Stability and Cycles in Multiclass Queueing Networks

Flora Spieksma
Mathematical Institute
University of Leiden
P.O.Box 9512
2300RA Leiden
the Netherlands
Email-address: spieksma@math.leidenuniv.nl

Abstract

It is well-known that multiclass networks maybe not be stable even when the load of each node of the network is smaller than 1. This is an undesirable phenomenon with respect to reliability of a network with a given configuration. An important question to address is therefore the following: can changes in processing speeds result in exploding queues even when the loads remain smaller than 1? How? Can we give (simple) conditions on the network configuration such that this cannot happen?

Our tool for studying this problem is through the space-time scaled asymptotics, which are in fact solutions of the so-called fluid equations. A problem here is, that fluid solutions may exist, that are not asymptotic scaling limits. Instead, we will study the scaling limit directly through the so-called "second vectorfield".

1 Model

We consider a network with the set of nodes $\mathcal{N} = \{1, \ldots, \mathfrak{n}\}$ and p job types. The set $\mathcal{J}(n)$ denotes the job types served at node n, with $\mathcal{J}(n)$, $i = 1, \ldots, \mathfrak{n}$, a partition of the set of nodes. By $n(j)$ we will denote the node serving job type j, i.e. $n(j) = n$ iff $j \in \mathcal{J}(n)$.

Each node n consists of a preemptive single server and $\mathcal{J}(n)$ infinite capacity buffers, storing the jobs according to their types. There is a fixed *work conserving* priority rule prescribing the type of the job to be served at any moment. By preemptiveness, whenever a higher priority job arrives at a node, service of a lower priority job immediately stops and the job is sent back to its queue.

We may number the jobs such, that the job type with the highest number has highest priority in a node. Thus we can partially order the set of job types by writing $j \prec k$ whenever j and k are served at the same node, and $j < k$, that is, j has lower priority than k.

The arrival stream of type k jobs is a Poisson stream with rate $\lambda_k \geq 0$. Type k jobs require an exponentially μ_k distributed amount of service. Upon service completion, the successor function $s : \{1, \ldots, p\} \to \{0, 1, \ldots, p\}$ prescribes the next type of the job *deterministically*; if $s(k) = 0$ then the type k job leaves the network. We assume that every job eventually leaves the network.

As a consequence each job type i also has a uniquely defined antecedent $a(i)$, which is set equal to 0 whenever $\lambda_i > 0$.

Finally we assume the arrival, service and routing processes to be independent of each other.

Because of this deterministic routing, with each type i we can associate a uniquely defined route $r(i, s) \in \{1, \ldots, p\}$, $s = 1, \ldots, s_i$ through the set of *types*: $a\big(r(i, 1)\big) = 0$, $s\big(r(i, s_i)\big) = 0$ and $r(i, s) = i$ for some value s. We can refer to the parameter s as the stage of the route through i.

For investigating stability, it is sufficient to associate a discrete time Markov chain $\{\xi_t = \{\xi_{i,t}\}_{i=1,\ldots,p}\}_t$ with the evolution of the number of jobs of the different types on discrete time instants. This Markov chain hhhas the following transition probabilities:

$$p_{xy} = \begin{cases} \lambda_k, & y - x = e(k), a(k) = 0 \\ \mu_k, & y - x = -e(k),\ k \in \mathfrak{p}(x), s(k) = 0 \\ \mu_k, & y - x = -e(k) + e(j), k \in \mathfrak{p}(x), s(k) = j \\ 1 - \sum_{z \neq x} p_{xz} & y = x, \end{cases}$$

where $\mathfrak{p}(x) \subset \{1, \ldots, p\}$ is the set of highest priority job types present in the different nodes in network state x.

Whenever the initial state x is given, we denote this by $\xi_t(x)$.

Notice that this Markov chain need not be irreducible. However, the empty state is reachable from any other state, so that there is at most one closed class. For shortness reasons we will not address the process evolution for inessential initial state. It can be shown that they do not play an essential role in the analysis (cf. [3]) provided some minor technical conditions are met.

Faces and induced chains

Let $\Lambda \subset \{1, \ldots, p\}$. Then the corresponding *face* B^Λ is defined as

$$B^\Lambda = \{x \in \mathbf{R}_+^p \mid x_i > 0, i \in \Lambda; x_i = 0, i \in \bar{\Lambda}\},$$

where $\bar{\Lambda} = \{1, \ldots, p\} \setminus \Lambda$.

Note that the jump probabilities are the same within each face, so that ξ_t is a *face-homogeneous* Markov chain. The set function $\mathfrak{p}(x)$ is therefore constant on faces and thus we can write $\mathfrak{p}(\Lambda)$ whenever $x \in \Lambda$. Also the mean drift vectors from points in the same face are equal and we can therefore write

$$V(\Lambda) = \sum_y (y - x)\, p_{xy},$$

for any given $x \in B^\Lambda \cap \mathbf{Z}_+^p$.

We can view Λ as a set of saturated queues. The corresponding induced chain ξ_t^Λ is the Markov chain on the set of type $j \in \bar{\Lambda}$ queues, given that the type $i \in \Lambda$ queues are saturated. Formally, ξ_t^Λ is the orthogonally projected Markov chain ξ_t unto the state space

$$C^\Lambda = \{y \in \mathbf{Z}_+^p \mid (x_0, y) = 0\},$$

where $x_0 \in B^\Lambda \cap \mathbf{Z}_+^p$ is fixed but arbitrary, i.e. it has transition probabilities

$$p_{xy}^\Lambda = \sum_{\substack{z \in \mathbf{Z}_+^p: \\ (y + x_0, y + x_0 - z) = 0}} p_{x + x_0, z}.$$

The induced chain ξ_t^Λ is *almost* a priority network on the set of types $i \in \bar{\Lambda}$. Indeed, there is some additional input whenever node n is empty and $\mathfrak{p}(\Lambda) \cap \mathcal{J}(n) \neq \emptyset$. This contrasts with the Jackson network, for which induced chains are again Jackson networks on a smaller set of nodes.

All measures associated with ξ_t^Λ have superscript Λ. The main one of these is the Λ-route $r^\Lambda(i, s)$ of type i: it is the part of type is whole route $r(i, \cdot)$ between two successive visits to Λ-queues or the outside world, containing i itself.

2 Λ-load equations and associated graphs

Assume now that the induced chain ξ_t^Λ is ergodic. Denote by ρ_i^Λ the stationary probability that type $i \in \bar{\Lambda} \cup \mathfrak{p}(\Lambda)$ is the highest priority type present in its node. Note for $i \in \mathfrak{p}(\Lambda)$ that ρ_i^Λ is the stationary probability that node $n(i)$ is empty in the induced chain.

The interpretation of type i *load* is therefore only valid for $i \in \bar{\Lambda}$. Indeed, for $i \in \mathfrak{p}(\Lambda)$ the stationary probability ρ_i^Λ is not equal to the load of type i offered per unit time.

By conservation of flow, the ρ_i^Λ are a solution of

$$\rho_i^\Lambda \mu_i = \mathbf{1}_{\{a(i)=0\}} \lambda_i + \sum_j \mathbf{1}_{\substack{\{a(i)=j: \\ j \in \bar{\Lambda} \cup \mathfrak{p}(\Lambda)\}}} \rho_j^\Lambda \mu_j, \quad i \in \bar{\Lambda}. \tag{2.1}$$

This equation has a $|\mathfrak{p}(\Lambda)|$-dimensional solution space, which for $i \in \bar{\Lambda}$ can be expressed by

$$\rho_i^\Lambda = \begin{cases} \rho_i, & r^\Lambda(i,1) = r(i,1) \notin \Lambda \\ 0, & r^\Lambda(i,1) \in \Lambda \setminus \mathfrak{p}(\Lambda) \\ \frac{1}{\mu_i} \rho_{r^\Lambda(i,1)}^\Lambda \mu_{r^\Lambda(i,1)}, & r^\Lambda(i,1) \in \mathfrak{p}(\Lambda). \end{cases} \tag{2.2}$$

To get the unique solution in the ergodic case, we have to require

$$\sum_{j\succeq i} \rho_j^\Lambda = 1, \quad \text{for any} \quad i \in \mathfrak{p}(\Lambda). \tag{2.3}$$

We will call (2.1) and (2.3) the Λ-*load equations.* A solution ρ_i^Λ, $i \in \bar{\Lambda} \cup \mathfrak{p}(\Lambda)$, to the Λ-load equations is said to be a *feasible* solution it is strictly positive and

$$\sum_{j\in\bar{\Lambda}\cap\mathcal{J}(n)} \rho_j^\Lambda < 1, \quad \text{for any node } n. \tag{2.4}$$

The latter is an extra condition for nodes for which $\bar{\Lambda}$ contains *all* types served at the node.

As is the case for the overall network, although feasibility is necessary in order to have stability (ergodicity), it does not imply it.

The following algorithm is useful for simplifying the Λ-load equations.

Reduction algorithm

Step 1. Plug in (2.2) for $i \in \bar{\Lambda}$. For any $i \in \bar{\Lambda}$ check whether $r^\Lambda(i,1) = r(i,1)$. If so, set $\rho_i^\Lambda = \rho_i = \lambda_{r(i,1)}\mu_{r(i,1)}/\mu_i$ and set $\Lambda_0 := \Lambda \cup \{i\}$. Set $\Lambda_1 := \mathfrak{p}(\Lambda)$.

Step 2. Choose any $i \in \Lambda_1$. If $j \in \Lambda_0$ for all $j \succ i$ then do the following:

- **i)** Set $\rho_i^\Lambda = 1 - \sum_{j\succ i} \rho_j^\Lambda$;
- **ii)** for $k = r^\Lambda(i,s) \in \bar{\Lambda}_0$, for some $1 < s \leq s_i^\Lambda$, set $\rho_k^\Lambda = \rho_i^\Lambda \mu_i/\mu_k$ and put $\Lambda_0 := \Lambda_0 \cup \{k\}$;
- **iii)** put $\Lambda_1 = \Lambda_1 \setminus \{i\}$.

Repeat Step 2 till there are no such i to be found anymore. Then stop.

We will show the existence of a cycle in the graph $\mathcal{G}(\Lambda)$ defined below, whenever the resulting set Λ_1 is non-empty.

Construction of the graph $\mathcal{G}(\Lambda)$.

Given Λ we can easily construct the following directed graph $\mathcal{G}(\Lambda)$ (we write $\mathcal{G}$ whenever $\Lambda = \emptyset$): it has the set of vertices $V(\Lambda) = \bar{\Lambda} \cup \mathfrak{p}(\Lambda)$.

Consider the Λ-route $r^\Lambda(i,\cdot)$ of type $i \in V(\Lambda)$ and suppose that i itself is the l-th stage of the route, i.e. $i = r^\Lambda(i,l)$. Then we draw an arrow $i \to j$ for $j \in V(\Lambda)$, whenever $j \succ r^\Lambda(i,s)$ for some $s < l$. In words: there is an arrow $i \to j$ when in the induced chain ξ_t^Λ j has priority over some antecedent of i.

Whenever $\Lambda \subset \Lambda'$, we have that the graph $\mathcal{G}(\Lambda')$ is a subgraph of $\mathcal{G}(\Lambda)$, but not the restriction of $\mathcal{G}(\Lambda)$ to the vertices of $V(\Lambda')$! It is possible to define an extended construction where this is the case indeed, but this is not necessary for our purposes. Note that this construction is based on ideas from [1] for detecting essential states.

The next lemma summarises the most general conclusions we can make so far.

Lemma 2.1 **i)** *If Λ_1 is empty, then the Λ-load equations have a unique solution.*

ii) *If Λ_1 is non-empty, then there is a cycle in the graph $\mathcal{G}(\Lambda')$ where*

$$\overline{\Lambda'} = \{i \mid \text{there exists } j \in \Lambda_1, \text{ such that } i \succ j\}.$$

iii) *If the Λ-load equations have a non-unique feasible solution, then the induced chain ξ_t^Λ is non-ergodic.*

It is precisely the graph $\mathcal{G}$, that will play a role in detecting possible occurrence of instability even when the loads of the different nodes are smaller than 1.

3 Second vectorfield, Euler paths and main result

Using the solutions to the Λ-load equations we can construct a vectorfield v on $\mathbf{R}_+^p$ (cf. [2]).

First, suppose that ξ_t^Λ is ergodic. Then Birkhoff's ergodic theorem implies for $x \in B^\Lambda$, that

$$\frac{1}{N}\xi_{[tN]}([xN]) \to x + tv(\Lambda),$$

where

$$v_i(\Lambda) = \begin{cases} 0, & i \in \bar{\Lambda} \\ \rho^\Lambda_{r^\Lambda(i,1)}\mu_{r^\Lambda(i,1)} - \mathbf{1}_{\{i \in \mathfrak{p}(\Lambda)\}}\rho_i\mu_i, & i \in \Lambda. \end{cases}$$

If ξ_t^Λ is non-ergodic, then there can be two possibilities. If an induced chain $\xi_t^{\Lambda'}$ for $\Lambda' \supset \Lambda$ is ergodic, and $v_i(\Lambda') > 0$ for $i \in \Lambda' \setminus \Lambda$, we set $v(\Lambda) = v(\Lambda')$ (note that $v(\Lambda)$ can be multi-valued and this really occurs). If such Λ' do not exist, then we set $v(\Lambda) \equiv 0$.

The problem here is to detect ergodic faces. One can use an iterative procedure described in [2].

The second vectorfield v is now defined as $v(x) = v(\Lambda)$, whenever $x \in B^\Lambda$.

With this field, one can associate paths.We use the definition from [2]. For any initial point $x \in \mathbf{R}_+^p$, an *Euler path* $\Gamma_x(t) : [0, \infty) \to \mathbf{R}_+^p$, of the priority network under consideration is a continuous mapping with the following properties:

i) $\Gamma_x(0) = x$;

ii) $\Gamma_x(\cdot)$ belongs to the union of ergodic faces except for a countable subset $\mathcal{T}(\Gamma_x)$ of $[0, \infty)$;

iii) for the points of the same interval belonging to $[0, \infty)\setminus\mathcal{T}(\Gamma_x)$, Γ_x is linear with velocity $d\Gamma_x/dt = v(\Lambda)$ when Γ_x runs through the ergodic face B^Λ.

We will be concerned with the occurrence of cyclic Euler paths. To this end we will use the following definition.

Definition 3.1 *An Euler path Γ_x is called acyclic if on any finite time interval it visits finitely many faces, otherwise it is called cyclic. The random walk is called acyclic if all Euler paths are acyclic. It is called cyclic if some Euler path is cyclic.*

We can finally state the desired characterisation.

Theorem 3.1 **i)** *If the priority network is acyclic, then it is stable (ergodic) whenever the load of each node is smaller than 1, i.e. (2.4) holds for $\Lambda = \emptyset$.*

ii) *If $\mathcal{G}$ is acyclic then the priority network is acyclic for any choice of service parameters.*

iii) *If $\mathcal{G}$ is cyclic, then there are service parameters for which the network is not acyclic.*

References

[1] V. Dumas (1996), Essential faces and stability conditions of multiclass networks with priorities. Technical report 3030, INRIA. *To appear in: Markov Proc. Relat. Fields.*

[2] G. Fayolle, V.A. Malyshev, and M.V. Menshikov (1994), *Constructive Theory of countable Markov Chains.* Cambridge University Press, Cambridge.

[3] F.M. Spieksma. On cycles in exponential priority queues with deterministic routing. *In Preparation.*