Queueing-Network Based Applications Under Worst-case Attacks

Jhonatan Tavori
Blavatnik School of Computer Science
Tel-Aviv University
jhonatant@mail.tau.ac.il

Hanoch Levy
Blavatnik School of Computer Science
Tel-Aviv University
hanoch@tauex.tau.ac.il

1. INTRODUCTION

A variety of today's critical applications are based on queueing networks whose performance, mainly delay, depends on routing and resource allocation. These include computer networks (the Internet), load-balancers on cloud systems and vehicular traffic networks.

These applications are vulnerable to malicious attacks which may include the dis-functioning of the network components by malwares, or other computer viruses. For computer networks there are many references (e.g., [1, 5]); for vehicular traffic networks see navigation platforms attacks (e.g., *Waze* [4]).

Such attacks will increase the delay experienced in the network and degrade its performance. We aim at understanding what are the weak-points of such networks. That is, how can a sophisticated attacker cause the maximal damage to the network using minimal attacking power. Furthermore, we are interested in analyzing how flexible is the network in reacting to such attacks by re-routing its traffic, and whether such flexibility grants significant protection.

An intriguing question, which we aim at addressing and which may affect network planning, is what will be the nature of an optimal attack: will it be concentrated at few nodes of the network, or would it be scattered over many regions (nodes).

Prior works (e.g., [1, 3, 5]) that dealt with such worst-case attacks on distributed systems did not address queueing delays and their effects.

To allow analytic treatment that will reveal the nature of these networks, we consider here a simplistic queueing model based on k queues which captures the main features of these networks: (i) Multi-commodity arrival flows, (ii) The ability of the network to migrate requests (users) from one route to another. Treatment of general structure networks is the subject of ongoing research which is based on this work.

Our analysis reveals somewhat surprising properties: The nature of an optimal attack varies as a function of the system parameters, and may shift from fully concentrated to fully scattered in the extreme cases. This is in contrast to the (no queueing) model and results of [5] which asserted that optimal attacks are concentrated, even when the system can defend itself using requests migration. This suggests that queueing mechanisms and the consideration of queueing delays may cause scattering of optimal attacks.

2. MODEL AND APPLICATIONS

We consider networks consisting of k M/M/1 queues which we denote *regions* (modeling k distributed data-centers, or k different highways), with capacities of $(c_1, \ldots, c_k)$. We assume that the flow in the network is multi-commodity. I.e., it consists of k different Poisson flows, $(\lambda_1, \ldots, \lambda_k)$, where λ_i is the arrival rate (of requests) to queue i. See Fig. 1(a).

Under this setting, we have in mind several systems at risk both in the cyber and physical domains, including: (i) Real-time distributed cloud services whose data-centers (resources) are distributed over k regions. Each region has a local demand within the region, and a capacity of resources. (ii) A simplistic model of vehicular traffic networks, where the system routes arriving vehicles through alternative roads. (iii) A simplistic model of data networks.

The expected time in system (delay) spent by an arbitrary request (representing a packet, vehicle, etc.) is given by:

$$T = \sum_{i=1}^{k} \frac{\lambda_i}{\Lambda} \cdot T_{\mu,c_i,\lambda_i} \qquad (1)$$

where $\Lambda := \sum_{i=1}^{k} \lambda_i$, and $T_{\mu,c,\lambda}$ is the sojourn time at an M/M/1 queue with capacity c and mean job length $1/\mu$, and is given by $T_{\mu,c,\lambda} := \frac{1}{\mu \cdot c - \lambda}$ [2].

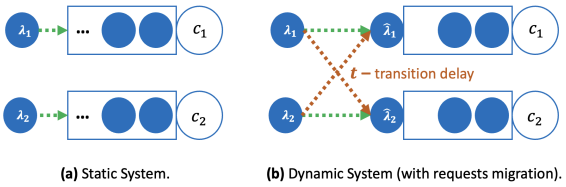

Figure 1: Multi-commodity 2-regions system.

Traffic Migration. Many real-world systems may balance delays using requests migration (e.g., providing service to users located in region i using the data-centers in region j, or by navigating drivers to a farther yet empty highway). Requests which are transferred between regions may possibly incur an additional *transition delay* which we assume to be constant and denote by t. This is in addition to experiencing the queue sojourn time expressed by $T_{\mu,c,\lambda}$.

Denote by $\hat{\lambda} = (\hat{\lambda_1}, \ldots, \hat{\lambda_k})$ the flows resulting from this migration. For example, if region i experiences an increased delay, part of that region's arriving requests, say δ, can be shifted to some other region, j, yielding $\hat{\lambda_i} = \lambda_i - \delta$ and

$\hat{\lambda_j} = \lambda_j + \delta$. See Fig. 1(b).

Given the original arrival rates λ, and the flow rates resulting from migration $\hat{\lambda}$, the expected in system delay is[1]:

$$\sum_{i=1}^{k} \frac{\hat{\lambda}_i}{\Lambda} \cdot T_{\mu,c_i,\hat{\lambda}_i} + \sum_{i=1}^{k} \frac{t}{2} \cdot |\hat{\lambda_i} - \lambda_i|. \quad (2)$$

Attacking The System. An attacker might disrupt or crash the system resources using, for example, malicious worms or by physical destruction. In our model this will result in reducing the capacities (c_i values) of the queues.

Let x_i be the attack volume in region i. The resulted capacity in region i as a result of the attack is: $c_i - x_i$. The attack vector (or simply, the *attack*) is $\mathbf{X} = (x_1, \ldots, x_k)$, where $|\mathbf{X}| \leq x$ is the attack size constraint (constrained by the attacker's resources). The expected system delay, as a result of the attack $\mathbf{X}$ is:

$$T(\mathbf{X}) = \sum_{i=1}^{k} \frac{\lambda_i}{\Lambda} \cdot T_{\mu,c_i-x_i,\lambda_i} = \sum_{i=1}^{k} \left[\frac{\lambda_i}{\Lambda} \cdot \frac{1}{\mu \cdot (c_i - x_i) - \lambda_i} \right]. \quad (3)$$

Throughout this work we assume that the size of the attack maintains that the arrival rate at any queue will not exceed the service rate ($\lambda \leq \mu \cdot c$). This is based on assuming that the attacker is rational and since $\lambda = \mu \cdot c$ yields already infinite delay, $\lambda > \mu \cdot c$ will waist the attacker's resources.

3. STATIC SYSTEMS

We consider static systems whereby the system cannot react to the attack by re-routing requests. We show that the optimal attack is concentrated. Namely, the attacker will pick some regions and attack them entirely, up to the constraint of $\lambda_i \leq \mu \cdot (c_i - x_i)$.

The attacker aims to maximize the expected system delay, by controlling $\mathbf{X}$. I.e., its objective function is: $\max_{\|\mathbf{X}\| \leq x} T(\mathbf{X})$ (Eq. (3)). Denote by $f_i(x_i) := \frac{\lambda_i}{\Lambda} \cdot T_{\mu,c_i-x_i,\lambda_i}$. Note that the second derivative of f_i with respect to x_i is

$$\frac{\partial^2 f_i}{\partial x_i^2} = -\frac{2 \cdot \lambda_i \cdot \mu^2}{\Lambda \cdot (\lambda_i - \mu \cdot (c_i - x_i))^3} \quad (4)$$

which is negative for any x_i such that $\lambda_i < \mu \cdot (c_i - x_i)$. Therefore, the delay in each region is *concave* (and monotonically *increasing*) in the attack volume, x_i. That is, the marginal benefit (to the attacker) increases with the attack size x_i. Thus, if the attacker chooses to attack at some volume x_i in some region i, it is optimal to continue investing attacking efforts in that region (i.e., increasing x_i up to the limit of $\lambda_i \leq \mu \cdot (c_i - x_i)$) [2].

4. DYNAMIC SYSTEMS

We analyze dynamic systems whereby the system may reduce the attack damage by migrating requests, and study the effect of the transition delay on the optimal attack policy. Denote by $T(\mathbf{X}, \hat{\lambda})$ the expected system delay as a result of the attack $\mathbf{X}$ and the requests migration $\hat{\lambda}$:

$$T(\mathbf{X}, \hat{\lambda}) := \sum_{i=1}^{k} \frac{\hat{\lambda}_i}{\Lambda} \cdot T_{\mu,c_i-x_i,\hat{\lambda}_i} + \sum_{i=1}^{k} \frac{t}{2} \cdot |\hat{\lambda_i} - \lambda_i| \quad (5)$$

The objective of the attacker is to find an attack which will maximize the expected delay, under the assumption that the system will defend optimally: $\max_{\|\mathbf{X}\| \leq x} \min_{\hat{\lambda}} T(\mathbf{X}, \hat{\lambda})$.

We first provide precise analysis of two extreme special cases: (i) where the transition cost is infinity, and (ii) where it is 0. We then address general t values and propose an algorithm which derives the optimal traffic (requests) migration given an attack, and use it to numerically evaluate the system performance under various attack strategies.

4.1 Special Case: $t = \infty$

As t is extremely high ($t = \infty$), the system would never prefer to migrate requests since the transition delay makes it not worthy. Thus, the system is practically perfectly static and the optimal attack policy coincides with that of Sec. 3.

4.2 Special Case: $t = 0$

We now move to analyze the opposite case, where the network is perfectly dynamic (i.e., $t = 0$), that is, traffic can be transferred from one region to another at no cost. We show that in contrast to the static case, in this case the optimal attack tends to be scattered over many regions.

Note that while the network is perfectly dynamic, the migration has to maintain the constraint that $\sum \hat{\lambda} = \sum \lambda = \Lambda$. Thus, we derive the optimal $\hat{\lambda}$ for a given system parameters (μ, c, λ) and attack $\mathbf{X}$ using the method of Lagrange multipliers. The Lagrangian is: $G = T(\mathbf{X}, \hat{\lambda}) + \beta \cdot \left[\sum_{i=1}^{k} \hat{\lambda}_i - \Lambda \right]$.

We are interested in solving $\frac{\partial G}{\partial \hat{\lambda}_i} = 0$ for $i = 1, \ldots k$. Differentiation yields $0 = \frac{\mu \cdot (c_i - x_i)}{\Lambda \cdot (\mu \cdot (c_i - x_i) - \hat{\lambda}_i)^2} + \beta$, or

$$\hat{\lambda}_i = \mu \cdot (c_i - x_i) \pm \sqrt{-\frac{\mu \cdot (c_i - x_i)}{\Lambda \cdot \beta}}. \quad (6)$$

Note that since we require that $\lambda_i < \mu \cdot (c_i - x_i)$, the $\pm$ has to be $-$. By summing over the last equation we have: $\sum_{i=1}^{k} \hat{\lambda}_i = \sum_{i=1}^{k} \mu \cdot (c_i - x_i) - \frac{1}{\sqrt{\beta}} \sum_{i=1}^{k} \sqrt{\mu \cdot (c_i - x_i)/\Lambda}$. The left-hand side just equals to Λ; thus

$$\frac{1}{\sqrt{\beta}} = \frac{\sum_{i=1}^{k} \mu \cdot (c_i - x_i) - \Lambda}{\sum_{i=1}^{k} \sqrt{\mu \cdot (c_i - x_i)/\Lambda}} \quad (7)$$

Substituting Eq. (7) in Eq. (6) yields: $\hat{\lambda}_i = \mu \cdot (c_i - x_i) - \sqrt{\frac{\mu \cdot (c_i - x_i)}{\Lambda}} \left(\frac{\sum_{i=1}^{k} \mu \cdot (c_i - x_i) - \Lambda}{\sum_{i=1}^{k} \sqrt{\frac{\mu \cdot (c_i - x_i)}{\Lambda}}} \right)$. Substitution of $\hat{\lambda}_i$ into the delay formula followed by some algebraic manipulation yields:

$$\min_{\hat{\lambda}} T(\mathbf{X}, \hat{\lambda}) = \frac{1}{\Lambda} \cdot \left(\frac{\left(\sum_{i=1}^{k} \sqrt{\mu \cdot (c_i - x_i)} \right)^2}{\sum_{i=1}^{k} \mu \cdot (c_i - x_i) - \Lambda} - k \right). \quad (8)$$

To derive the attacker strategy we must maximize Eq. (8) over all valid $|\mathbf{X}| \leq x$. To this end, note that Eq. (8) is monotone increasing in any x_i. Thus the attacker will operate *on the constraint* (namely $|\mathbf{X}| = x$). Since the denominator is constant over all such attacks, the decisive expression is $(\sum_{i=1}^{k} \sqrt{\mu \cdot (c_i - x_i)})^2$ from the numerator. That is,

[1]Note that we assume that we do not migrate twice (i.e., migrations from i to j and then from j to i are not allowed).

[2]A full proof of the optimality of concentrating efforts on a small number of regions, where the objective function is a sum of concave functions, can be found in [5], along with algorithms to derive such an optimal attack.

$\sum_{i=1}^{k} \sqrt{c_i - x_i}$ needs to be maximized under the constraints $\sum x_i = x$, and $0 \leq x_i \leq c_i$. The optimal solution strives to equate $c_i - x_i$ across all regions. It is achieved by finding the maximal $0 \leq n \leq k$ for which the n highest capacity queues hold $x_i = c_i - {}^{(\sum_{\text{highest}} c_i - x)}/_{n} \geq 0$. As a special case, on a symmetric system we obtain $x_i = x_j$ for all i, j. Hence, the optimal attack policy might (and tends to) be scattered over many regions, as opposed concentrated.

4.3 Algorithmic Approach for $0 < t < \infty$

Recall that the objective function of the attacker is a *max-min* problem (with constraints). Theoretically, it can be solved numerically using optimization programs, but at a potentially high computation cost, as each outer *max* solution (attack $\mathbf{X}$) requires an evaluation of the inner *min* problem ($\hat{\lambda}$ values). We propose an algorithm which solves the inner *min* problem (i.e., derives the optimal $\hat{\lambda}$). It can be used to speed up the overall problem solution process.

We derive the optimal $\hat{\lambda}$ in a greedy manner while migrating the requests step-by-step. The algorithm proceeds as follows: It begins with the original flow of the network (the λ values), and performs "discrete" steps of size δ.[3] At each step, it calculates the values of the marginal revenue (loss) from migrating δ from (to) region i:

$$\Delta_j^-(\hat{\lambda_j}, \delta) = T_{\mu,c_i,\hat{\lambda_i}} - T_{\mu,c_i,\hat{\lambda_i}-\delta}, \quad (9)$$

$$\Delta_j^+(\hat{\lambda_j}, \delta) = T_{\mu,c_i,\hat{\lambda_i}+\delta} - T_{\mu,c_i,\hat{\lambda_i}}. \quad (10)$$

Using the Δ values, the migration with the maximal value is picked, and so on iteratively. The algorithm stops when the next migration step is not worthy (due to the transition delay). See the pseudo code described in Algorithm 1.

Algorithm 1 Pseudo-code of Optimal Requests Migration

Set $(\hat{\lambda_1}, \ldots, \hat{\lambda_k}) \leftarrow (\lambda_1, \ldots, \lambda_k)$.
while True **do**
  Let j s.t. $\Delta_j^-(\hat{\lambda_j}, \delta) = \max_i \Delta_i^-(\hat{\lambda_i}, \delta)$.
  Let k s.t. $\Delta_k^-(\hat{\lambda_k}, \delta) = \min_i \Delta_i^+(\hat{\lambda_i}, \delta)$.
  if $\Delta_j^-(\hat{\lambda_j}, \delta) - \Delta_k^+(\hat{\lambda_k}, \delta) > t$ **then**
    $\hat{\lambda}_j = \hat{\lambda}_j - \delta$ and $\hat{\lambda}_k = \hat{\lambda}_k + \delta$.
  else
    Return $\hat{\lambda} = (\hat{\lambda_1}, \ldots, \hat{\lambda_k})$.

The optimality of the algorithm results from two properties of the expected delay in the system: (i) It is a separable function, as it is the sum of the ${}^{\hat{\lambda_i}}/_{\Delta} \cdot T_{\mu,c_i,\hat{\lambda_i}}$ expressions; (ii) ${}^{\hat{\lambda}_i}/_{\Lambda} \cdot T_{\mu,c_i,\hat{\lambda}_i}$ is convex in $\hat{\lambda}_i$ (as its second derivative is positive for any $\hat{\lambda}_i < \mu \cdot c_i$). Thus, Δ_j^- is monotonically decreasing in $\hat{\lambda}_i$ and Δ_j^+ is monotonically increasing in $\hat{\lambda}_i$. Hence, the migrations chosen by the algorithm are non-regrettable as the marginal benefit from each migration throughout the process is monotonically non-increasing.

The running time of the algorithm is bounded by $O(k + \frac{1}{\delta} \cdot \Lambda)$ (maximal number of migrations of size δ).

We use Algorithm 1 to numerically demonstrate the effect of various attack strategies on the expected system delay, depending on the value of the transition delay, t. Figure 2 plots the expected system delay on a 2-regions symmetric system as a function of the attack vector, and as a function of t. For example, at $t = 0.02$, the optimal attack vector is $(0.7 \cdot x, 0.3 \cdot x)$ (or $(0.3 \cdot x, 0.7 \cdot x)$, as the system is symmetric).

As can be seen, the concentrated attack (blue) is superior at high values of t, and the superiority, comparing to the other attacks, increases in t. On the other hand, the balanced attack (yellow) is superior at low values of t and the superiority decreases in t. This is consistent with our results from Sec. 4.1 and 4.2, which derived the optimal attack strategies for the special cases of $t = 0$, where the optimal strategy was to scatter the attack, and of $t = \infty$ where the optimal strategy was to concentrate the attack.

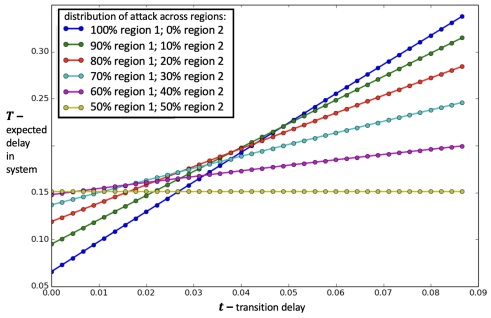

Figure 2: The system delay, as a function of the attack vector and t, on a 2-regions symmetric system.

5. CONCLUSIONS AND FURTHER WORK

We used a model which captures some fundamental properties of real-world queueing networks, and analyzed and evaluated optimal attack policies for static and dynamic systems. Our results reveal that optimal attack nature may range from full concentration to full scattering, depending on the ability to re-route traffic, and on the transition cost.

In an ongoing work we extend the model and account for (i) requests rejection combined with queueing delay, and inspect whether this affects the optimal attack policies; (ii) attacks which target the system with a load of "fake-requests" (i.e., artificially increase λ_i values); We work on developing an algorithm to derive an optimal attack for any t, towards analyzing arbitrary flow networks as were described in [2].

Acknowledgment

This research was supported in part by the Israel Science Foundation (grant No. 2482/21) and by the Blavatnik Family Foundation.

6. REFERENCES

[1] U. Ben-Porat. *Vulnerability of Network Mechanisms to Sophisticated DDoS attacks*. PhD dissertation, 2014.

[2] L. Kleinrock. Queueing systems, volume 2: Computer applications, 1976.

[3] H. Levy and J. Tavori. Worst case attacks on distributed resources systems. *ACM SIGMETRICS Performance Evaluation Review*, 48(2):9–11, 2020.

[4] M. B. Sinai, N. Partush, S. Yadid, and E. Yahav. Exploiting social navigation. *arXiv:1410.0151*, 2014.

[5] J. Tavori and H. Levy. Tornadoes in the cloud: Worst-case attacks on distributed resources systems. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2021.

[3] As δ decreases, the accuracy of the algorithm increases.