, X and Y. Reaction 1 (Server 1) is bimolecular: $X + Y \rightarrow Z$, so the server pulls packets from two queues to execute the service. Reaction 2 (Server 2) is unimolecular, pulling only from queue Y. If we assume the reaction constants $k_1 = 1000/(\text{packet} \cdot s)$ and $k_2 = 1000/\text{s}$, that X begins with two packets in its queue, and Y begins with 3 packets in its queue, then the reaction rates v_r , r = 1, 2 are respectively $v_1 = k_1 c_X c_Y = 1000 \cdot 2 \cdot 3 = 6000$ and $v_2 = k_2 c_Y = 1000 \cdot 3 = 3000$. The occurrence time is the reciprocal of the reaction rate, so the occurrence times τ_r are respectively $\tau_1 = \frac{1}{6}ms$ and $\tau_2 = \frac{1}{3}ms$. That means the first server executes its action first, extracting packets from both X and Y.

Since the occurrence time of r_2 is coupled with r_1 (both servers pull from queue Y), the action of r_1 requires a rescheduling of r_2 . After r_1 pulls a packet each from X and Y, there is 1 packet

left in X and 2 in Y, which means we have to recalculate the rate $v_2 = 1000 \cdot 2 = 2000$. The occurrence time of r_2 is at ms $\frac{1}{3}$, so its time of execution hasn't arrived. But thanks for r_1 's effect, we have to rescale and reschedule the occurrence time of r_2 . This is done by the following:

$$\tau_{r,\text{new}} - \frac{\nu_{r,\text{new}}}{\nu_{r,\text{old}}}(\tau_{r,\text{old}} - t_{\text{now}}) + t_{\text{now}},$$

where $(\tau_{r,old} - t_{now})$ is the time remaining between the original execution time and the current time. The multiplier in front is a scaling effect.

In this example, at $t_{now} = 1/6ms$, r_2 was supposed to go at time 1/3ms, but will now be prolonged.

A note here, I did the math for their specific example, and it seems off. I think the multiplier should be as I've written above. The authors wrote the reciprocal, which prolongs too far. I'll work to contact the authors to verify this.

There are other timed scheduling algorithms utilized in computer networking, such as Earliest Deadline First, which require tagging each packet with a timestamp. This scheduler does not require such an imposition.

6.2. THE CHEMICAL CONTROL PLANE. The authors discuss another application of a LoMAscheduled queuing network – traffic shaping. Traffic shaping is a management technique utilized to manipulate and prioritize network traffic with a goal of reduce the impact of heavy use cases. It should be evident that this sort of management can either be extremely complex or lead to ad-hoc "empirical" techniques.

Here, the authors describe what they term as a chemical control plane that is intended to avoid the messy necessity of sending packets through a complex queueing network in order to shape packet flow as desired. The control plane takes advantage of concepts in enzymatic chemical reactions in order to control flow. This is a different application than the flow networks discussed thus far (as I understand it).



FIGURE 5. The chemical flow architecture

Here the forwarding plane which executes actions is separated from the control plane which will shape the flow of packets in the forwarding plane. The chemical control plane will dynamically determine the service rates; the servers do not have them predefined. There are some number of FIFO queues *n*, one for each type of ingress packet flow and they are drained by one server

each, representing a unimolecular reaction. In the control plane, each queue is represented by an input species X_i and product species X_i^* . The chemical reaction network lives abstractly in the control plane, which is designed by a traffic engineer and can look like any digraph or network he wishes.

Here we note the difference here between the prior sections, which dealt with physical flows modeled by a chemical reaction network, and moving the chemical reaction network to an abstract control plane. The queues now are not necessarily physically linked together, but we can choose to couple them abstractly to shape traffic.

When a packet physically enters one of the queues, the control plane injects one instance of the corresponding molecule species into the abstract network. The scheduler described previously is implemented and eventually an instance of the output species is generated. Once this happens, the corresponding server in the forwarding plane physically processes the packet and dequeues the next. The advantage here is that the abstract molecules in the control plane have no payload, so implementation of this model only requires storing an integer value for each species that keeps track of the number of packets in each queue. This allows analysis of behavior at the design phase.

In the simplest case, a unimolecular reaction $X \to X^*$ in the chemical control plane acts like a low-pass filter to the packet flow, smoothing bursts with high frequency components. If the differential equation $\dot{x} = \lambda - kx$ that approximates a unimolecular reaction is converted to the frequency domain via the Laplace transform, the transfer function F(s) has a cut-off frequency at k, the reaction constant:

$$F(s) = \frac{\mu(s)}{\lambda(s)} = \frac{k}{s+k}$$

That is, higher-frequency flows will be attenuated, much like dark glasses do with sunlight. Applying this filter at an ingress point of a network leads to less chaotic traffic patterns, but with a cost of a delay $\frac{1}{k}$ and memory to buffer the packets. Therefore, the mean queue length for this single queue will grow proportionally with the delay and flow rate. That is, $\hat{x} = \frac{\lambda}{k}$

Another consideration of the LoMA queues described by Meyer and Tschudin that differs from the standard M/M/I queuing models is that the service rate is ultimately unbounded (for infinite capacity queues/networks), since it is proportional to the queue length. This is undesirable to allow in a network, and thus the authors borrow from biological systems and design an abstract enzymatic reaction to limit the rate of packet flow.

In biological systems, enzymes bind to reactant molecules X, called substrates in order to prevent a particular molecule from reacting immediately. Some amount of enzyme molecules E exist, and they can either exist free-form or bound in a complex (EX). The more enzyme molecules in bound form, the slower the *rate* of transmission grows for an increasing arrival rate. At equilibrium, the influx and efflux of substrate-enzyme complex molecules are equal according to Kirchoff's Law, so

$$k_w c_X c_E = k_s c_{EX}$$

Take a look at Figure 5 in the chemical control plane to see this action. The number of enzymes is constant, so $c_E + c_{EX} = e_0$, which yields the Michaelis-Menten equation, expressing the transmission rate μ in terms of the queue length c_X .

$$\mu = \nu_{\max} \frac{c_X}{K_M + c_X},$$

which yields a hyperbolic saturation curve. $v_{\text{max}} = k_s e_0$, and $K_M = \frac{k_s}{k_w}$ and specifies the concentration of X at which half of v_{max} is reached.

When the queue length at queue X is high, the transmission rate converges to v_{max} , and behaves like a normal unimolecular reaction when queue length is short.

The authors also extend this model to handle dynamic changes to the topology of the queuing network, which means that instances of queues and flow relations can be generated "on the fly," as it were. Tschudin [13] has created an executable string and multiset rewriting system called *Fraglets* that allow for the implementation and running of protocols based on the ideas put forth thus far. They describe in the paper how to implement explicitly the enzymatic rate-limiter in the chemical control plane in Figure 5. In this implementation, rather than flow interactions being static and determined at the design phase, each fraglet (packet) sorts itself into a queue. After a packet is serviced, the header of a fraglet is treated as code, allowing a packet to determine its route comparable to active networking. The relationship between the abstract model and execution layer remains, which allows a mathematical model of the behavior of a Fraglets implementation to be generated automatically, and a queuing network to be design and then realized easily in Fraglets language.

§7. CHEMICAL CONGESTION CONTROL ALGORITHM



(b) C3A⁺, additionally reacting to RTT variations

Figure 13: Reaction network of two chemical congestion control algorithms – The reaction network in (a) is reused in (b) (shaded in grey) and extended by new species and reactions (black) in order to sense and react to round-trip-time (RTT) variations.

As a final application of chemical networks, the authors combine LoMA-scheduled queues and flow-filtering patterns to schedule segments of a transport protocol to control congestion. To illustrate, they re-implement the additive increase/multiplicative decrease of the congestion avoidance mode of TCP-Reno. The congestion control algorithm reacts to packet loss automatically and naturally.

The reproduced figure above shows how the chemical congestion control is implemented as a chemical reaction network (and by extension, a queueing network).

- (I) Arriving packets are put into a queue *D*. The transmission rate v_{tx} is controlled by the quantity of pacemaker molecules *R*, so $v_{tx} = k_1 c_R c_D$, once again according to the Law of Mass Action. To mimic the additive (linear) increase mechanism of TCP-Reno, the number of pace-maker molecules is increased at a rate v_{inc} .
- (2) Before packets are transmitted, they are tagged with a sequence number. If there is a gap in the sequence number of acknowledgments from the destination, the source regenerates the packets at a queue L.
- (3) A lost packet will catalyze the destruction of pacemaker molecules by another reaction r₂, which will lead to the exponential decay of *R*—molecules and thus decrease the transmission rate. However, we wish to prevent too fast a destruction of pacemaker molecules, so a third reaction r₃ will delay the destruction.

The authors encourage the use of such a reaction graph at the design phase of flow management policies. The feedback nature is much clearer. In addition, the papers give a formal proof that their congestion control model is TCP-fair at equilibrium; that is, the transmission rate is proportional to $\frac{1}{\sqrt{p_{loss}}}$ where p_{loss} is the probability of packet loss between source and destination. They also discuss an extended version that reacts to variations in round trip time (RTT) variation to more fully exploit the link bandwidths. The traffic statistics are not computed symbolically with chemical reactions. Instead, another reaction builds a difference sch that at equilibrium the fill level of its queue is proportional to the excess transmission rate. That signal decays the pacemaker molecules. They also supply simulations to illustrate their implementations.

§8. Design Motifs

This section is of a particular interest to me personally, and was treated the least. No design process can ever be fully automated, but the authors claim to have developed several design motifs of chemical reaction networks for a variety of purposes, including arithmetic computation of fill levels to communication patterns (anycast, neighborhood discovery, etc). Unfortunately, they do not give a direct citation as to where to look further. This report will be updated when such information is found.



Figure 12: Reaction network motifs.

FIGURE 6. The only two motifs provided

Figure 6 shows the only two motifs provided by the authors. One for rate limiting (a) and the other bimolecular reaction to compute the difference between arrival rates for two queues. The concept of using these as design elements is extremely intriguing, and it was unfortunate that the authors did not choose to expand this further.

CONCLUSION

Meyer and Tschudin have given an extensive report showing how powerful the application of chemical kinetics and chemical networks can be for the computer networking space. There are several research opportunities available for further study and implementation. As yet, there have been no citations of this work of note (the report came out in 2011), and thus the opportunity seems ripe for exploration.

References

- Dittrich, P., Ziegler, J., and Banzhaf, W. "Artificial chemistries a review". In: Artificial Life 7 (3 2001), pp. 225–275.
- [2] Feinburg, M. "Complex balancing in general kinetic systems". In: Archive for Rational Mechanics and Analysis 49 (3 1972).
- [3] Gadgil, C., Lee, C., and Othmer, H. "A stochastic analysis of first-order reaction networks". In: *Bulletin of Mathematical Biology* 67 (5 2005), pp. 901–946.
- [4] Gibson, M. and Bruck, J. "Effcient stochastic simulation of chemical systems with many species and many channels". In: *Journal of Physical Chemistry* 104 (9 2000), pp. 1876–1889.
- [5] Gillespie, D. "The chemical Langevin and Fokker-Planck equations for the reversible isomerization reaction". In: *Journal of Physical Chemistry* 106 (20 2002), pp. 5063–5071.
- [6] Gillespie, D. "The chemical Langevin equation". In: *Journal of Chemical Physics* 113 (1 2000).
- [7] Horn, F. "On a connexion between stability and graphs in chemical kinetics". In: *Proceed*ings of the Royal Society of London 334 (1598 1973), pp. 299–330.
- [8] Kamimura, K., Hoshino, H., and Shishikui, Y. "Constant Delay Queuing for jitter-sensitive IPTV distribution on home network". In: *IEEE Global Telecommunications Conference* (2008).
- [9] Laidler, K. Chemical Kinetics. McGraw-Hill, 1950.
- [10] McQuarrie, D. "Stochastic Approach to Chemical Kinetics". In: *Journal of Applied Probability* 4 (3 1967), pp. 413–478.

- [II] Meyer, T. and Tschudin, C. Flow Management in Packet Networks Through Interacting Queues and Law-of-Mass-Action-Scheduling. Technical Report. University of Basel, 2011.
- [12] Pocher, H. L., Leung, V., and Gilles, D. "An application- and management-based approach to ATM scheduling". In: *Telecommunication Systems* 12 (1999), pp. 103–122.
- [13] Tschudin, C. "Fraglets- a metabolistic execution model for communication protocols". In: Proceedings of the 2nd annual symposium on autonomous intelligent networks and systems (2003).