Wednesday, June 8

08:15 -- 08:20 am  Opening Remarks

08:20 -- 08:50 am  Path-vector routing stability analysis
                  D. Papadimitriou (Alcatel-Lucent Bell), Florin Coras, A. Cabellos
                  (Technical University of Catalonia)

08:50 -- 09:20 am  A deterministic algorithm of single failed node recovery in MSR-
                  based distributed storage systems
                  H. Zhao, Y. Xu (University of Science & Technology of China)

09:20 -- 09:50 am  Robust heterogeneous data center design: A principled approach
                  S. Garg, S. Sundaram, H.D. Patel (University of Waterloo)

09:50 -- 10:15 am  Coffee Break

10:15 -- 10:45 am  On network criticality in wireless networks
                  A. Tizghadam, A. Leon-Garcia (University of Toronto)

10:45 -- 11:15 am  Diffusion and cascading behavior in random networks
                  M. Lelarge (INRIA-ENS)

11:30 -- 12:30 pm  FCRC Plenary Talk

12:30 -- 01:30 pm  Lunch

01:30 -- 02:00 pm  Search in non-homogeneous random environments
                  O.H. Abdelrahman, E. Gelenbe (Imperial College)

02:00 -- 02:30 pm  On estimation problems for the G/G/Infinity queue
                  H. Feng, P. Dube, L. Zhang (IBM T.J. Watson Research Center)

02:30 -- 03:00 pm  Dispatching to incentivize fast service in multi-server queues
                  S. Doroudi (CMU), R. Gopalakrishnan, A. Wierman (California Institute of
                  Technology)

03:00 -- 03:30 pm  The power of partial power of two choices
                  O.T. Akgun, R. Righter, R. Wolff (University of California, Berkeley)

03:30 -- 04:00 pm  Coffee Break

04:00 -- 04:30 pm  Settling for less -- A QoS compromise mechanism for opportunistic
                  mobile networks
                  R. Pai (USC), S. Kosta (University of Rome, Sapienza), P. Hui (Telekom
                  Labs)

04:30 -- 05:00 pm  Investigating the effect of node heterogeneity and network
                  externality on security adoption
                  Z. Yang, J.C.S. Lui (The Chinese University of Hong Kong)

05:00 -- 05:30 pm  Implications of peer selection strategies by publishers on the
                  performance of P2P swarming systems
                  D. Sadoc Menasche (University of Massachusetts), A.A. de A. Rocha
                  (Fluminense Federal University), E.A. de Souza e Silva (Federal
                  University of Rio de Janeiro), D. Towsley (University of Massachusetts),
                  R.M. Meri Leao (Federal University of Rio de Janeiro)

05:30 -- 06:00 pm  How impatience affects the performance and scalability of P2P video
                  on-demand systems
                  S. Aalto, P. Lassila (Aalto University), P. Savolainen, S. Tarkoma (Helsinki
                  Institute for Information Technology)
In this paper, we define a set of metrics that characterize the local stability properties of path-vector routing protocols such as BGP (Border Gateway Protocol). By means of these stability metrics, we propose a method to analyze the effects of BGP policy- and protocol-induced instability on local routers.

1. **INTRODUCTION**

Research efforts to understand BGP's instability led to categorize them into policy- and protocol-induced instabilities.

**Policy-induced instabilities**: addressing routing stability consistently with planned BGP routing policy implies eliminating non-deterministic routing states resulting from policy interactions and in particular, non-deterministic and unintended but unstable states. Griffin et al.'s seminal work [1] modeled BGP as a distributed algorithm for solving the Stable Paths Problem, and derived a general sufficient condition for BGP stability, known as "No Dispute Wheel". This sufficient condition guarantees the existence of a stable solution to which BGP always converges. Informally, this sufficient condition allows nodes to have more expressive and realistic preferences than always preferring shorter routes to longer ones. The game theoretic approach introduced in [2] relies on the best-reply BGP dynamics: a convergence game model in which each Autonomous System (AS) is instructed to continuously execute the following actions: i) receive update messages from BGP peering nodes announcing their routes to the destination, ii) choose a single peering node whose route is most preferred to send traffic to, iii) announce the new route to peering nodes. However, as proved in [2], best-reply BGP dynamics is not incentive-compatible even if No Dispute Wheel condition holds: even if all but one AS are following the BGP rules, the remaining AS may not have the incentive to follow them. Interestingly, as demonstrated in [2], incentive-compatible BGP dynamics requires combining an additional global condition (Route Verification) together with the "No Dispute Wheels" to guarantee stability. Consequently, all known conditions for global stability are sufficient but not necessary conditions (detecting them is an NP-hard problem and enforcing them requires a global deployment of an additional mechanism); on the other hand, local instability effects have yet to be characterized.

**Protocol-induced instabilities**: BGP is an inter-AS path-vector routing protocol subject to Path Selection Algorithm which is similar to any other path-vector algorithm: BGP routers may announce as valid, routes that are affected by a topological change and that will be withdrawn shortly after subsequent routing updates. This phenomenon is the main reasons for the large number of routing updates received by BGP routers which exacerbrate inter-domain routing system instability and processing overhead [3]. Both result in delaying BGP convergence time upon topology change/failure [4]. Several mitigation mechanisms exist to partially limit the effects of path exploration; however, none actually eliminate its effects. Hence, BGP is intrinsically subject to instability.

The goals of this paper are to 1) Develop a methodology to process and interpret the data part of BGP routing information bases in order to identify and document occurrences of Internet BGP routing stability phenomena; 2) Identify a set of stability metrics and develop methods for using them in order to provide a better understanding of the BGP routing system’s stability; and 3) Investigate how path-vector routing protocol behavior and network dynamics mutually influence each other. The proposed approach aims to bring rigor and consistency to the study of routing stability. For example, it would allow for a unified approach to the cross-validation of techniques for looking at improving path exploration effects on the routing system.
input events stop. A routing system, which remains in an unending condition of transition from one state to another when disturbed by an external or internal event, is considered to be unstable. More precisely, let $|\text{ART}(t+1)|$ be the magnitude of the change to the routing table (RT) at some time $t+1$.

**Definition 2:** A RT, which returns to its initial equilibrium state, when disturbed by an external and/or internal event, is considered to be **stable** when the following condition is met: $|\text{ART}(t+1)| \leq \alpha, t \to \infty$, where $\alpha > 0$ is small.

**Definition 3:** A RT, which transitions to a new equilibrium state, when disturbed by an external and/or internal event, is considered to be **marginally stable** when the following condition is met $|\text{ART}(t+1)| \leq \beta, t \to \infty$, where $\beta > 0$ is small, $\alpha < \beta$.

**Definition 4:** A RT, which remains in an unending condition of transition from one state to another when disturbed by an external or internal event, is considered to be **unstable** when the following condition is met: $|\text{ART}(t+1)| > \beta, t \to \infty$.

The values $\alpha$ and $\beta$ shall be set based on operational criteria. Among other things, $\alpha$ and $\beta$ depend on the observation sampling period that needs to be the MRAI timer. This ensures one routing update per sampling period. A similar reasoning to the one applied for the Loc_RIB stability (that corresponds to the BGP routing table) can be applied to the Adj_RIB_In (which stores incoming routes from neighbors). It is also interesting to measure the instability induced by the BGP selection process.

### 2.3 Stability Metrics

To measure the degree of stability of the Loc_RIB, Adj_RIB_In, and determine how close the system is to being unstable the following stability metrics are defined:

- **Stability $\phi_i$** of selected routes $r_i$: characterizes the stability of the selected routes stored in the Loc_RIB (|Loc_RIB| = $N$) by quantifying the magnitude of change on these routes form the selected route at time $t$. The computation of $|\text{ART}(t+1)|$ can then be derived from the stability of individual routes.

When route $r_i$ is created: $\phi_i(t) \leftarrow 0$.

When $r_i$ experiences a path or attribute change

- if $r_i(t+1) \neq r_i(t)$ then $\phi_i(t+1) \leftarrow \phi_i(t) + 1$
- else if $\phi_i(t) = 0$ then $\phi_i(t+1) \leftarrow 1$
- else if $\phi_i(t) > 0$ then $\phi_i(t+1) \leftarrow \phi_i(t) - 1$

end if
end if

With this processing of the stability metric for an individual route, one can compute the stability metric for an entire routing table (RT). Let $|\text{ART}(t+1)|$ denote the change in stability metric associated with a single route $r_i$, from time $t$ to $t+1$. These values are used to compute $|\text{ART}(t+1)|$, the change in stability metric for the entire routing table from time $t$ to $t+1$. Moreover, $|\text{ART}(t+1)|$ is normalized so that $0 \leq |\text{ART}(t+1)| \leq 1$, where 0 implies perfect stability, and 1 indicates complete instability.

#### For $i=1$ to $N$ /* total nbr of routes in RT(t+1) */

- if $r_i(t+1)$ is a new route then $\Delta_r(t+1) \leftarrow 0$
- else if $\phi_i(t) = 0$ and $\phi_i(t+1) = 0$ then $\Delta_r(t+1) \leftarrow 0$
- else if $\phi_i(t) > 0$ then $\Delta_r(t+1) \leftarrow \phi_i(t+1) - \phi_i(t)$

end if
end if

Most stable route in the Adj_RIB_In (|Adj_RIB_In| = $M$): quantifies the relative stability between the routes to the same destination $d$, learned from different upstream BGP peers. Let $\mathcal{W}_d$ denote the set of node's $u$ BGP peers, $|\mathcal{W}_d| = W \leq M$. At node $u$, the most stable route to destination $d$ at time $t$: $r'_u(d), \text{stable}(u, W, d) \in \mathcal{P}_u$, where $\mathcal{P}_u$ is the set of all paths in the BGP routing table. This metric provides a measure of the relative measure of route's $\delta_r(t)$ stability with respect to the most stable route for the same destination $d$, $\phi_{stable}$.

Best selectable route in the Adj_RIB_In: quantifies the relative stability between routes to the same destination $d$ as learned from all upstream peers and the one amongst them selected by BGP at time $t$ as the best route (thus following the BGP route selection). The computational procedure is the same as the previous one if one replaces $\phi_{stable}$ by $\phi_{stable}$.

**Difference**

- **Stability** between the most stable route in the Adj_RIB_In and the selected route stored in the Local_RIB for the same destination $d$: characterizes the stability of the currently selected routes against most stable routes as learned from upstream neighbors. This metric provides a measure of the stability of the learned routes compared to the stability of the currently selected route. A variant of this metric, denoted $\delta_r(u, V, d)$, characterizes the stability of the selected path $p(u,v)$ at time $t$ for destination $d$ against the stability of the path $p'(u,v)$ as selected at time $t$ following the BGP selection rules and that would replace $p(u,v)$ at time $t+1$: $\delta_r(t) = p(u,v) - p'(u,v)$. In turn, if $\delta_r(t) > 0$, then the replacement of $r(t)$ by $p'(t)$ increases stability of the route to destination $d$; otherwise, the safest decision is to keep the currently selected route $r(t)$.

Application during the BGP selection process of the metric $\delta_r$ would prevent replacement of more stable routes by less stable one but also enable selection of more stable routes than the currently selected routes. However, for this assumption to hold we must prove the consistency of the stability-based selection with the preferential-based selection model that relies on path ranking function. For each $u \in V$, there is a non-negative, integer-value ranking function $\lambda_u$, defined over $\mathcal{P}_u$, which represents how each node $u$ ranks its paths: if $p(u,v)$ and $p'(u,v)$ in $\mathcal{P}_u$ and $\lambda_u(p'(u,v)) < \lambda_u(p(u,v))$ then $p(u,v)$ is said to be preferred over $p'(u,v)$.

**Definition 5:** The route selection problem is consistent with the stability function $\delta_r$ (for each $u \in V$ and $p(u,v)$ and $p'(u,v)$ in $\mathcal{P}_u$) if $\lambda_u(p'(u,v)) < \lambda_u(p(u,v))$ then $\delta_r(t) = \phi_i(t) - \phi_j(t) \geq 0$ (2) if $\lambda_u(p'(u,v)) < \lambda_u(p(u,v))$ then $\delta_r(t) = 0$.

**Theorem 1:** if $p(u,v)$ and $p'(u,v)$ in $\mathcal{P}_u$ and $p(u,v)$ is embedded in $p'(u,v)$ then the route selection problem is consistent with the stability function $\delta_r$ and the route selection is stable decreasing.
Proof: Assume without loss of generality that \( p_2(u,v) = (v_k, v_{k-1}, \ldots, v_1) \) is embedded in \( p_1(u,v) = (v_k, v_{k-1}, \ldots, v_1, v_0) \). Then, applying formula \( \delta_2(t) = \Phi_2(t) - \Phi_1(t) \), we find \( \delta_2 = \Phi_2(v_k, v_{k-1}, \ldots, v_1) - \Phi_1(v_k, v_{k-1}, \ldots, v_1, v_0) \).

Assuming that paths \( p_1 \) and \( p_2 \) result from the composition of sub-paths (without increasing their total length), \( p_1 \) and \( p_2 \) can be each written as the concatenation of three sub-paths \( p(v_k, v_{k-1}, \ldots, v_1) \) and \( p(v_k, v_{k-1}, \ldots, v_1, v_0) \), respectively. The resulting stability function: \( \delta_2 = [\Phi(v_k, v_{k-1}, \ldots, v_1) + \Phi(v_{k-1}, \ldots, v_1, v_0)] + [\Phi(v_{k-1}, \ldots, v_1, v_0)] = [\Phi(v_k, v_{k-1}, \ldots, v_1)] + [\Phi(v_{k-1}, \ldots, v_2)] \geq 0 \), proving the first part of Theorem 1.

Moreover, from its decomposition, the length \( d_2 \) of path \( p_2 \) verifies \( d_2(u,v) > d_1(u,v) \), where \( d_1 \) is the length of path \( p_1 \). Hence, the route selection is stretch decreasing.

3. Experimental Results

This section presents a set of experimental results obtained by applying the metrics defined in Section 2 to real-world BGP data. The dataset we used was obtained from the Route Views project [5] that comprises archives containing BGP feeds from a set of worldwide distributed Linux PCs running Zebra. As the only policy applied by Route Views sets the next hop to the peer IP address, only Adj_RIBs_In is accessible. As a consequence, the policy applied by Route Views sets the next hop to the peer IP address. We can observe that the difference \( \delta_2 \) result exclusively from the sub-paths defined between nodes \( v_{i+1}, v_1 \) and assuming that the only instabilities are policy and/or protocol-induced, we obtain \( \delta_2 = \Phi_2(v_{i+1}, v_1) + \Phi(v_{i+1}, v_1) - \Phi(v_{i+1}, v_1) = \Phi(v_{i+1}, v_1) \geq 0 \), proving the first part of Theorem 1.

Moreover, from its decomposition, the length \( d_2 \) of path \( p_2 \) verifies \( d_2(u,v) > d_1(u,v) \), where \( d_1 \) is the length of path \( p_1 \). Hence, the route selection is stretch decreasing.

4. Future Work

In this paper, we define and provide a first validation of several stability metrics that characterize the effects of BGP policy- and protocol-induced instabilities on local routers. Our initial results show that the proposed method enables identifying instability events and their impact on local routing tables. Ongoing work includes verifying if the route selection problem is consistent with the stability function \( \delta_2 \), and determining the general trade-offs between stability-based route selection and the resulting stretch increase/decrease factor on the selected routing paths.

5. References


Fig.1. Stability of the selected routes

Most Stable Route: Fig.2 shows that on average the routes have slowly decreasing stability when compared to the most stable route. As a result, the plot has a small but positive slope. It does sometime present local minima which mark the points in time when the most stable route experiences changes. The average of the maximum metric value per destination \( d \) shows a positive but larger slope: the most unstable routes have a faster paced increasing instability. Further, during the entire observation duration (6 days), a subset of routes continuously presented instabilities leading to a monotonic increase of the metric.

Best Selectable Route: It can be seen from Fig.3 that the BGP selected route has, on average, a better stability than the other routes out of which it is selected. However, comparison between Fig.2 and Fig.3 reveals that the selected route exhibits slightly more changes than the most stable route (a lower metric value indicates a higher instability). Additionally, for the avg curve, the local maxima are correlated with the local minima of the previous metric and are likewise due to a diminishing stability of the most stable route. One can also observe the same monotonically increasing trend of the metric for both the average and the maximum, due to routes with sustained instability.
A Deterministic Algorithm of Single Failed Node Recovery in MSR-based Distributed Storage Systems

Haotian Zhao
Department of Computer Science & Technology
University of Science & Technology of China
zhaohw@mail.ustc.edu.cn

Yinlong Xu
Department of Computer Science & Technology
University of Science & Technology of China
ylxu@ustc.edu.cn

1. INTRODUCTION

In distributed storage systems, redundant storage is widely used to achieve a good reliability at unreliable nodes. It is important to keep the redundancy in case of node failure. Dimakis et al. [1] studied the problem of reducing network bandwidth cost for failed nodes recovery and proposed two mechanisms MSR and MBR[4]. Previous works based on MSR mechanism like Y.Hu et al. [2] used random algorithms. But random algorithms need very large finite fields to achieve good reliability and can not guarantee the reconstruction property after the failed nodes being recovered.

In this paper, we study the problem of single failed node recovery in MSR-based distributed storage systems. We analyze the two encoding steps of single failed node recovery of MSR mechanism, and give encoding conditions of these two steps to keep the reconstruction property. Then we give a deterministic algorithm which meets these conditions. Our algorithm uses a much smaller finite field than random algorithms [3] and can guarantee the reconstruction property after the failed node being recovered. The encoding complexity will be greatly reduced with a much smaller finite field.

2. PROBLEM STATEMENT

Suppose that a file $F$ is divided into $k(n-k)$ fragments and the $k(n-k)$ fragments are encoded into $n(n-k)$ fragments stored evenly at $n$ nodes $N_1, N_2, \ldots, N_n$, where node $N_i$ stores $n-k$ fragments $F_1, F_2, \ldots, F_{n-k}$. A user, which connects to any $k$ of the $n$ nodes and downloads $k(n-k)$ fragments from these $k$ nodes, is able to decode the original $k(n-k)$ fragments, i.e., reconstruct $F$. This is what we say reconstruction property.

When a node failed, say $N_n$ without loss of generality, the repair process is evoked with a new node $N_0$ to replace the failed node. $N_0$ connects to all $n-1$ remaining nodes and each node $N_i (1 \leq i \leq n-1)$ generates an encoded fragment $E_i$ and transfers it to $N_0$. Then $N_0$ encodes the $n-1$ fragments $E_1, E_2, \ldots, E_{n-1}$ into $n-k$ fragments.

Figure 1: An example of single failed node recovery in a MSR-based distributed storage system with $(n,k) = (4,2)$.

$F_0, F_2, F_4, \ldots, F_{(n-k)}$ and stores them. The key issue is that the new node $N_0$ together with the $n-1$ remaining nodes $N_1 (1 \leq i \leq n-1)$ still keep the reconstruction property.

Figure 1 is an example with $(n,k) = (4,2)$. The original file is divided into 4 fragments $A, B, C, D$ and encoded into 8 fragments including the original 4 fragments. These 8 fragments are stored at 4 nodes, where each node has 2 fragments, as shown in the left half of Figure 1. A user can reconstruct the original file by connecting to any 2 nodes. For example, a user choosing node $N_1$ and $N_4$ gets 4 fragments $A+C, B+D, A+2C, B+2D$ and the encoding vectors of these 4 fragments are $(1, 0, 1, 0), (0, 1, 0, 1), (1, 0, 2, 0)$ and $(0, 1, 0, 2)$. Since these 4 fragments are linearly independent, the user is able to decode the original fragments $A, B, C, D$.

In the rest of this paper, we say “fragment” short for “the encoding vector of this fragment” when there is no ambiguity.

When a node failed, the system starts the repair process. Suppose that node $N_4$ in Fig. 1 fails, a new node $N_0$ is used to replace the failed node. $N_0$ gets an encoded fragment from each of the three remaining nodes. Then it encodes these three fragments into two fragments using a Vandermonde matrix, and stores these two fragments to repair the failed node. After repairing, $N_0$ together with $N_1, N_2, N_3$ still satisfy the reconstruction property. A user is able to reconstruct the original file by connecting to any 2 of these 4 nodes.

We can see that in the repair process, there are two steps which need encoding. The first one is that each of the $n-1$ remaining nodes generates an encoded fragment to transfer it to the new node, and the second one is the new node encodes these $n-1$ fragments into $n-k$ fragments. Because a random algorithm selects the encoding coefficients at these two steps randomly, the probability of reconstruction property being kept is approaching to 1 with the increase of $|F|$.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 20XX ACM X-XXXX-XX-X/XX/XX ...$10.00.
the size of the finite field, but can not be infinite large. To achieve a high reliability, a random algorithm needs a very large field $\mathbb{F}$. This paper will give a deterministic algorithm to guarantee the reconstruction property with a much smaller field.

3. CONDITIONS TO KEEP RECONSTRUCTION PROPERTY

In this section, we will analyze the conditions to keep reconstruction property at the two encoding steps. We now go on to the next part.

Firstly suppose that $E_1, \ldots, E_{n-k}$ are stored at $N_0$ to ease our analysis. Let $R = \{1, 2, \ldots, n-1\}$ be the set of subscripts of the remaining $n-1$ nodes, $S = \{s_1, s_2, \ldots, s_{k-1}\}$ of $k-1$ elements be a subset of $R$, and $T = \{t_1, t_2, \ldots, t_{n-k}\} = R - S$. After the repair process, a user may choose arbitrary $k$ nodes from $N_0, N_1, \ldots, N_{n-1}$ to reconstruct the original file. If these $k$ nodes do not include $N_0$, the user can reconstruct the original file as the same as before the repair process. Otherwise suppose that the chosen $k$ nodes are $N_0, N_{s_1}, N_{s_2}, \ldots, N_{s_{k-1}}$, the user gets $(n-1)+(k-1)(n-k)$ fragments $E_1, \ldots, E_{n-1}, F_{s_1}, \ldots, F_{s_{k-1}}, F_{s_{k-1}-(n-k)}$. Among these fragments, $E_i (1 \leq i \leq k-1)$ is a linear combination of $F_{s_1}, F_{s_2}, \ldots, F_{s_{i-1}}$ because $E_{s_i}$ is transmitted from $N_0$ to $N_i$. Apart from the $k-1$ fragments $E_{s_1}, E_{s_2}, \ldots, E_{s_k}$, there are $k(n-k)$ fragments $E_{t_1}, E_{t_2}, \ldots, E_{t_{n-k}}$, $F_{s_{n-k}}, \ldots, F_{s_{n-1}}$. If and only if these $k(n-k)$ fragments are linearly independent, the user can reconstruct the original file. So we have the following Lemma 1.

**Lemma 1:** Let $R = \{1, 2, \ldots, n-1\}$ be the set of subscripts of the remaining $n-1$ nodes, $S = \{s_1, s_2, \ldots, s_{k-1}\}$ of $k-1$ elements be a subset of $R$, and $T = \{t_1, t_2, \ldots, t_{n-k}\} = R - S$. The reconstruction property is kept only if that for any $S$ of $k-1$ elements, $k(n-k)$ fragments $E_{s_1}, E_{s_2}, \ldots, E_{s_{k-1}}, F_{s_1}, \ldots, F_{s_{n-k}}, F_{s_{n-k}-(n-k)}$ are linearly independent.

At the second encoding step, $N_0$ uses a $(n-k) \times (n-1)$ matrix $M$ for encoding, i.e., $(F_{00}, F_{01}, \ldots, F_{0(n-k)})^T = M \cdot (E_{01}, E_{02}, \ldots, E_{0(n-k)})^T$. A user may choose $k$ nodes $N_0, N_{s_1}, \ldots, N_{s_{k-1}}$, and get $(F_{00}, F_{01}, \ldots, F_{0(n-k)})^T$. Among these fragments, $(F_{00}, F_{01}, \ldots, F_{0(n-k)})^T = M \cdot (E_{s_1}, E_{s_2}, \ldots, E_{s_{k-1}})^T$, which can be considered as $k$ $n-k$ equations for $n-1$ fragments $E_{s_1}, \ldots, E_{s_{k-1}}$. Because $E_{s_i} (1 \leq i \leq k-1)$ is a linear combination of $F_{s_1}, F_{s_2}, \ldots, F_{s_{i-1}}$, so $E_{s_1}, E_{s_2}, \ldots, E_{s_{k-1}}$ are regarded as being got from $F_{s_1}, \ldots, F_{s_{k-1}}, F_{s_{k-1}-(n-k)}$. Then we have $k(n-k)$ variables $E_{s_1}, E_{s_2}, \ldots, E_{s_{n-k}}$ left and $n-k$ equations. If the coefficient matrix of the $n-k$ equations, i.e., the $n-k$ columns corresponding to $E_{t_1}, E_{t_2}, \ldots, E_{t_{n-k}}$ in $M$ are linearly independent, $E_{t_1}, E_{t_2}, \ldots, E_{t_{n-k}}$ can be decoded from $F_{00}, F_{01}, \ldots, F_{0(n-k)}$. Combining with Lemma 1, we have the following Lemma 2.

**Lemma 2:** If any $n-k$ columns of the encoding matrix $M$ are linearly independent and $E_1, E_2, \ldots, E_{n-k}$ satisfy Lemma 1, the reconstruction property is maintained after the repair process.

4. ENCODING ALGORITHM

Lemma 1 and Lemma 2 present the conditions to keep reconstruction property at the two encoding steps in single failed node recovery process of MSR mechanism. In this section, we will propose a recovery algorithm for single failed node which satisfies the conditions. Algorithm 1 is the algorithm for the first encoding step, i.e., constructing $E_1, E_2, \ldots, E_{n-k}$. The following is the main idea of Algorithm 1.

Node $N_1$ just chooses $E_{i1}$ as $E_{i}$. For node $N_i$ ($2 \leq i \leq n-k-1$), there are $c_i = \binom{n-k}{i-1}$ choices to select $k-1$ nodes from $N_1, \ldots, N_{n-k}$. Suppose that $S_1, S_2, \ldots, S_{c_i}$ are the $c_i$ sets of $k-1$ subscripts, each corresponding to a choice of the $k-1$ selected nodes, where $S_j = \{s_{j1}, s_{j2}, \ldots, s_{j(n-k-1)}\}$ are the set of the subscripts of the $k-1$ chosen nodes of the $j$-th choice. In Algorithm 1, we define "a fragment $D$ satisfies $S_j$ for node $N_i$ ($2 \leq i \leq n-k-1$)" means that $D, E_{s_1}, E_{s_2}, \ldots, E_{s_{k-1}}, F_{s_1}, \ldots, F_{s_{n-k}}, F_{s_{n-k}-(n-k)}$ are linearly independent.

In Algorithm 1, we construct an $E_i$ to satisfy $S_1, S_2, \ldots, S_{c_i}$. $E_i$ is gradually constructed as $E_{i1}, E_{i2}, \ldots, E_{ic_i}$, with $E_{ij}$ satisfying $S_1, \ldots, S_{ij}$ and $E_{ic_i} = E_i$. In the next paragraph, we will show that for any $S_j$, there is an $F_{im}$ from $F_1, \ldots, F_{n-k}$ that satisfies $S_j$. When $j = 1$, set $E_{i1}$ be the $F_{im}$ which satisfies $S_1$. When $j \geq 2$, $E_{ij}$ is constructed from $E_{i(j-1)}$ and the $F_{im}$ which satisfies $S_j$. If $E_{i(j-1)}$ satisfies $S_j$, set $E_{ij} = E_{i(j-1)}$ and $E_{ij}$ satisfies $S_1, \ldots, S_j$. Otherwise set $E_{ij} = r \cdot E_{i(j-1)} + F_{im} (r \in \mathbb{F})$. Since $F_{im}$ satisfies $S_j$ and $E_{i(j-1)}$ does not satisfy $S_j$, $E_{ij}$ satisfies $S_j$ for any $r$. So to make $E_{ij}$ satisfy $S_1, \ldots, S_j$, is to choose $r$ to make $E_{ij}$ satisfy $S_1, \ldots, S_j$. For each $S_j (1 \leq j \leq t)$, there is at most one $r$ that makes $E_{ij}$ do not satisfy $S_j$ (It will be shown in the following paragraph). So there are at most $\binom{n-1}{t}$ different elements in $\mathbb{F}$ chosen as $r$ make $E_{ij}$ do not satisfy $S_1, \ldots, S_{j-1}$, any element except these $j-1$ elements

**Algorithm 1:** Construct $E_1, E_2, \ldots, E_{n-k}$

1. Define $\text{satisfy}(D, S_j)$:
   (1) case of $2 \leq i \leq n-k-1$:
   $D, E_{i1}, \ldots, E_{i(j-1)}, F_{s_{ij}}, \ldots, F_{s_{ij}-(n-k)}$ are linearly independent;
   (2) case of $n-k \leq i \leq n-1$:
   $F_{s_1}, \ldots, F_{s_{i-(n-k-1)}}, F_{s_{i-(n-k-1)}}, \ldots, F_{s_{i-(n-k-1)}}$ are linearly independent.

2. $E_1 = F_{i1}$
   for $i = 2$ to $n-1$ do
   if $i < n-k$ then $c_i = \binom{n-k-1}{i-1}$ else $c_i = \binom{n-k}{i-1}$
   for $j = 1$ to $c_i$ do
   if $i < n-k$ then $S_j = \{s_{j1}, s_{j2}, \ldots, s_{j(n-k-1)}\}$
   else $S_j = \{s_{j1}, s_{j2}, \ldots, s_{j(n-k-1)}\}$
   $T_j = \{t_1, t_2, \ldots, t_{j-1}\} = R - S - \{i\}$
   for $m = 1$ to $n-k$ do
   if $\text{satisfy}(F_{im}, S_j)$ then break
   if $j = 1$ then $E_{i1} = F_{im}$
   else if $\text{satisfy}(E_{i(j-1)}, S_j)$ then $E_{ij} = E_{i(j-1)}$
   else for $r = 1$ to $|\mathbb{F}|$ do
   $E_{ij} = r \cdot E_{i(j-1)} + F_{im}$; $l = 1$
   while satisfy($E_{ij}, S_j$) and $l < j$ do
   $l = l + 1$
   if $l = j$ then break
   $E_{ic_i} = E_{i}$. 


Lemma 1. \(i \text{ most one } r \) will make \( E_i \) satisfy \( S_1, \ldots, S_r \). Finally we will get \( E_{ic} \) satisfies \( S_1, \ldots, S_r \), and \( E_{ic} \) is actually the \( E_i \) we need.

The existence of \( F_{im} \): \( E_1, E_{i-1}, F_{ij1}, \ldots, F_{ij1(n-k)}, \ldots, F_{ij(k-1)}, \ldots, F_{ij(k-1)(n-k)} \) are linearly independent from the reconstruction property and the construction of \( E_1, E_{i-1} \). If there is not an \( F_{im} \) from \( F_{11}, \ldots, F_{1(n-k)} \) that satisfies \( S_j \) for some \( j \), all \( F_{ij1}, \ldots, F_{ij1(n-k)} \) are linear combinations of \((k-1)(n-k) + (i-1)\) fragments \( E_1, E_{i-1}, F_{ij1}, \ldots, F_{ij1(n-k)} \). From the reconstruction property, \( k(n-k) \) fragments \( F_{11}, \ldots, F_{1(n-k)}, F_{1j1}, \ldots, F_{1j1(n-k)} \) are linearly independent and cannot be linear combinations of less than \( k(n-k) \) fragments. Since \((k-1)(n-k) + (i-1) < k(n-k)\), which leads to a contradiction, there is an \( F_{im} \) from \( F_{11}, \ldots, F_{1(n-k)} \) that satisfies \( S_j \) for any \( j \).

At most one \( r \) making \( E_{ij} \) do not satisfy \( S_j \): If there are \( r' \neq r \) such that neither \( E = r' \cdot E_{ij-1} + F_{im} \) nor \( E' = r' \cdot E_{ij} \). If \( E_{ij-1} + F_{im} \) and \( E_{ij} \) are linear combinations of \( E_1, E_{i-1}, F_{ij1}, \ldots, F_{ij1(n-k)}, \ldots, F_{ij(k-1)(n-k)} \). So \( E = E' = r' \cdot E_{ij} \) is a linear combination of these fragments. Because \( E_{ij-1} \) satisfies \( S_j \) and \( r' = r \neq 0 \), \( E' = E \) can not be a linear combination of these fragments, which leads to a contradiction. So there is at most one \( r \) making \( E_{ij} \) do not satisfy \( S_j \).

The construction of \( E_i \) at node \( N_i \) for the case of \( n-k \leq i \leq n-1 \) is quite similar to the case of \( 2 \leq i \leq n-k-1 \). The differences exist at: (1) The \( n-k-1 \) nodes are arbitrarily selected from \( N_1, \ldots, N_{n-1} \) instead of \( k-1 \) nodes from \( N_{n-1}, \ldots, N_1 \). (2) The definitions of \( c \) and \( S_j \) are different, (3) A new definition \( T_j = \{ t_1, t_2, \ldots, t_{n-k-1} \} = R - S - \{t\} \). (4) Term “\( D \) satisfy \( S_j \)” is changed to that \( D \) is linearly independent with \( E_{ij1}, E_{ij2}, \ldots, E_{ij(n-k-1)}, F_{ij1}, \ldots, F_{ij1(n-k)}, \ldots, F_{ij(k-1)(n-k)} \). See Algorithm 1 for details.

Lemma 3: Fragments \( E_1, E_2, \ldots, E_{n-1} \) constructed by Algorithm 1 satisfy Lemma 1.

Proof. For any \( T = \{ t_1, t_2, \ldots, t_{n-k} \} \), suppose \( t_1 < t_2 < \cdots < t_{n-k} \), then \( t_1, t_2, \ldots, t_{n-k} \geq n - k \). In Algorithm 1, since \( t_1, t_2, \ldots, t_{n-k} \) are from \( \{1, 2, \ldots, (n-k-1)\} \), no matter what \( t_1, t_2, \ldots, t_{n-k} \) are, while \( E_{ij} \) is linearly independent with \( E_{ij1}, E_{ij2}, \ldots, E_{ij(n-k-1)}, F_{ij1}, \ldots, F_{ij1(n-k)}, \ldots, F_{ij(k-1)(n-k)} \). This is actually Lemma 1.

At the second encoding step, any matrix that satisfies Lemma 2 can be used as the encoding matrix. In our algorithm, we use a Vandermonde matrix \( M = (m_{ij})_{(n-k)\times(n-1)} \) with \( m_{ij} = j^{i-1} \). If \( |F| \geq n-1 \), any \( n-k \) columns of \( F \) form an \( n-k \) order square Vandermonde matrix which any two columns are distinct from each other. So the determinant of the square Vandermonde matrix is nonzero and matrix has a full rank, which means that the \( (n-k)\times(n-1) \) Vandermonde matrix satisfies Lemma 2. Since the fragments constructed by Algorithm 1 satisfies Lemma 1 and the encoding matrix at the second step satisfies Lemma 2, the correctness of our algorithm is proved.

We now analyze the size of the finite field \( F \) used in our algorithm. At the first encoding step, if \( |F| \geq j \), there is \( r \) such that \( E_r = r \cdot E_{ij-1} + F_{im} (r \in F) \) satisfying \( S_1, \ldots, S_j \). For each \( N_i (2 \leq i \leq n-1) \), the largest \( j \) equals to \( c_i \).

When \( 2 \leq i \leq n-k-1 \), \( c_i = \binom{n-k-1}{i-1} \) and \( \text{Max} \{c_i\} = c_{n-1} = \binom{n-2}{n-k-1} = \binom{n-2}{n-k-1} \). So if \( |F| \geq \binom{n-2}{n-k} \), \( F \) is large enough for Algorithm 1 to successfully construct \( n-1 \) fragments \( E_1, E_2, \ldots, E_{n-1} \) that satisfy Lemma 1. If \( |F| \geq n-1 \), the Vandermonde matrix can be constructed at the second encoding step. So our algorithm needs a field of size \( \binom{n-2}{n-k} \). In practical storage systems, \( n-k \) is usually very small. So our algorithm needs a much smaller size of finite field.

The time complexity of Alg. 1 is \( O((n-2)\binom{n-2}{n-k})^3 \). But in practical distributed storage systems, \( n-k \) is usually quite small and \( n \) is not large. So compare with the time of encoding many MB or even GB of a file data, the time of computing the encoding vectors, i.e., the time of our algorithm, is much shorter and can even be negligible.

5. CONCLUSION
This paper studies the single failed node recovery in a MSR-based distributed storage system and analyzes the conditions to keep reconstruction property at the two encoding steps during the repair process. We give a deterministic algorithm that using a finite field smaller than random algorithms and guarantee the distributed storage system keeping reconstruction property after the repair process. For future work, we plan to study some particular circumstances, e.g., the original fragments being special constructed, and see if there are some good properties brought by these particular restriction.

6. ACKNOWLEDGMENTS
This work is supported by the National Nature Science Foundation of China under Grant No. 61073038 and the National High Technology Research and Development Program of China under Grant No. 2009AA01A348.

7. REFERENCES
1. INTRODUCTION

Data centers represent the fastest growing component of information and communication technologies (ICT) energy footprint. With the advent of cloud computing, data centers will increasingly be used to process a wide array of jobs with differing characteristics such as degree of parallelism, memory access patterns etc. From an energy efficiency perspective, the most energy efficient server architecture differs for jobs with different characteristics [4], motivating the need to consider heterogeneous data center designs consisting of many server types [3, 5].

Even though types of jobs that a data center is expected to serve might be known at design time, the workload statistics are often unknown until the data center is deployed. Therefore, data centers should be designed keeping in mind the uncertainty in workload statistics — in this paper, we outline a principled approach to designing energy-efficient, heterogeneous data centers that are robust against data center workload variations, using Wald’s minimax criterion as a starting point. In the proposed formulation, we assume that the only thing that is known at design time is an upper bound on the total rate (over all job types) at which jobs arrive at the data center, and design the data center to have the minimum worst-case energy consumption over all job type mixes. We then highlight a number of potential avenues for further investigation.

2. PROBLEM FORMULATION

We assume that there $N$ types of commercially available server architectures that can be used to design the data center, and that the data center is expected to serve $M$ different job types that are pre-characterized. The power consumed while executing an instance of a job of type $j$ ($1 \leq j \leq M$) on a server of type $i$ ($1 \leq i \leq N$) is given by $e_{ij}$, and the corresponding service rate is given by $\mu_{ij}$. Furthermore, the idle mode power consumption of server $i$ is given by $e_{i}^{idle}$.

We assume that per-type job arrivals are modeled as independent point Poisson processes, and that the arrival rate of jobs of type $j$ is given by $\lambda_j$. The vector $\Lambda = [\lambda_1, \lambda_2, \ldots, \lambda_M]^T$ is referred to as the arrival rate vector. To reflect the uncertainty in workload statistics at design time, we assume that the individual arrival rates for each job type are not known a priori. Instead, we only know an upper bound on the total arrival rate of all jobs into the data center, i.e., $\sum_{j \in [1,M]} \lambda_j \leq \lambda_{\text{max}}$.

We denote by $q_i$ the number of servers of type $i$ that are included in the data center. The vector $Q = [q_1, q_2, \ldots, q_N]^T$ is referred to as the data center design vector. Each server has an associated cost, for example, its market price, $c_i$, and the data center must be designed within a total budget given by $\text{budget}$.

Finally, we assume that there exists a data center resource manager (RM) that maps incoming jobs to servers at runtime. We assume that the RM can distinguish between job types and slots them into one of $M$ virtual queues, one for each job type. The RM schedules the execution of jobs in each virtual queue to a subset of available servers in the data center such that the queues remain stable, i.e., the service rate of each queue matches (or exceeds) its arrival rate. The matrix $R \in \mathbb{R}^{N \times M}$ is called the data center scheduling matrix, and element $r_{ij}$ indicates the number of servers of type $i$ that are used to process jobs of type $j$. The problem set-up is illustrated in Figure 1. It is interesting to note that the model described here is an generalization of the data center modeled discussed by [2], where the authors assume a homogeneous data center with single job and server types and a known job arrival rate.

The goal of the robust heterogeneous data center design problem is to find the optimal data center design vector $Q^*$, that minimizes the worst-case power consumption over all admissible arrival rate vectors, assuming that for any data
center design vector and arrival rate vector, the RM makes optimal scheduling decisions.

2.1 Minimax Optimization

We begin with some definitions for ease of exposition.

**Definition 1.** Given an arrival rate vector Λ and data center design vector \( Q \), \( R_S(Q, \Lambda) \) is defined to be the set of all RM scheduling decisions that ensure that the queues remain stable. More specifically, \( R_S(Q, \Lambda) \) is the feasible region of the following set of linear inequalities:

\[
\begin{align*}
\sum_{i\in[1,N]} \mu_{ij} r_{ij} &\geq \lambda_j \quad \forall j \in [1, M] \\
\sum_{i\in[1,N]} r_{ij} &\leq q_i \quad \forall i \in [1, N] \\
r_{ij} &\geq 0 \quad \forall i \in [1, N], \forall j \in [1, M]
\end{align*}
\]

**Definition 2.** \( E(Q, \Lambda, R) \) is defined to be the power consumption of a data center for a given data center design vector \( Q \), arrival rate vector \( \Lambda \), and data center scheduling matrix \( R \).

\[
E(Q, \Lambda, R) = \begin{cases} 
\sum_{\infty} \sum_{j\in[1,M]} r_{ij}(e_{ij} - e_{idle}) + q_i e_{idle} & R \in R_S(Q, \Lambda) \\
\text{otherwise}
\end{cases}
\]

We begin with by formulating a deterministic problem in which both arrival rate vector \( \Lambda \) and data center design vector \( Q \) are known, and the goal is to determine the optimal scheduling matrix \( R' \), that minimizes the data center power consumption.

**LP-Deterministic.**

\[
E^*(Q, \Lambda) = \min_{R} E(Q, \Lambda, R) \quad (1)
\]

subject to:

\[
R \in R_S(Q, \Lambda)
\]

The robust heterogeneous data center design problem can now be written as:

**Minimax.**

\[
\min \max_{Q, \Lambda} E^*(Q, \Lambda) \quad (2)
\]

subject to:

\[
\sum_{i\in[1,M]} \lambda_i \leq \lambda_{\text{max}} \]

\[
\sum_{i\in[1,M]} c_i q_i \leq c_{\text{budget}}
\]

To solve the minimax problem, we first define some additional terminology.

**Definition 3.** \( \Lambda' \) (\( i \in [1, M] \)) represents an arrival rate vector in which jobs of type \( i \) arrive at rate \( \lambda_{\text{max}} \) and all other jobs arrive at rate 0. Formally, \( \lambda_j' = 0 \) if \( j \neq i \) and \( \lambda_j' = \lambda_{\text{max}} \) if \( j = i \).

**Lemma 1.** Let \( \Lambda' \) be any admissible arrival rate vector, i.e., \( \sum_{i\in[1,M]} \lambda_i' \leq \lambda_{\text{max}} \), then:

\[
E^*(Q, \Lambda') \leq \max_{i\in[1,M]} (E^*(Q, \Lambda'))
\]

**Proof.** We begin by noting that for any \( \Lambda' \), there exists a vector \( \delta = [\delta_1, \delta_2, \ldots, \delta_M]^T \) such that \( \Lambda' = \sum_{i\in[1,M]} \delta_i \lambda_i' \), where \( \delta_i \geq 0 \) (\( \forall i \in [1, M] \)) and \( \sum_{i\in[1,M]} \delta_i \leq 1 \).

Now define \( R' = \sum_{i\in[1,M]} \delta_i R^*(Q, \lambda_i') \). It can be shown that \( R' \) is a feasible solution for data center design vector \( Q \) and arrival rate \( \lambda_i' \), i.e., \( R' \in R_S(Q, \lambda_i') \).

In addition, we can show that:

\[
E(Q, \Lambda', R') = \sum_{i\in[1,M]} \delta_i E^*(Q, \lambda_i')
\]

By definition, \( E^*(Q, \lambda_i') \leq E(Q, \lambda_i', R') \) and given that \( \sum_{i\in[1,M]} \delta_i E^*(Q, \lambda_i') \leq \max_{i\in[1,M]} (E^*(Q, \lambda_i')) \), the desired result is obtained.

**Lemma 1** allows us to simplify the Minimax problem as follows:

**Minimax.**

\[
\min_{Q, \gamma} \gamma \quad (3)
\]

subject to:

\[
E^*(Q, \Lambda') \leq \gamma \\
\sum_{i\in[1,M]} c_i q_i \leq c_{\text{budget}}
\]

We will now determine an analytical expression for \( E^*(Q, \lambda_i') \) (\( t \in [1, M] \)). Recall that \( E^*(Q, \lambda_i') \) represents the minimum data center power consumption given a data center design \( Q \) and assuming that jobs of only type \( t \) arrive at the data center, and the rate at which they arrive is \( \lambda_{\text{max}} \). We will assume, for notational simplicity and without any loss of generality, that for job type \( t \), the following relationship holds:

\[
\frac{e_{it} - e_{idle}}{\mu_{it}} \leq \frac{e_{it} - e_{idle}'}{\mu_{it}} \quad \forall i, j \in [1, N]; j \geq i
\]

Note that \( e_{it} - e_{idle}' \) can be viewed, in a sense, as a measure of the energy efficiency of servers of type \( i \) in processing jobs of type \( t \). This is becomes more clear if we set \( e_{idle}' = 0 \). Equation 4 simply says that for jobs of type \( t \), the servers are indexed in decreasing order of energy efficiency\(^1\). Under this assumption, we can write an expression for \( E^*(Q, \lambda_i') \).

**Definition 4.**

\[
e_{it} - e_{idle}' = e_{diff} \quad \forall i \in [1, N]
\]

**Lemma 2.**

\[
E^*(Q, \lambda_i') = \max_{j\in[1,M]} \left( \frac{\lambda_{\text{max}} - \sum_{i=1}^{j-1} q_i \mu_{it} e_{idle}'}{\mu_{jt}} \right)
\]

\(^1\)Similar results can be derived for every other task type by appropriately re-ordering the indices in decreasing order of energy efficiency.
to have unit cost and the maximum total arrival rate of jobs to be zero for both servers. Finally, each server is assumed to process each job at the same rate of 1 job/second, i.e., can process each job at the same rate of 1 job/second, i.e.,

\[ \frac{e_{it} - e_{i \text{idle}}}{\mu_{it}} + q_i e_{i \text{idle}} \]

subject to:

\[ \sum_{i=1}^{3} \mu_{it} r_{it} \geq \lambda_{\text{max}} \]

\[ r_{it} \leq q_i \quad \forall i \in [1, N] \]

\[ r_{it} \geq 0 \quad \forall i \in [1, N] \]

Substituting the variable \( g_{it} = \mu_{it} r_{it} \) (\( \forall i \in [1, N] \)), we get the following equivalent LP:

\[ E^*(Q, \Lambda^i) = \min_{g_{it}} \sum_{i \in [1, N]} (g_{it} \frac{e_{it} - e_{i \text{idle}}}{\mu_{it}} + q_i e_{i \text{idle}}) \]

subject to:

\[ \sum_{i \in [1, N]} g_{it} \geq \lambda_{\text{max}} \]

\[ g_{it} \leq \mu_{it} q_i \quad \forall i \in [1, N] \]

\[ g_{it} \geq 0 \quad \forall i \in [1, N] \]

It can be shown that in this case the greedy solution to the LP problem, i.e., one in which the most energy efficient servers are allocated first, followed by the next most efficient servers and so on till the total service rate becomes at least equal to \( \lambda_{\text{max}} \), is also the optimal solution. Note from Equation 7 servers will get picked in decreasing order of the coefficients \( \frac{e_{it} - e_{i \text{idle}}}{\mu_{it}} \).

Greedy allocation of servers to the incoming job stream results in the power consumption being a piecewise linear function of the data center design vector \( Q \), which is apparent from Equation 5.

Lemma 2 allows us to re-write the Minimax constraint \( E^*(Q, \Lambda^i) \leq \gamma \) as \( N \) linear constraints, i.e.,:

\[ \sum_{j=1}^{N} \left( q_i e_{i,j}^{\text{diff}} + \frac{\lambda_{\text{max}} - \sum_{j=1}^{i-1} q_i \mu_{it} e_{i,j}^{\text{diff}}}{\mu_{it}} \right) \leq \gamma \quad \forall j \in [1, N] \]

Therefore, the robust heterogeneous data center design problem can be expressed as a LP with \( N + 1 \) variables (data center design vector \( Q \) and \( \gamma \)) and \( NM + 1 \) constraints.

### 2.2 Illustrative Example

We use a simple example with two job types (\( M = 2 \)) and two server types (\( N = 2 \)) to illustrate the proposed approach. In this simple example, we assume that each server can process each job at the same rate of 1 job/second, i.e., \( \mu_{ij} = 1 \) (\( i \in [1, 2] \) and \( j \in [1, 2] \)). Moreover, in terms of energy consumption, we assume that servers of type 1 are optimized to run jobs of type 1, while servers to type 2 are optimized to run jobs of type 2. In particular, \( e_{11} = \lambda_{\text{max}} \), \( e_{12} = 2, e_{21} = 4, e_{22} = 1 \). The idle power consumption is assumed to be zero for both servers. Finally, each server is assumed to have unit cost and the maximum total arrival rate of jobs from Equation 5.

\[ E^*(Q, \Lambda^i) = \min_{R} \sum_{i \in [1, N]} (r_{it} (e_{it} - e_{i \text{idle}}) + q_i e_{i \text{idle}}) \]

into the system is assumed to be 10 jobs/second. Figure 2 depicts the optimal data center design for different values of \( c_{\text{budget}} \), which in this case simply constrains the total number of servers allowed in the data center. As can be seen from the figure, if only ten servers are allowed, the optimal design has 6 servers of type 1 and 4 servers of type 2, and dissipates 22 units of power in the worst case.

### 3. Future Work

The robust heterogeneous data center design problem we address in this paper is based implicitly on a number of assumptions that we are now working on relaxing. First, we assume that the data center performance specification only requires queue stability but does not account for either queuing or execution latency. We assumed also that the servers exist in one of two states, either ON or IDLE, but in general, servers have access to a range of power states which they can switch between at run-time. In addition, worst-case optimization using a minimax objective function is perhaps too pessimistic. We are currently looking at addressing scenario when some of the moments of arrival rate vector \( \Lambda \) are known. Much of this work falls squarely within the framework of adjustable robust optimization [1], with the data center design vector \( Q \) corresponding to the “here-and-now” variables, \( R \) (or any run-time scheduling decision) corresponding to the “wait-and-see” variables and \( \Lambda \) corresponding to the parametric uncertainty.

### 4. References


Figure 2: Illustrative Example
On Network Criticality in Wireless Networks

Ali Tizghadam  
University of Toronto  
ali.tizghadam@utoronto.ca

Alberto Leon-Garcia  
University of Toronto  
alberto.leongarcia@utoronto.ca

ABSTRACT
Network criticality is a graph-theoretic metric that quantifies network robustness, and that was originally designed to capture the effect of environmental changes in core networks. This paper investigates the application of network criticality in designing robust power allocation and flow assignment algorithms for wireless networks. Achieving robust behavior in wireless networks is a challenging task due to constant changes in channel conditions and the interference. We consider network criticality as a natural robustness metric, and propose approaches to preserve the useful convexity properties of network criticality, while resolving issues related to the non-convexity of Shannon’s capacity.

1. INTRODUCTION
Due to the time-varying nature of the wireless channels, the design of resource allocation methods in wireless networks significantly differs from that in wireline networks. Channel variations make it necessary to design robust algorithms for power allocation and flow assignment. The robust design of wireless networks has been tackled from different standpoints. For example in [1], the rate maximization in cognitive radio networks (CRN) is considered. It is shown that direct maximization of user rates results in large ripples and instability in the achievable rates of individual users. The authors in [1] proposed a max-min method to design a robust rate maximization method. The proposed algorithm in [1] removes instability; however, the total achievable rate is significantly reduced in comparison to the original case.

Network criticality is a robustness metric to capture the effect of environmental changes such as traffic variation and topology changes in networks. A network is modeled as a weighted graph, where the weights of a link denote the capacity or desirability of a path. Network criticality is defined as the average random-walk betweenness of a link (node) normalized by its weight. This quantity is independent of link (node) location and it is a decreasing and strictly convex function of link weights. Network criticality can be written in terms of the components of the undirected Moore-Penrose Laplacian matrix [2]:

$$\hat{\tau} = \frac{2}{n-1} Tr(L^+)$$

There is a useful interpretation of network criticality in terms of electrical circuits: network criticality is the unweighted average of the equivalent resistances in a network of resistors. Therefore optimizing criticality is equivalent to minimizing the average resistance or maximizing the average conductance of a network, which explains why network criticality can be considered as a global robustness metric. Furthermore, according to Thomson’s principle from physics, Kirchhoff equations yield the set of currents that minimize the overall power consumed in a resistive network [3]. Clearly, algorithms based on network criticality attempt to route network flows similarly to the way electric currents flow through an electrical network.

Network criticality was designed for wireline core networks, but it is conceptually applicable wherever the network is represented with a weighted graph in which the link weights are well defined. In this paper, we investigate appropriate definitions of link weights that allow us to use network criticality as an objective while preserving its convexity. In the following discussion, we assume that we have a wireless network in which the capacity of a link depends on not only the power allocated to itself, but also the powers allocated to the other links due to interference.

Throughout, we assume that the topology of a wireless network is given by a directed graph $G(N, E, W)$ (in most general form, it is a full-mesh, since potentially all the wireless nodes can send/receive traffic to/from all other nodes), where $N$, $E$, and $W$ denote the link set, node set, and link weight matrix respectively. The weight matrix is in general asymmetric; however, in calculating the network criticality we use an undirected symmetric matrix of the graph defined as $W_{sym} = \frac{W + W^T}{2}$, where $W^T$ denotes the transpose of $W$. Clearly $W_{sym}$ is a symmetric matrix.

2. SERVICE TIME CONTROL
Let $G = [g_{ij}]$ be the channel gain matrix of the wireless network, where $g_{ij}$ is the power gain from the transmitter of link $i$ to the receiver of link $j$. In an ideal interference-free environment only the diagonal entries of $G$ (i.e. $g_{ii}$, $\forall i \in E$, where $E$ is the set of wireless links) are nonzero, but in a typical wireless network the off-diagonal terms $g_{ij}$, $\forall i \neq j$
are nonzero and due to interference. In order to model the variability of channel gains, we adopt a probabilistic framework in which $G \in \{G_1, G_2, ..., G_k\}$, where each channel gain matrix $G_u$ ($u$ is referred to as the channel gain state) exists with probability $p_u$. In such environments and if there is no prior interference cancellation mechanism in place, the upper bound for achievable rate is given by Shannon’s formula:

$$r_{i,u} \leq \log(1 + \frac{g_{i,u}p_{i,u}}{\sigma_i + \sum_{k \neq i} g_{k,i}p_{k,u}})$$  \hspace{1cm} (1)

where $r_{i,u}$ and $p_{i,u}$ denote the rate (bits per channel use) and power of the transmitter of wireless link $i$ at channel gain state $u$ respectively. Furthermore, let $p_{i,u}$ denote the transmit power of user $i$ in channel gain state $u$. We assume the transmitter of link $i$ has the transmit power constraint of $p_i^{max}$, i.e. $0 \leq p_{i,u} \leq p_i^{max}$. Suppose at a known channel gain state $u$, the transmitter of link $i$ wishes to send equal-length packets of length $\gamma_i$ to the receiver of link $i$. Let $t_{i,u}$ be the number of channel uses or the service time of each packet, then:

$$t_{i,u} \geq \frac{\gamma_i}{r_{i,u}}$$  \hspace{1cm} (2)

We introduce variable $x_i$ as the maximum SINR (signal to interference plus noise ratio) at the receiver of link $i$. Hence:

$$x_{i,u} \leq \frac{g_{i,u}p_{i,u}}{\sigma_i + \sum_{k \neq i} g_{k,i}p_{k,u}}$$  \hspace{1cm} (3)

Combining inequalities (2) and (3), we have:

$$\frac{\gamma_i}{\log(1 + x_{i,u})} \leq t_{i,u}$$  \hspace{1cm} (4)

Our goal is to find to construct a convex optimization problem in order to find the optimal values for service times and user powers to minimize the network criticality (considering (2) and (4) as constraints of the optimization problem); however, in their present form inequalities (2) and (4) do not impose convex sets. To find an equivalent convex form, we rewrite constraint (3) as:

$$\frac{x_{i,u}}{g_{i,u}p_{i,u}} + \sum_{k \neq i} \frac{x_k g_{k,i}p_{k,u}}{g_{i,u}p_{i,u}} \leq 1$$  \hspace{1cm} (5)

We apply the following change of variables:

$$y_{i,u} = \ln(x_{i,u}), \quad q_{i,u} = \ln(p_{i,u})$$  \hspace{1cm} (6)

Equation $q_i = \ln(p_{i,u})$ involves a slight loss of generality, because we should have $p_{i,u} > 0$, but we will see that it is worth since it provides convexity. Applying equation (6) in (4) and (5) results in:

$$\frac{\gamma_i}{\log(1 + e^{y_{i,u}})} \leq t_{i,u}$$  \hspace{1cm} (7)

$$\log(1 + e^{y_{i,u}}) \leq t_{i,u}$$

$$\ln\{e^{y_{i,u}} - q_{i,u} - \ln(q_{i,u})\} + \ln(q_{i,u}) \leq 0$$  \hspace{1cm} (8)

Convexity of constraint (7) can be examined by evaluating the second derivative of function $f(x) = \frac{1}{\log(1 + e^x)}$ with respect to $x$. Note that $f(x) = \frac{1}{\ln(1 + e^x)}$, therefore, we have:

$$\frac{d^2}{dx^2} f(x) = -\ln(2) \frac{e^x}{(1 + e^x)^2} \left(1 - \frac{2e^x}{ln(1 + e^x)}\right)^2$$  \hspace{1cm} (9)

Equation (9) shows that $(f(x)$ is a decreasing function of $x$. By finding the derivative of both sides in equation (9) and after some simplification, we have:

$$\frac{d^2}{dx^2} f(x) = \frac{1 - \frac{2e^x}{ln(1 + e^x)}}{ln(1 + e^x)^2} f^2(x)$$

$$\frac{d^2}{dx^2} f(x) > \frac{1}{ln(1 + e^x)^2} f^2(x) > 0$$  \hspace{1cm} (10)

Considering the fact that $e^x > 0$, we have $e^x > ln(1 + e^x)$, hence we conclude that:

$$1 - \frac{2e^x}{ln(1 + e^x)} < -1$$

$$\frac{d^2}{dx^2} f(x) > \ln(2) \frac{e^x}{(1 + e^x)^2} f^2(x) > 0$$  \hspace{1cm} (11)

The second derivative of $f(x)$ according to (11) is nonnegative; therefore, constraint (7) represents a convex set. Furthermore, the left hand side of constraint (8) is of the form Log-Exp-Sum which is also convex.

We are now ready to formulate our robust convex optimization problem using network criticality. There are different approaches to construct such an optimization problem. The general idea is to define weights for the graph such that an increase in the weight increases the desirability of the link, while at the same time preserving the convexity of network criticality with respect to the variables. We will discuss some examples of such weight assignments for different purposes.

First, we define link weights as a concave decreasing function of the mean service time over all possible channel gain states: $\tilde{\ell}_i = \sum p_{i,u} \ell_{i,u}$’s (service times), i.e. $w_{i,u} = \Phi(\ell_{i,u})$, where $\Phi$ is a concave decreasing function of $\ell_{i,u}$. We symmetrize the weight matrix (denoted by $W_{sym}$) as discussed in section 1, and we use it to calculate $\tilde{\tau}$. Then, the robust convex optimization will be:

$$\text{minimize} \quad \tilde{\tau}(W_{sym})$$

$$\text{subject to}$$

$$(\forall i \in E, \forall u \in \{1, 2, ..., k\}) :$$

$$\ell_{i,u} = \sum_{k \neq i} p_{k,u} \ell_{k,u}$$

$$\frac{\gamma_i}{\log(1 + e^{y_{i,u}})} \leq t_{i,u}$$

$$\ln\{e^{y_{i,u}} - q_{i,u} - \ln(q_{i,u})\} + \ln(q_{i,u}) \leq 0$$  \hspace{1cm} (8)

$$q_{i,u} \leq \ln(p_{i,u}^{max})$$

Note that since $\Phi$ is a concave function of $\tilde{\ell}_i$’s and $\tilde{\tau}$ is a convex decreasing function of weights [2], network criticality ($\tilde{\tau}$) is a convex function of $\tilde{\ell}_i$’s [4]. Moreover, optimization problem (12) is valid for the whole range of SINR values (low SINR regime up to high SINR regime).

We discuss a second robust optimization problem, where our goal is to provide robustness with regards to the SINR. To this end, we define the weight of a link $i$ for the channel gain state $u$ as $w_{i,u} = y_{i,u}$ (note that increasing $y_{i,u}$ will improve desirability of link $i$), and after symmetrizing the weight matrix, we let the mean of network criticality over all channel gain states be the objective function of the optimization problem. Let $\tilde{\tau}$ denote the network criticality for channel gain state $u$, then the optimization problem can be
summarized as follows:

\[
\begin{aligned}
\text{minimize} & \quad \sum_u \tilde{p}_u \tilde{r}_u \\
s\text{subject to} & \quad (\forall i \in E, \forall u \in \{1, 2, \ldots, k\} :)
\end{aligned}
\]

\[
\begin{aligned}
\frac{\gamma_i}{\log(1+e^{\gamma_i})} & \leq t_{i,u} \\
\sum_{k \neq i} e^{y_{k,u} - q_{i,u} + q_{k,u}} & \leq \ln(p_{i,u}^{\max}) \\
q_{i,u} & \leq \ln(p_{i,u}^{\max}) \\
y_{i,u} & \geq 0
\end{aligned}
\]

The last constraint (nonnegativity of \(y_{i,u}\)) is added to guarantee that the link weights are nonnegative (in order for network criticality to exist). In terms of SINR, the nonnegativity of \(y_{i,u}\) according to (4) means that signal-to-noise-interference ratio should be more than 1. This is true in most practical situations as usually \(g_{i,u} > g_{k,u}\). Please note that we have not assumed that the system is in high SINR regime. The formulation is valid for low SINR regime as long as \(SINR > 1\).

We can define link weights differently to achieve other objectives. Suppose, we need to have a robust power allocation scheme as the primary goal. Then we define the weight of link \(i\) as \(w_{i,u} = q_{i,u}\), and after symmetrizing the weight matrix, we calculate the mean of network criticality in optimization problem (13) based on that. The rest is similar to the optimization problem (13).

3. HIGH-SINR REGIME

The special case of high-SINR regime is easier and it is already considered in the literature. Let \(E = [\gamma_{ij}]\) be the traffic flow matrix that needs to be routed in the wireless network. In [5], the joint robust resource allocation and power assignment problem is discussed. We adopt the problem introduced in [5] and provide a robust version of it. We would like to have robustness in the traffic distribution; therefore, we choose the available capacity of a communication link (i.e. the Shannon capacity of a link minus the traffic flow passing through the link) as the weight of our link: \(w_{i,u} = c_{i,u} = f_{i,u} - f_{i,u}\), where \(f_{i,u}\) is the total flow of link \(i\) for channel gain state \(u\), and \(c_{i,u} = \log(1 + \frac{q_{i,u}}{\gamma_{i,u} \sum_{k \neq i} \frac{g_{k,u}}{g_{i,u}}})\).

Applying the change of variable \(q_{i,u} = \ln(p_{i,u})\), we can approximate the wireless link capacity as follows [5]:

\[
c_{i,u} = \frac{-log(a_{i,u})}{\gamma_{i,u}} - \sum_{k \neq i} \frac{g_{k,u}}{g_{i,u}} e^{\gamma_{i,u} q_{i,u}}
\]

Equation (14) is a Log-Sum-Exp expression; therefore, it is a concave function (due to the negative sign) [4]. In order to write the flow conservation for the network, we define total incoming traffic to node \(d\) as \(\gamma^{(d)} = \sum_{i} \gamma_{i,d}\). Moreover, we define the flow of link \(i\) for destination \(d\) as \(f^{(d)}_{i}\). Then the flow conservation equations can be written in matrix form as \(Bf^{(d)} = \gamma^{(d)}\), where \(f^{(d)}\), \(\gamma^{(d)}\), and \(B\) denote the vector of link flows for node \(d\), the external input traffic vector for \(d\), and the graph incidence matrix respectively. Now suppose the goal is to minimize the total power assigned to all the wireless users, while we would like to robustly distribute the traffic flows. We can provide robustness by introducing a constraint to guarantee that the value of network criticality does not exceed a given threshold. The optimization problem for joint resource and power assignment is:

\[
\begin{aligned}
\text{Minimize} & \quad \sum_u e^{\bar{q}_u} \\
\text{Subject to} & \quad (\forall i \in E, \forall d \in N, \forall u \in \{1, 2, \ldots, k\} :)
\end{aligned}
\]

\[
\begin{aligned}
\bar{q}_i & = \sum_u \tilde{p}_u \bar{q}_{i,u} \\
Bf^{(d)} & = \gamma^{(d)} \\
f_{i,u} & \leq \frac{\alpha}{\bar{q}_{i,u}} - \sum_{k \neq i} \frac{g_{k,u}}{g_{i,u}} e^{\gamma_{i,u} q_{i,u}} - w_{i,q} \\
f_{i,u}^\max & \geq 0 \\
\sum_u \tilde{p}_u \bar{r}_u & \leq a
\end{aligned}
\]

where \(a\) is a known upper bound for network criticality. Optimization problem (15) assigns user powers and distributes traffic flows such that the sum of power consumption is minimized while we make sure that network criticality remains below the specified upper bound (i.e. \(a\)), which guarantees the robustness in flow distribution. Lower constraint bound for network criticality makes the problem more robust but we expect to see more power required.

4. CONCLUSIONS AND ROAD MAP

In this paper we investigated a number of approaches to construct convex optimization problems to provide robust joint resource allocation and power assignment for wireless interference aware networks. We have shown how network criticality can be introduced to provide robust designs while maintaining desirable convexity properties.

We are in the initial steps of developing robust solutions for wireless networks using network criticality. In this paper we explained some theoretical methods that we have developed to construct convex optimization problems for joint power allocation and flow assignment. The next step is to test the proposed frameworks on real wireless systems. Our goal is to design robust algorithms based on the proposed optimization problems to do the power allocation and flow assignment in a distributed manner.

5. REFERENCES


ABSTRACT
The spread of new ideas, behaviors or technologies has been extensively studied using epidemic models. Here we consider a model of diffusion where the individuals' behavior is the result of a strategic choice. We study a simple coordination game with binary choice and give a condition for a new action to become widespread in a random network. We also analyze the possible equilibria of this game and identify conditions for the coexistence of both strategies in large connected sets. Finally we look at how firms use social networks to promote their goals with limited information. Our results differ strongly from the one derived with epidemic models and show that connectivity plays an ambiguous role: while it allows the diffusion to spread, when the network is highly connected, the diffusion is also limited by high-degree nodes which are very stable.

Categories and Subject Descriptors
G.3 [Mathematics of Computing]: PROBABILITY AND STATISTICS

Keywords
social networks, diffusion, random graphs, empirical distribution

1. INTRODUCTION
To illustrate our point, consider the basic game-theoretic diffusion model proposed in [7]. Consider a graph $G$ in which the nodes are the individuals in the population and there is an edge $(i,j)$ if $i$ and $j$ can interact with each other. Each node has a choice between two possible behaviors labelled $A$ and $B$. On each edge $(i,j)$, there is an incentive for $i$ and $j$ to have their behaviors match, which is modeled as the following coordination game parameterised by a real number $q \in (0,1)$: if $i$ and $j$ choose $A$ (resp. $B$), they each receive a payoff of $q$ (resp. $(1-q)$); if they choose opposite strategies, then they receive a payoff of 0. Then the total payoff of a player is the sum of the payoffs with each of her neighbors. Consider a network where all nodes initially play $A$. If a small number of nodes are forced to adopt strategy $B$ and other nodes in the network apply best-response updates, then these nodes will be repeatedly applying the following rule: switch to $B$ if enough of your neighbors have already adopted $B$. There can be a cascading sequence of nodes switching to $B$ such that a network-wide equilibrium is reached in the limit. Most of the results on this model are restricted to deterministic (possibly infinite) graphs. In this work, we analyze the diffusion in the large population limit when the underlying graph is a random network $G(n,d)$ with $n$ vertices and where $d = (d_i)^n_i$ is a given degree (i.e. number of neighbors) sequence, similarly to [4].

In this simple model, agents play a local interaction binary game where the underlying social network is modeled by a sparse random graph. First considering the deterministic best response dynamics, we compute the contagion threshold for this model, confirming the heuristic result of [8]. We find that when the social network is sufficiently sparse, the contagion is limited by the low connectivity of the network; when it is sufficiently dense, the contagion is limited by the stability of the high-degree nodes. This phenomenon explains why contagion is possible only in a given range of the global connectivity (i.e. the average number of neighbors).

We identify the set of agents able to trigger a large cascade: the pivotal players, i.e. the largest component of players requiring a single neighbor to change strategy in order to follow the change. When contagion is possible, both in the low and high-connectivity cases, the number of pivotal players is low, resulting in rare occurrences of cascades. However in the high-connectivity case, we found that the system displays a robust-yet-fragile quality: while the cascades are very rare, their sizes are very large. This feature makes global contagions exceptionally hard to anticipate.

Motivated by social advertising, we also consider cases where contagion is not possible if the set of initial adopters is too small, i.e. a negligible fraction of the total population, as in [3]. We compute the final size of the contagion as a function of the fraction of the initial adopters. We find that the low and high-connectivity cases still have different features: in the first case, the global connectivity helps the spread of the contagion while in the second case, high connectivity inhibits the global contagion but once it occurs, it facilitates its spread.

We also analyze possible equilibria of the game and in particular, we find conditions for the existence of equilibria with co-existent conventions. Finally, we analyze a gen-

Marc Lelarge
INRIA-ENS
Paris, France
marc.lelarge@ens.fr
eral percolated threshold model for the diffusion allowing to
give different weights to the (anonymous) neighbors. This
model allows us to study rigorously semi-anonymous thresh-
old games of complements with local interactions on a com-
plex network. Our general analysis gives explicit formulas
for the spread of the diffusion in terms of the initial con-
tion, the degree sequence of the random graph, and the
distribution of the thresholds.

We refer to [5] for these last two points.

2. ANALYSIS OF A SIMPLE MODEL OF
cascades

2.1 Graphs: the configuration model

We consider a set \([n] = \{1, \ldots, n\}\) of agents interacting
over a social network. Let \(d = (d^{(n)}_i)_{i=1}^n\) be a se-
quen of non-negative integers such that \(\sum_{i=1}^n d_i\) is even.
For notational simplicity we will usually not show the depen-
dency on \(n\) explicitly. This sequence is the degree sequence
of the graph: agent \(i \in [n]\) has degree \(d_i\), i.e. has \(d_i\) neigh-
ors. We define a random multigraph (allowing for self-loop
and multiple links) with given degree sequence \(d\), denoted
by \(G'(n, d)\) by the configuration model [1]. Conditioned on
the multigraph \(G'(n, d)\) being a simple graph, we obtain a
uniformly distributed random graph with the given degree
sequence, which we denote by \(G(n, d)\).

We will let \(n \to \infty\) and assume that we are given \(d = (d_i)_{i=1}^n\)
satisfying the following regularity conditions, see [6]:

**Condition 1.** For each \(n\), \(d = (d^{(n)}_i)_{i=1}^n\) is a sequence
of non-negative integers such that \(\sum_{i=1}^n d_i\) is even and, for
some probability distribution \(p = (p_r)_{r=0}^\infty\) independent of \(n,

(i) \(|\{i \mid d_i = r\}|/n \to p_r\) for every \(r \geq 0\) as \(n \to \infty\);

(ii) \(\lambda := \sum_{r \geq 0} r p_r \in (0, \infty)\);

(iii) \(\sum_{i \in [n]} d_i^2 = O(n)\).

(iv) \(\sum_{i \in [n]} d_i^3 = O(n)\).

In words, we assume that the empirical distribution of the
degree sequence converges to a fixed probability distribution
\(p\) with a finite mean \(\lambda\).

2.2 Contagion threshold for random networks

An interesting perspective is to understand how different
network structures are more or less hospitable to cascades.
Going back to previous diffusion model, we see that the lower
\(q\) is, the easier the diffusion spreads. In [7], the contagion
threshold of a connected infinite network (called the cascade
capacity in [2]) is defined as the maximum threshold \(q_e\) at
which a finite set of initial adopters can cause a complete cas-
cade, i.e. the resulting cascade of adopters of \(B\) eventually
causes every node to switch from \(A\) to \(B\). There are two
possible models to consider depending whether the initial
adopters changing from \(A\) to \(B\) apply or not best-response
update. It is shown in [7] that the same contagion threshold
arises in both models. In this section, we restrict ourselves
to the model where the initial adopters are forced to play
\(B\) forever. In this case, the diffusion is monotone and the
number of nodes playing \(B\) is non-decreasing. We say that
this case corresponds to the permanent adoption model: a
player playing \(B\) will never play \(A\) again.

We now compute the contagion threshold for a sequence
of random networks. Since a random network is finite and
not necessarily connected, we first need to adapt the de-

inition of contagion threshold to our context. For a graph
\(G = (V, E)\) and a parameter \(q\), we consider the largest con-

cected component of the induced subgraph in which we keep
only vertices of degree strictly less than \(q^{-1}\). We call the ver-
tices in this component pivotal players: if only one pivotal
player switches from \(A\) to \(B\) then the whole set of pivotal
players will eventually switch to \(B\) in the permanent adop-
tion model. For a player \(v \in V\), we denote by \(C(v, q)\) the
final number of players \(B\) in the permanent adoption model
with parameter \(q\), where the initial state consists of only \(v
playing \(B\), all other players playing \(A\). Informally, we say
that \(C(v, q)\) is the size of the cascade induced by player \(v\).

**Proposition 2.** Consider the random graph \(G(n, d)\) satis-
ifying Condition 1 with asymptotic degree distribution
\(p = (p_r)_{r=0}^\infty\), and define \(q_e\) by:

\[
q_e(p) = q_e = \sup \left\{ q : \sum_{2 \leq r < q^{-1}} r(r-1)p_r > \sum_{1 \leq r_1} r p_r \right\}.
\]

Let \(P^{(n)}\) be the set of pivotal players in \(G(n, d)\).

(i) For \(q < q_e\), there are constants \(0 < \gamma_q(p) \leq \delta_q(p)\)
such that w.h.p. \(\lim_{n \to \infty} \frac{\lambda(n)}{\gamma_q(p)} = \gamma_q(p)\) and for any \(v \in P^{(n)}, \lim_{n \to \infty} C(v, q) = o(n)\).

(ii) For \(q > q_e\), for an uniformly chosen player \(v\), we have
\(C(v, q) = o_p(n)\). The same result holds if \(o(n)\) players
are chosen uniformly at random.

This result is in accordance with the heuristic result of [8]
(see in particular the cascade condition Eq. 5 in [8]), and is proved in [5].

![Figure 1: q_e(λ) for Erdős-Rényi random graphs and for power law graphs (dashed curve) as a function of the average degree λ.](image-url)
neighbor. In the case of the scale free random network considered, the parameter \( q \) must be much lower and any node with no more than nine neighbors must be able to adopt \( B \) with a single adopting neighbor. This simply reflects the intuitive idea that for widespread diffusion to occur there must be a sufficient high frequency of nodes that are certain to propagate the adoption.

We also observe that in both cases, for \( q \) sufficiently low, there are two critical values for the parameter \( \lambda \), \( 1 < \lambda(q) < \lambda_s(q) \) such that a global cascade for a fixed \( q \) is only possible for \( \lambda \in (\lambda(q); \lambda_s(q)) \). The heuristic reason for these two thresholds is that a cascade can be prematurely stopped at high-degree nodes. For Erdős–Rényi random graphs, when \( 1 \leq \lambda < \lambda_s(q) \), there exists a “giant component”, i.e. a connected component containing a positive fraction of the nodes. The high-degree nodes are quite infrequent so that the diffusion should spread easily. However, for \( \lambda \) close to one, the diffusion does not branch much and progresses along a very thin tree, “almost a line”, so that its progression is stopped as soon as it encounters a high-degree node. Due to the variability of the Poisson distribution, this happens before the diffusion becomes too big for \( \lambda < \lambda_s(q) \). Nevertheless the condition \( \lambda > \lambda_s(q) \) is not sufficient for a global cascade. Global diffusion also requires that the network not be too highly connected. This is reflected by the existence of the second threshold \( \lambda_s(q) \) where a further transition occurs, now in the opposite direction. For \( \lambda > \lambda_s(q) \), the diffusion will not reach a positive fraction of the population. The intuition here is clear: the frequency of high-degree nodes is so large that diffusion cannot avoid them and typically stops there since it is unlikely that a high enough fraction of their many neighbors eventually adopts. Following [8], we say that these nodes are locally stable.

\[
\text{Figure 2: Size } s(q, \lambda) \text{ of the cascade (in percent of the total population) for Erdős–Rényi random graphs as a function of } \lambda \text{ the average degree for a fixed } q = 0.15. \text{ The lower curve gives the asymptotic fraction of pivotal players } \gamma(q, \lambda). \]

The lower curve in Figure 2 represents the number of pivotal players in an Erdős–Rényi random graph as a function of \( \lambda \) the average connectivity for \( q^{-1} = 6.666... \). By the same heuristic argument as above, we expect two phase transitions for the size of the set of pivotal players. Indeed the phase transitions occur at the same values \( \lambda_s(q) \) and \( \lambda_s(q) \) as can be seen on Figure 2 where the normalized size (i.e. fraction) \( \gamma(q, \lambda) \) of the set of pivotal players is positive only for \( \lambda \in (\lambda(q); \lambda_s(q)) \). Hence a cascade is possible if and only if there is a ‘giant’ component of pivotal players. Note also that both phase transitions for the pivotal players are continuous, in the sense that the function \( \lambda \mapsto \gamma(q, \lambda) \) is continuous. This is not the case for the second phase transition for the normalized size of the cascade given by \( s(q, \lambda) \) in Proposition 2: the function \( \lambda \mapsto s(q, \lambda) \) is continuous in \( \lambda(q) \) but not in \( \lambda_s(q) \) as depicted on Figure 2. This has important consequences: around \( \lambda(q) \) the propagation of cascades is limited by the connectivity of the network as in standard epidemic models. But around \( \lambda_s(q) \), the propagation of cascades is not limited by the connectivity but by the high-degree nodes which are locally stable.

2.3 Advertising with word of mouth communication

We consider now scenarios where \( \lambda \not\in [\lambda(q), \lambda_s(q)] \) and the initial set of adopters grows linearly with the total population \( n \).

Given a distribution \( p = (p_z)_{z \in \mathbb{N}} \), we define the functions:

\[
\begin{align*}
    h(z; \alpha, p) &:= (1 - \alpha) \sum_{s \geq -\lceil zq \rceil} p_s \sum_{r \geq -\lceil zq \rceil} r b_s r(z), \\
    g(z; \alpha, p) &:= \lambda z^2 - h(z; \alpha, p), \\
    h_1(z; \alpha, p) &:= (1 - \alpha) \sum_{s \geq -\lceil zq \rceil} p_s \sum_{r \geq -\lceil zq \rceil} b_s r(z).
\end{align*}
\]

We define

\[
\tilde{z}(\alpha, p) := \max \{ z \in [0, 1] : g(z; \alpha, p) = 0 \}.
\]

PROPOSITION 3. Consider the random graph \( G(n, d) \) for a sequence \( (d_i)_i \) satisfying Condition 1. If the size of the set of initial adopters is \( \alpha n \), then the final number of buyers is given by \( (1 - h_1(\tilde{z}, \alpha, p))n + \alpha_0(n) \) provided \( \tilde{z}(\alpha, p) = 0 \), or \( \tilde{z}(\alpha, p) \in (0, 1] \), and further \( g(\alpha, p) < 0 \) for any \( z \) in some interval \( (\varepsilon, \tilde{z}) \).

We refer to [5] for a discussion of this result.

3. REFERENCES

[3] A. Galeotti and S. Goyal. Influencing the influencers: a heuristic argument as above, we expect two phase transitions for the size of the set of pivotal players. Indeed the phase transitions occur at the same values \( \lambda_s(q) \) and \( \lambda_s(q) \) as can be seen on Figure 2 where the normalized size (i.e. fraction) \( \gamma(q, \lambda) \) of the set of pivotal players is positive only for \( \lambda \in (\lambda(q); \lambda_s(q)) \). Hence a cascade is possible if and only if there is a ‘giant’ component of pivotal players. Note also that both phase transitions for the pivotal players are continuous, in the sense that the function \( \lambda \mapsto \gamma(q, \lambda) \) is continuous. This is not the case for the second phase transition for the normalized size of the cascade given by \( s(q, \lambda) \) in Proposition 2: the function \( \lambda \mapsto s(q, \lambda) \) is continuous in \( \lambda(q) \) but not in \( \lambda_s(q) \) as depicted on Figure 2. This has important consequences: around \( \lambda(q) \) the propagation of cascades is limited by the connectivity of the network as in standard epidemic models. But around \( \lambda_s(q) \), the propagation of cascades is not limited by the connectivity but by the high-degree nodes which are locally stable.

2.3 Advertising with word of mouth communication

We consider now scenarios where \( \lambda \not\in [\lambda(q), \lambda_s(q)] \) and the initial set of adopters grows linearly with the total population \( n \).

Given a distribution \( p = (p_z)_{z \in \mathbb{N}} \), we define the functions:

\[
\begin{align*}
    h(z; \alpha, p) &:= (1 - \alpha) \sum_{s \geq -\lceil zq \rceil} p_s \sum_{r \geq -\lceil zq \rceil} r b_s r(z), \\
    g(z; \alpha, p) &:= \lambda z^2 - h(z; \alpha, p), \\
    h_1(z; \alpha, p) &:= (1 - \alpha) \sum_{s \geq -\lceil zq \rceil} p_s \sum_{r \geq -\lceil zq \rceil} b_s r(z).
\end{align*}
\]

We define

\[
\tilde{z}(\alpha, p) := \max \{ z \in [0, 1] : g(z; \alpha, p) = 0 \}.
\]

PROPOSITION 3. Consider the random graph \( G(n, d) \) for a sequence \( (d_i)_i \) satisfying Condition 1. If the size of the set of initial adopters is \( \alpha n \), then the final number of buyers is given by \( (1 - h_1(\tilde{z}, \alpha, p))n + \alpha_0(n) \) provided \( \tilde{z}(\alpha, p) = 0 \), or \( \tilde{z}(\alpha, p) \in (0, 1] \), and further \( g(\alpha, p) < 0 \) for any \( z \) in some interval \( (\varepsilon, \tilde{z}) \).

We refer to [5] for a discussion of this result.

3. REFERENCES

Search in Non-Homogenous Random Environments

[Extended Abstract]

Omer H. Abdelrahman  
Electrical & Electronic Engineering Dept.  
Imperial College, London SW7 2BT, UK  
o.abd06@imperial.ac.uk

Erol Gelenbe  
Intelligent Systems and Networks Group  
Imperial College, London SW7 2BT, UK  
e.gelenbe@imperial.ac.uk

Summary Using a mixed jump and diffusion model, we compute the time and energy needed to find an object placed at a finite distance D from a search’s initial location within an infinite non-homogenous search space, assuming that the searcher has imprecise information about where and how to search, and also that the searcher may be blocked or destroyed during the search. This problem arises in large wired or wireless networks with imprecise routing tables and packet losses [4, 8, 10, 13], in large databases with uncertain or approximately represented data such as the content of images [5, 14], and in the search by robots in hostile environments such as minefields [7].

Introduction An animal’s search for prey was modelled in [9, 12] when the predator renews its energy reserve during the search. Randomly connected finite graphs in [11] represent search in a computer network or a system of roads. In [10] it was shown that the time it takes a data packet to travel from a source to a destination node in an infinitely large and unreliable network is finite on average, if a time-out mechanism destroys the packet after a predetermined time, replacing it with a new one that starts at the source proceeding at random and independently of its predecessor. This was generalised [15] to N searchers which are simultaneously, but independently sent out in the quest for the same object. Most of the literature considers homogenous search spaces, and in this paper we develop a mixed analytical-numerical method for an infinite random non-homogenous medium that generalises the work in [15] obtaining expressions for the average time and energy that it takes the searcher to eventually find the object it is seeking. An interesting phase transition is exhibited concerning the eventual success of the search depending on the relative speed of approach of the searcher and the intensity of events which block the searcher’s progress.

The Model Although traditionally most models in computer systems and networks are discrete [3], here we consider a continuous distance Y(t) of the searcher to the object at time t ≥ 0. The searcher starts at Y(0) = D and the search ends at time T = inf{t : Y(t) = 0}. If the random variable s(t) represents the state of the searcher, s(t) ∈ {S, W, P, ...}, then s(t) ∈ S if the search is proceeding with the search and its distance from the destination is Y(t) > 0. The probability density function of Y(t) is denoted f(z, t)dz = P[z < Y(t) ≤ z + dz, s(t) = S]. s(t) ∈ W if the searcher’s life-span has ended, and so has its search. This can happen because the searcher was destroyed or became lost, and the source was informed via the time-out. After an additional exponentially distributed delay of parameter μ, meant to avoid mistakes in assuming that the searcher is “dead”, a new searcher is placed at the source and a new search immediately begins. We write W(t) = P{s(t) = W}. s(t) ∈ L if the searcher is destroyed or lost, and the search is interrupted until a new searcher can be sent out. The time spent in this state is exponentially distributed with parameter r which is the same parameter as that of the “time-out” or life-span, since the source realises that the searcher is lost or destroyed via the life-span or time-out effect. At the end of this exponentially distributed time, the searcher is handled just as if it has “died”, and we denote L(t) = P{s(t) = L}. s(t) ∈ P if the searcher has reached its destination, i.e. it has found the object it sought and the search process ends. However, as an artefact to construct an indefinitely repeating recurrent process, after one time unit the search process restarts at the source and a new searcher is sent out. We will use the notation P(t) = P{s(t) = P}. Notice that the process repeats itself indefinitely. If E[T] is the average time that it takes from any successive start of the search until the first instance when state P is reached again, and P(t) is the probability that the model we have just described is in state P at time t ≥ 0, and P = limt→∞ P(t), then P = 1/(1 + E[T]), E[T] = P−1 − 1.

During the searcher’s travel in state S while {Y(t) = z > 0} the following events can occur in the time interval [t, t + Δt]. With probability λ(z)Δt + o(Δt) the searcher is destroyed or lost, and enters state L. From that state it enters state W after an exponentially distributed delay of parameter r. With probability rΔt + o(Δt) the searcher’s life-span runs out and it enters state W. Note that 1/r is the average life-span. As indicated earlier, when it enters state W, after an additional delay of average value 1/μ, the searcher is replaced with a new one at the source. The average rate per unit time at which the searcher approaches the object being sought when it is at distance z is b(z), and the variance of the distance travelled in the interval [t, t + Δt] is denoted by...
increases when the search speed increases. When probability density function time to the object being sought. The searcher’s location

Figure 1: Average search time $E[T]$ versus $\gamma = \lambda_1 - \lambda_2$ for $S = 10 - 15$ with a step size of 1.

Figure 2: Average search time $E[T]$ versus $S$ when $\lambda_1 = 10/S$ for different values of $b_1$. The optimum protection area needed becomes smaller so that $\lambda_1$ increases when the search speed increases.

$c(z)\Delta t$ so that:

$$b(z) = \lim_{\Delta t \to 0} \frac{E[Y_{i+\Delta t} - Y_i | Y_i = z]}{\Delta t},$$

$$c(z) = \lim_{\Delta t \to 0} \frac{E[(Y_{i+\Delta t} - Y_i)^2 - (E[Y_{i+\Delta t} - Y_i])^2 | Y_i = z]}{\Delta t}$$

When $b(z) < 0$, on average the searcher gets closer over time to the object being sought. The searcher’s location probability density function $f(z, t)$ at time $t \geq 0$, satisfies the diffusion equation [1] as in models used previously for packet flow in communication traffic flow in transportation systems or packet flows in communication systems [2, 6]. The equations that $f(z, t)dz, z > 0$, and the probability

masses $L(t), W(t)$ and $P(t), t \geq 0$ satisfy are:

$$\frac{\partial f(z, t)}{\partial t} = \frac{1}{2} \frac{\partial^2 [c(z)f(z, t)]}{\partial z^2} - \frac{\partial [b(z)f(z, t)]}{\partial z} - (\lambda(z) + r)f(z, t) + [P(t) + \mu W(t)]\delta(z - D)$$

$$\frac{dL(t)}{dt} = -rL(t) + \int_0^\infty \lambda(z)f(z, t)dz$$

$$\frac{dW(t)}{dt} = -\mu W(t) + r[L(t) + \int_0^\infty f(z, t)dz]$$

$$\frac{dP(t)}{dt} = -P(t) + \lim_{z \to 0^+} \frac{1}{2} \frac{\partial [c(z)f(z, t)]}{\partial z} - b(z)f(z, t)$$

$$1 = P(t) + W(t) + L(t) + \int_0^\infty f(z, t)dz$$

where the local behaviour of the searcher is captured in the drift $b(z)$, instantaneous variance $c(z)$ as well as loss parameter $\lambda(z)$. This is equivalent to also letting the time-out parameter $r$ be location dependent because it could be included in $\lambda(z)$. We simplify the model by considering a finite unbounded number of “segments” having different parameters for the Brownian motion describing the searcher’s movement as a function of distance to the object, while within each segment the parameters are the same. The first segment is in immediate proximity of the object being sought, starting at distance $z = 0$. Each segment may have different size, and there are a total of $m < \infty$ segments. By choosing as many segments as we wish, and letting each segment be as small as needed (all segments need not be of the same length), we can approximate closely any physical situation that arises where the searcher’s motion characteristics vary over the distance of the searcher to the object being sought. This representation leads to a neat algebraic “product form” for the average search time, providing a useful analytic form that offers an intuitive representation of the analytical results. Let $0 \leq Z_k < \infty$ be the boundary between the $k$-th and $(k+1)$-th segments with $Z_0 = 0$. The last segment goes from $Z_{m-1}$ to $+\infty$, and we assume that both $m$ and $Z_{m-1}$ are finite but unbounded. Thus for greater accuracy in representing the search we can take as many segments as we wish, and they may be as small as needed, but they are all finite except the last segment. Thus for $1 \leq k \leq m$, the $k$-th segment represents the range of distances $Z_{k-1} \leq z \leq Z_k$, and let $S_k = Z_k - Z_{k-1}$ denote its size. We use $n$ to denote the segment number in which the source point of the search is located, i.e. $Z_{n-1} < D \leq Z_n$. The stationary solution of the location dependent diffusion equation for any segment $k \neq n$ is then:

$$0 = c_k \frac{d^2 f_k(z)}{dz^2} - b_k \frac{df_k(z)}{dz} - (\lambda_k + r)f_k(z)$$

while the equation for the segment where the source is located is:

$$-[P + \mu W]\delta(z - D) = c_n \frac{d^2 f_n(z)}{dz^2} - b_n \frac{df_n(z)}{dz} - (\lambda_n + r)f_n(z)$$

(2)

Main Result Let $u_k, v_k$ be the positive and negative real roots of the characteristic polynomial of the stationary diffusion equation for the $k$-th segment:

$$u_k, v_k = \frac{b_k \pm \sqrt{b_k^2 + 2c_k(\lambda_k + r)}}{c_k}$$

Figure 3: Average search time $E[T]$ versus $\rho$ when $\lambda_k = e^{1/(\rho k)}$ and $b_k = e^{-\rho/(\rho k)}$ for different values of $\rho$: $c_k = 1$, $D = 10$, $r = 0.05$, $\mu = 0.025$ and $S_k = 1$, $k < m = 20$. 
Then the total average search time obtained by solving for $P$ so that $E[T] = P^{-1} - 1$:

$$E[T] = \left(\frac{1}{r} + \frac{1}{\mu}\right) \times \left[\frac{b_2^2 + 2c_0(\lambda_n + r)}{b_1^2 + 2c_1(\lambda_1 + r)} - \frac{b_0}{b_1} \int_0^{\lambda_1} e^{u_n(S_n - 1)} \frac{d\lambda_n}{\sqrt{2\lambda_n}} \right]$$

where the remaining parameters are as follows. Let:

$$\alpha_k = \frac{c_k u_k - c_{k-1} u_{k-1}}{c_k (u_k - u_{k-1})}, \quad \beta_k = \frac{c_k u_k - c_{k+1} u_{k+1}}{c_k (u_k - u_{k+1})}$$

Then set $A_1 = 1$ and $B_1 = -1$ and for $2 \leq k \leq n$ compute:

$$\left[\frac{A_k}{B_k}\right] = \left[\begin{array}{cc} \alpha_k & \beta_k \\ 1 - \alpha_k & 1 - \beta_k \end{array}\right] \left[\begin{array}{cc} e^{u_k - 1} \lambda_{k-1} & 0 \\ 0 & e^{-u_k + 1} \lambda_{k-1} \end{array}\right] \left[\frac{A_{k-1}}{B_{k-1}}\right]$$

Then set $F_m = 0$ and $G_m = e^{v_m Z_m}$, and start another computation at $k = m + 1$ for $n \leq k \leq m + 1$ with:

$$\left[\frac{F_k}{G_k}\right] = \left[\begin{array}{cc} \alpha_k & \beta_k \\ 1 - \alpha_k & 1 - \beta_k \end{array}\right] \left[\begin{array}{cc} e^{-u_k + 1} \lambda_{k+1} & 0 \\ 0 & e^{-u_k + 1} \lambda_{k+1} \end{array}\right] \left[\frac{F_{k+1}}{G_{k+1}}\right]$$

This completes all terms in $E[T]$. The proof is omitted.

**Special cases** If the last segment $m = n$ includes the starting point $z = D$ then:

$$E[T] = \frac{r + \mu}{r \mu} \sqrt{\frac{b_2^2 + 2c_0(\lambda_n + r)}{b_1^2 + 2c_1(\lambda_1 + r)}} - \int_0^{\lambda_1} e^{u_n(D - Z_{n-1} - 1)} \frac{d\lambda_n}{\sqrt{2\lambda_n}}$$

and if the search space is homogenous $m = n = 1$ then [15]:

$$E[T] = \left(\frac{1}{r} + \frac{1}{\mu}\right) \left[\int_{Z_{k-1}}^{Z_k} e^{u_1 D} - 1 \right]$$

If the searcher consumes energy only when it is moving, and not while it is is the lost or while it is waiting to be retransmitted, then the average energy consumed is simply given by

$$E[J] = (1 + E[T]) \sum_{k=1}^{m} \int_{Z_{k-1}}^{Z_k} f_k(z) dz.$$  \hspace{1cm} (8)

**Search in a Protected Neighbourhood** Consider the case where the neighbourhood of the object being sought, up to a distance $S$, is protected by randomly located traps that destroy the searcher. In the rest of the search space accidental destruction of the searcher may occur, but at much lower rate. Thus we take $m = n = 2$, so that $E[T]$ is obtained from (7) with $\lambda_2 = \lambda$ and $\lambda_1 = \lambda + \gamma$, $\gamma > 0$. Figure 1 shows the manner in which $E[T]$ sharply increases with $\gamma$, for $S$ ranging between 10 and 15, $D = 100$, $b_2 = b_3 = 0.25$, $c_1 = c_2 = 1$, $\lambda = 0$. Also $\mu = 1/10$ and $r$ is set to the value that minimises $E[T]$ when $\gamma = 0$ and $S = 10$. Figure 2 indicates that $S$ and $\lambda$ can be chosen to maximise the protection offered to the object being sought.

**Phase Transition when Defence is More Effective than Search** The destruction of the searcher and the timeout, both relaunch the search and allow the searcher to improve its chances to attain the object. However we will see that if the object being sought is heavily defended when the searcher gets close, then the searcher may never attain it. In Figure 3 we observe that if $\log \lambda_2 = \frac{c_2}{c_1}$, as $\rho$ becomes very small, $E[T]$, and also $E[J]$ in (8) which is not shown on the graph, tend to infinity despite the fact that the search speed and its accuracy grow as the searcher approaches the object. Thus if the searcher’s speed of approach to the object grows faster than the rate at which the searcher is destroyed then both $E[T]$ and $E[J]$ remain finite or tend to zero, while in the opposite case they tend to infinity, presenting a form of phase transition.

**REFERENCES**


On Estimation Problems for the G/G/∞ Queue

Hanhua Feng, Parijat Dube, Li Zhang
IBM T. J. Watson Research Center
Yorktown Heights, NY, 10598, USA.
{hanhfeng,pdube,zhangli}@us.ibm.com

ABSTRACT
We consider estimation problems in G/G/∞ queue under incomplete information. Specifically, we are interested in scenarios where it is infeasible to track each individual job in the system and only aggregate statistics are known or observable. We first show that the minimum expected square estimator for the queue length process can be written as the sum of the minimum square estimators for an indicator function associated with each job, which is simply the survival function of the service time variable for each job. We also obtain tight lower and upper bounds on the time average of the square estimation error. Next we look at the inverse problem of estimating the service time distribution when the observed process is only the queue length process. We develop an on-line stochastic-optimization based estimation algorithm for the service time distribution and study its convergence under different parameter settings.

Keywords
G/G/∞ queue, estimation, mean square estimation error, stochastic optimization.

1. INTRODUCTION
The count process \(N(t)\) (number of busy servers or queue length process) of any infinite server queue can be constructed by just knowing the arrival process \(\{A_i\}\) and the service time process \(\{X_i\}\) of each job. The arrival process on the other hand can just be constructed by knowing \(N(t)\), as the jump epochs in \(N(t)\) correspond to arrival times of jobs. We study non-parametric estimation of statistics of different processes in a G/G/∞ queue when only partial information is available. While we make no assumptions on the arrival process, the service times are assumed to be i.i.d. Specifically, we pose two estimation questions for G/G/∞ type queue:

- Given the arrival process and the service time distribution of jobs, what is the best estimator for \(N(t)\)?
- Given only \(N(t)\), what is the best estimator for service time distribution?

Large population systems are characteristic of many real-world systems at different temporal and spatial dimensions. An important concern with such large-scale systems is that often it is infeasible to track each individual in the system or due to other concerns like privacy etc. However aggregate statistics are easier to obtain like the count process \(N(t)\) which at any time is known by just knowing the total arrivals and total departures without tracking arrival and departure time for each individual. Earlier works adopt this position, e.g., for road traffic [2] and for industrial production [6]. For a single or multiple servers with Poisson arrivals, Larson [5] considered the problem of inferring the mean waiting time and mean queue length from observed service starting and ending times of all jobs. Pickands and Stine [7] considered the problem of estimating arrival rate and service time distribution of a discrete time M/G/∞ queue using only the count process information. Bingham and Pitts [1] applied the theory developed in Grubel and Pitts [4] for estimating the service time distribution in M/G/∞ queue under three different settings for the observed process. Heavily exploiting the property of a Poisson arrival process, these existing works are not directly applicable to the G/G/∞ scenario.

2. ESTIMATOR FOR COUNT PROCESS
We first consider the problem of finding the best estimator for the count process given the arrival process and the service time distribution. Consider a G/G/∞ queue where the service time \(X\) is i.i.d. with cumulative distribution function \(F(x)\). Consider a single job arriving at time \(0\). Let \(I_X(t)\) be the indicator random variable which is 1 for \(0 \leq X < t\) and 0 otherwise. Basically \(\{I_X(t)\}\) is a stochastic process taking binary values, 1 when the job is in the queue and 0 when the job is not in the queue. Let us use a deterministic real-valued function \(\phi(t)\) as an estimator for for \(\{I_X(t)\}\). The mean of \(\{I_X(t)\}\) is given by:

\[
E[I_X(t)] = \begin{cases} \Pr[X > t], & (t \geq 0) \\ 0, & (t < 0) \end{cases} =: \tilde{F}(t),
\]

where \(\tilde{F}(t)\) is the survival function of \(X\), defined differently than the Complementary Cumulative Distribution Function (CCDF) of \(X\) for \(t < 0\). Thus knowing the survival function one can use \(\tilde{F}(t)\) as an estimator for \(I_X(t)\), \(\forall t\). It can be easily shown that \(\phi = \tilde{F}\) minimizes the expected square estimation error for \(I_X(t)\) among class of all real-valued functions. Further when \(\phi \equiv \tilde{F}\), the expected square estimation error \(\mathbb{E}[\{I_X(t) - \tilde{F}(t)\}]^2\) equals \(\tilde{F}(t) - \tilde{F}^2(t)\) which is same as the variance of \(I_X(t)\).

2.1 A Deterministic Function as Estimator for Count Process
Let \(a_i\) be the arrival time of job \(i\), and \(X_i\) be its service time. Then the number of jobs in the system can be written

\[
N(t) := \sum_{i \in I} I_{X_i}(t - a_i),
\]

where \(I\) is the set of job indices, which can be either the set of integers or its subset. Since \(X_i\) are i.i.d., we use a deterministic \(\phi(\cdot)\) to estimate \(I_{X_i}(\cdot)\) for all \(i \in I\), we have

\[
\tilde{N}(t) := \sum_{i \in I} \phi(t - a_i).
\]

Let \(\mathcal{E}(t)\) denote the expected square estimation error, i.e.,

\[
\mathcal{E}(t) = \mathbb{E} \left[ \left( N(t) - \tilde{N}(t) \right)^2 \right].
\]

Lemma 1. For any deterministic function \(\phi(t)\), \(\mathcal{E}(t)\) is minimized when \(\phi \equiv \tilde{F}\) and the minimum \(\mathcal{E}(t)\) is simply the...
sum of the variances of individual $I_{x_i}(t-a_i)$, $i \in \mathcal{I}$ i.e.,

$$E_{\text{min}}(t) := \sum_{i \in \mathcal{I}} (\bar{F}(t-a_i) - \bar{F}^2(t-a_i)).$$

(1)

**Proof.** From the definition of $N(t)$, $\bar{N}(t)$ and $E(t)$ we have:

$$E(t) = E\left[\left(\sum_{i \in \mathcal{I}} (I_{x_i}(t-a_i) - \phi(t-a_i))^2\right)^2\right]$$

$$= \sum_{i \in \mathcal{I}} \mathbb{E}[I_{x_i}(t-a_i) - \phi(t-a_i)]^2$$

$$+ \sum_{i \neq j \in \mathcal{I}} \mathbb{E}[(I_{x_i}(t-a_i) - \phi(t-a_i))(I_{x_j}(t-a_j) - \phi(t-a_j))].$$

As $I_{x_i}(t-a_i)$ is independent of $I_{x_j}(t-a_j)$ for $i \neq j$, we get

$$E(t) = \sum_{i \in \mathcal{I}} \mathbb{E}[I_{x_i}(t-a_i) - \phi(t-a_i)]^2$$

$$+ \sum_{i \neq j \in \mathcal{I}} \mathbb{E}[I_{x_i}(t-a_i) - \phi(t-a_i)]\mathbb{E}[I_{x_j}(t-a_j) - \phi(t-a_j)]$$

$$= \sum_{i \in \mathcal{I}} \text{Var}[I_{x_i}(t-a_i) - \phi(t-a_i)]$$

$$+ \left\{ \sum_{i \in \mathcal{I}} \mathbb{E}[I_{x_i}(t-a_i) - \phi(t-a_i)] \right\}^2.$$

Observe that $\text{Var}[I_{x_i}(t-a_i) - \phi(t-a_i)] = \text{Var}[I_{x_i}(t-a_i)]$. Thus we get

$$E(t) = \sum_{i \in \mathcal{I}} \left(\bar{F}(t-a_i) - \bar{F}^2(t-a_i)\right)$$

$$+ \left(\sum_{i \in \mathcal{I}} \bar{F}(t-a_i) - \phi(t-a_i)\right)^2.$$  \hspace{1cm} (2)

Clearly, this expected square error is minimized when $\bar{F} \equiv \phi$, in which case we get (1). \hfill \square

### 2.2 Time average of the square error

The time average of the square error is defined to be

$$\lim_{T \to \infty} \frac{1}{T} \int_0^T \left(\bar{N}(t) - \bar{N}(t)\right)^2 dt = E\left[\lim_{T \to \infty} \frac{1}{T} \int_0^T \left(\bar{N}(t) - \bar{N}(t)\right)^2 dt\right]$$

for a stationary queue. Let us assume that $0 \leq a_i \leq T$. We assume that the number of arrivals in any finite interval is finite and the service time distribution has finite first moment, i.e., $\int_0^T F(t) dt < \infty$. Then,

$$\frac{1}{T} \int_0^T \left(\bar{N}(t) - \bar{N}(t)\right)^2 dt = \frac{1}{T} \int_0^T E(t) dt$$

$$= \frac{1}{T} \sum_{i \in \mathcal{I}} \int_0^T \left(\bar{F}(t-a_i) - \bar{F}^2(t-a_i)\right) dt$$

$$= \frac{1}{T} \sum_{i \in \mathcal{I}} \int_{t-a_i}^T \left(\bar{F}(t) - \bar{F}^2(t)\right) dt,$$

where $|\mathcal{I}|$ is the number of arrivals within $[0,T]$. As $T \to \infty$, it approaches to

$$\lim_{T \to \infty} \frac{|\mathcal{I}|}{T} \int_0^T (\bar{F}(t) - \bar{F}^2(t)) dt = \lambda \int_0^\infty (\bar{F}(t) - \bar{F}^2(t)) dt$$

Let the long-term average of number of arrivals be $\lambda$ and define $\rho = \lambda \mathbb{E}[X]$. Since $\int_0^\infty \bar{F}(t) dt = \mathbb{E}[X]$ and $0 \leq \bar{F}^2(t) \leq \bar{F}(t)$: we have

$$0 \leq \lim_{T \to \infty} \frac{1}{T} \int_0^T \left(\bar{N}(t) - \bar{N}(t)\right)^2 dt$$

$$= \lambda \int_0^\infty (\bar{F}(t) - \bar{F}^2(t)) dt \leq \lambda \mathbb{E}[X] = \rho$$

Both bounds are tight; when $X$ is of deterministic distribution, the error is zero. We can use a $D_2$ distribution [3] to show that the upper bound is also tight. In addition, we can construct two series $D_3$ distributions of any given mean and variance, such that one series approaches to the lower bound and the other approaches to the upper bound. It is easy to deduce that if $X$ is exponentially distributed, the average square error is $\rho/2$. In other words, we can see that the average error is tightly bounded above by $\sqrt{\rho}$ and below by $0$, and exponential service time gives $\sqrt{\rho}/2$.

### 3. ESTIMATION OF THE SERVICE TIME DISTRIBUTION

Observe that the RHS in (2) is a quadratic functional of $\phi$ minimized only at $\phi \equiv \bar{F}$. Discretizing the arrival and service times, and assuming $\phi$ has a finite support, we can represent $\phi$ using a high-dimensional vector. Therefore, we can use convex optimization techniques to find the minimal solution of $\phi$ using $E$ as the objective function, if we can compute $E$ for each given $\phi$ easily. The process $N$ is fixed if the arrival process is given; however, $N$ is a stochastic process depending on both the arrival process and the service times. Let $\phi_k = \phi_k(T_k)$ be the vector entry at discrete time $k$, where $T_k$ is the length of a unit sample interval.

#### 3.1 Off-Line Estimation Algorithm

We can design a simple gradient descent method algorithm for offline estimation.

1. Choose a step size $\alpha$ and a positive number $n$.
2. Choose an initial $\phi_0 = \phi_0$.
3. At iterate $j$, first generate a fixed finite arrival sequence $\{a_1, a_2, \ldots, a_n\}$ and use this same arrival sequence to repeatedly run the queue with random service times. $E[N(t)]$ is then approximated by averaging observed $N(t)$ in these repetitions. Then the elements of the gradient $\nabla E^2(t)$ for each discrete time $t$ is given by

$$\frac{\partial E^2(t)}{\partial \phi_k} = 2 \left(\bar{N}^2(t) - \mathbb{E}[\bar{N}^2(t)]\right) \frac{\partial \bar{N}^2(t)}{\partial \phi_k}.$$

Note that $\bar{N}^2(t) = \sum \phi_i(t-a_i)$. Therefore, $\partial \bar{N}^2(t)/\partial \phi_k = 1$ only if $kT_i = (t-a_i)$ for some $i \in \mathcal{I}$, or zero otherwise.
4. Update $\phi^{(j+1)} = \alpha \nabla E^2(t)$ and go back to step 3.

The problem of the off-line problem is that we cannot restart the queue and provide a fixed arrival process in a real system. In fact, usually we have no control over the arrival process. What we get is only a simple sample path of the arrival process and the service times. Hence in the next section we provide an on-line stochastic optimization algorithm to estimate $\bar{F}$.

#### 3.2 On-Line Estimation Algorithm

The on-line algorithm is based on stochastic gradient descent algorithm. For on-line estimation we need to assume that the arrival process is a stationary process.

We assign a fixed $\phi$ to each job. As a new job comes in, it gets the most updated $\phi$. For every $m$ arrivals, we compute the gradient using the formula as in the off-line algorithm as follows:

1. First choose an initial $\phi = \phi_0$.
2. For each discrete time $t$:
   (a) Observe $N(t)$.
   (b) Compute $\bar{N}(t) = \sum \phi_i(t-a_i)$. Since we assume $\phi$ has a finite support, say $[0, s]$, then we need only to add all jobs with $a_i + s \geq t$.
   (c) Compute $\Delta N(t) = \bar{N}(t) - \bar{N}(t)$.
   (d) Compute $\nabla E(t) = \partial \bar{N}(t)/\partial \phi_k = 2\Delta N(t), \forall k$, if $kT_i = (t-a_i)$ for some $i \in \mathcal{I}$.
3. $t \leftarrow t + 1$.
4. If $(jm+1)$-th arrival comes, $\forall j$, update $\phi$: $\phi \leftarrow \phi - \alpha \nabla E$. 


(Optional) Constrain \( \phi \) so that it is a decreasing function with value between 0 and 1.

Output an average value of \( \phi \) as an approximation of \( \bar{F} \) (we can use the average of entire history of \( \phi \), or using other averaging method, e.g., exponential moving average).

(Project \( \phi \) into the space of decreasing functions between 0 and 1 before output it if Step 5 is not done.)

Go to step 2.

In step 5, optionally we can limit \( \phi \) to be a valid CCDF function for \( t \geq 0 \) by projecting \( \phi \) to the space of \( \bar{F} \) in every iteration. However, even without the projection, \( \phi \) approaches to \( \bar{F} \) in the general functional space; therefore, we can do the projection before estimating an approximation of \( F \) in Step 7. We will show that the algorithm gives a better estimation of \( \bar{F} \) if we don’t impose the constraints in Step 5 but instead use it only for output.

### 3.2.1 Simulation Study of On-Line Algorithm

![Figure 1: Error with different step sizes for exponential service time distribution case.](image1)

![Figure 2: Estimates of \( \phi \) at different iterations converge to the (unknown) uniform service time distribution.](image2)

![Figure 3: Error with different step sizes for uniform service time distribution case.](image3)

![Figure 4: Estimates of \( \phi \) at different iterations converge to the (unknown) uniform service time distribution.](image4)

Figures 1 and 2 show the simulation results for exponential service times with mean 10. We set \( T_a = 0.01, m = 2 \) and \( s = 100 \). The arrival process is a renewal process with i.i.d. inter-arrival time uniformly distributed between 0 and 2 (with mean of 1). Therefore, the expected load will be 10. Figure 1 shows the convergence of square root of time average of \( \mathcal{E} \) with different step sizes (\( \alpha = 0.001, 0.01, 0.02 \)). For \( \alpha = 0.001 \) we show both, optimization with constraints (0 \( \leq \phi \leq 1 \) and \( \phi_k \geq \phi_{k+1} \); i.e., on-line algorithm with Step 5), and without constraints. For \( \alpha = 0.01 \) and \( \alpha = 0.02 \), we show only unconstrained case. We can see that with smaller steps, the errors converge to a closer neighborhood of the optimal expected error due to our stochastic setting. Figure 2 shows the estimated \( \phi \) from the optimization algorithm, averaged over time, at \( t = \alpha_{16340} \), and \( t = \alpha_{16350} \), for \( \alpha = 0.01 \) and with unconstrained case. We can see a gradual convergence of the average \( \phi \) towards the expected uniform distribution. At \( t = \alpha_{16370} \) (not drawn in the figure), the average \( \phi \) is almost indistinguishable from the CCDF of the exponential distribution.

**REFERENCES**


Dispatching to Incentivize Fast Service in Multi-Server Queues

[Extended Abstract]

1. INTRODUCTION

As a field, queueing theory predominantly assumes that the arrival rate of jobs and the system parameters, e.g., service rates, are fixed exogenously, and then proceeds to design and analyze scheduling policies that provide efficient performance, e.g., small response time (sojourn time). However, in reality, the arrival rate and/or service rate may depend on the scheduling and, more generally, the performance of the system. For example, if arrivals are strategic then a decrease in the mean response time due to improved scheduling may result in an increase in the arrival rate.

Understanding the effect of such strategic interactions is the focus of “queueing games”, which consider the interaction of classic queueing models and game theory. Typically, research on queueing games has focused on (i) strategic arrivals, which model jobs as strategic entities with utilities that depend on the performance received in the system, and (ii) profit-maximization, which allow the system to strategically price service (usually in the presence of strategic arrivals) in order to maximize profit. Examples of (i) include [11, 6, 8] and examples of (ii) include [14, 9, 12].

In this work, we depart from the research cited above by considering a model where arrivals are not strategic, but where servers strategically choose their service rates. The motivating example for this work is call centers, where servers are people who have control over how quickly and efficiently they work. In this setting, a dispatch design that focuses only on efficiency may seek to send calls to the fastest and most efficient servers; however by doing so the dispatch policy is actually disincentivizing hard work by requiring the most effort from its best employees, which can hurt employee retention and job satisfaction. Resultantly, call center designs seek to ensure that call dispatching is “fair”, in that servers have similar amounts of idle time [3, 4, 13].

In this work, we highlight that the strategic behavior of the servers has a fundamental impact on the design of dispatch policies. To do this, we focus on a simple model, an M/M/2 queue, where each server can strategically choose its service rate so as to maximize its utility, which is taken as the sum of a decreasing function of the chosen service rate and an increasing function of idle time experienced (which depends on the choice of service rate of the other server). In this setting, we consider the design of a non-preemptive, work-conserving dispatch policy, which decides which idle server to send the next job to, and our focus is on understanding the symmetric Nash equilibria (for the service rates) that emerges. Note that we focus on symmetric equilibria due to the importance of “fairness” in settings such as call centers.

In classic queueing theory, the most commonly proposed dispatch policies for this setting include Fastest Server First (FSF), Longest Idle Server First (LISF), and Random (which sends the job to each server with equal probability). When strategic servers are not considered, FSF is the natural choice for reducing the mean response time when forced to be work-conserving (though it is not optimal in general [5, 10]). However, we prove that FSF has no symmetric equilibria when servers are strategic. Further, we prove that LISF, a commonly suggested policy for call centers due to its fairness properties, has the same, unique, symmetric equilibrium as random dispatching. Thus, when strategic servers are considered, LISF does not even do better than the most naive, random dispatcher, Random. This highlights the importance of designing dispatch policies while being aware of the incentives they create.

With this in mind, one might suggest that Slowest Server First (SSF) would be a good dispatch policy, since it incentivizes servers to work fast; however, we prove that, like FSF, SSF has no symmetric equilibrium. But, by “softening” the bias placed by SSF toward slow servers we are able to give policies that are guaranteed to have a unique symmetric equilibrium and provide mean response times that are smaller than the response time at equilibrium under LISF and Random.

A key message provided by the results in this work is that dispatch policies must carefully balance two conflicting goals in the presence of strategic servers: they must make efficient use of the service capacity (e.g., by sending work to fast servers) while still incentivizing servers to work fast (e.g., by sending work to slow servers). While these two goals are inherently in conflict, it is possible to balance them in a way that provides improved performance over Random.

2. MODEL

Our motivating example throughout this abstract is call centers. Incoming jobs (calls) are served by one of many servers (agents). In the rest of this section, we describe the model in two parts—the queueing model and the game-theoretic model.

2.1 Queueing model

We assume that jobs arrive according to a Poisson process with rate normalized to 1, into a central, First Come First Served (FCFS) queue. The job sizes are independently exponentially distributed with rate normalized to 1. We assume that there is no abandonment, that is, every arrival is eventually served. In general, there are m servers that each choose the rates at which they work on the jobs, \( \mu_i, i = 1, \ldots, m \), according to the game-theoretic model described in the next section. Let \( \mathbf{\mu} = (\mu_1, \ldots, \mu_m) \) denote the vector of chosen service rates. In this abstract, we report results only for the
case of \( m = 2 \). In other words, we have an \( M/M/2/FCFS \) queue with an infinite buffer. The response time of a job is the amount of time it spends in the system, which is the sum of its waiting time (time spent in queue) and service time (time spent while being served). The performance objective for the system is to minimize the expected response time of a job, \( E[T] \).

The focus of this work is on the dispatch policy, which assigns jobs to servers. We restrict our attention to non-preemptive and work-conserving dispatch policies that do not use job size information. At any given instant, if at least one server is idle, and the queue is non-empty, such a policy will always pick the next job in queue and assign it to one of the idle servers. So, the defining aspect of the dispatch policy is how to make the choice of which idle server gets the next job in queue.

Four commonly studied dispatch policies are FSF (Fastest Server First), SSF (Slowest Server First), Random, and LISF (Longest Idle Server First). As their names indicate, FSF assigns the next job in the queue to the fastest idle server, SSF to the slowest idle server, Random to any of the idle servers that has been idle the longest. Among these dispatch policies, FSF minimizes the expected response time. However, it was shown in [5, 10] that even when there are two servers, FSF is not an optimal dispatch policy – there are non-work-conserving policies that improve upon it. But, in [1], it was shown that as the number of servers grows large and the arrival rate approaches the service capacity in the Halfin-Whitt regime, FSF is asymptotically optimal. A drawback of FSF is that prioritizing faster servers does not distribute jobs to the faster servers could lower employee satisfaction and degrade performance [3, 4]. As such, LISF is a “fairer” policy than FSF in the sense that asymptotically, it shares the idle time among the servers in proportion to their service rates [2]. It is this “fairness” property that leads LISF to be commonly used in practice.

In addition to these dispatch policies, we study two broad classes of dispatch policies: rate-based and idle-time-based.

### 2.1.1 Rate-based dispatch policies

Let \( I(t) \) denote the set of idle servers at time \( t \). In a rate-based dispatch policy, jobs are assigned to idle servers only based on the rate vector \( \mu \), restricted to \( I(t) \). We consider a parameterized class of rate-based dispatch policies that we term \( r \)-dispatch policies (\( r \in \mathbb{R} \)). Under these policies, at time \( t \), the next job in queue is assigned to idle server \( i \in I(t) \) with probability

\[
p_i(\mu; t; r) = \frac{\mu_i^r}{\sum_{j \in I(t)} \mu_j^r},
\]

Notice that for special values of the parameter \( r \), we recover well-known policies. For example, setting \( r = 0 \) results in Random; as \( r \to \infty \), it approaches FSF; and as \( r \to -\infty \), it approaches SSF.

### 2.1.2 Idle-time-based dispatch policies

Let \( s(t) = (s_1, \ldots, s_{|I(t)|}) \) denote the ordered vector of idle servers at time \( t \), where server \( s_j \) became idle before server \( s_k \) whenever \( j < k \). Let \( P_n = \Delta([1, \ldots, n]) \) denote the set of all probability distributions over the set \( [1, \ldots, n] \). An idle-time-based dispatch policy is defined by a vector of probability distributions \( p = (p_1, \ldots, p_m) \), such that \( p_j \in P_j \), \( j = 1, \ldots, m \). Under this policy, at time \( t \), the next job in queue is assigned to idle server \( s_j \in s(t) \) with probability \( p_{|I(t)|}(j) \). Examples of idle-time-based dispatch policies include Random, LISF, and SISF (Shortest Idle Server First).

### 2.2 Game-theoretic model

The novelty of our model is its game-theoretic aspect—servers act as strategic players in a noncooperative game. Specifically, each server chooses its service rate, \( \mu_i \), from its action set, given by \( \{\mu_i^\infty\} \), where \( \mu \) is a minimum required service rate. This captures the fact that agents in a call center are expected to perform above a minimum level of efficiency, failing which they could be fired. We set \( \mu = \frac{1}{\delta} \), which is enough to ensure stability. The decisions made by the servers constitute their joint action profile, \( \mu = (\mu_1, \ldots, \mu_m) \).

To form a dispatch policy \( \Pi \), servers aim to selfishly maximize their utility functions, given by

\[
U_i(\mu; \Pi) = I_i(\mu; \Pi) - c(\mu_i),
\]

where \( I_i(\mu; \Pi) \) is the steady state fraction of time that server \( i \) is idle, \( c(\mu_i) \) is an increasing function representing the cost incurred by a server to work at rate \( \mu \). We assume that servers are homogeneous, so they all have the same cost function. Our utility function captures the fact that servers value idle time, but have fatigue.\(^2\) Note that we assume a fixed payment model where servers are paid a fixed periodic salary, and therefore the wages would just add a constant term to the utility function. We normalize the cost function, so that \( c(\mu) = 0 \). We assume that \( c \) is convex, and satisfies \( c'(\mu) < \frac{\mu}{\delta} \), and \( c''(\mu) \geq 0 \). In particular, the assumption that \( c'(\mu) < \frac{\mu}{\delta} \) ensures voluntary participation.

Our choice of solution concept for this game is Nash equilibrium, which is a vector of service rates \( \mu^* \), such that for each server \( i \), \( U_i(\mu_1, \mu^*_i; \Pi) = \max_{\mu_i \geq 0} U_i(\mu, \mu^*_i; \Pi) \).\(^3\) We restrict ourselves to symmetric Nash equilibria, since they best represent a “fair” outcome in our model with homogeneous servers. With a slight abuse of notation, for brevity, we say that \( \mu^* \) is a symmetric Nash equilibrium if \( (\mu^*, \ldots, \mu^*) \) is a Nash equilibrium.

### 3. RESULTS

Our interest in this work is on understanding how the choice of dispatch policy affects the system performance in the presence of strategic servers. We use the expected response time of a job, \( E[T] \), as symmetric equilibrium as the measure of system performance. The goal is to choose a dispatch policy that minimizes \( E[T] \) at symmetric equilibrium.

To this end, we study the following questions:

- How do well known dispatch policies like FSF, SSF, Random, LISF perform in the presence of incentives?
- What dispatch policies admit a symmetric equilibrium? Do such policies admit unique symmetric equilibria?
- How do dispatch policies compare in terms of \( E[T] \) at symmetric equilibrium?

In the rest of this section, we state our answers to these questions for the case of two servers \( (m = 2) \), and the following assumptions on the cost function \( c(\mu) \): (i) \( c(0) = 0 \), (ii) \( c'(\mu) > 0 \), (iii) \( c''(\mu) > 0 \), (iv) \( c'(\mu) < \frac{\mu}{\delta} \), and (v) \( c''(\mu) \geq 0 \). Note that we set \( \mu = \frac{1}{\delta} \).

\(^{1}\)Even though the action profile \( (\mu_1, \ldots, \mu_m) \) is admissible, it can be shown that it is never an equilibrium, so there will be no stability issues.

\(^{2}\)In general, any function that is increasing in \( I_i(\mu; \Pi) \) and decreasing in \( c(\mu_i) \) could model this behavior, for example, \( U_i(\mu; \Pi) = -(1 - I_i(\mu; \Pi))c(\mu_i) \).

\(^{3}\)\( \mu^*_i = (\mu_1, \ldots, \mu_i - 1, \mu_{i+1}, \ldots, \mu_m) \) denotes the vector of service rates of all the servers except server \( i \).
Our first result deals with the existence of symmetric equilibria, under rate-based r-dispatch policies. It asserts that, under mild assumptions on the cost function, there are policies that do not admit symmetric Nash equilibria.

**Theorem 3.1.** There exists a bounded interval for r outside of which no r-dispatch policy admits a symmetric Nash equilibrium.

Note that this result implies that well known policies FSF and SSF that have been used to construct asymptotically optimal and/or near-optimal dispatch policies in the classic setting [2] do not admit a symmetric equilibrium when incentives are considered. Intuitively, FSF does not admit a symmetric equilibrium because, given a symmetric action profile, decreasing its service rate slightly is a better response for either server, which would improve its idle time as well as cost. This dynamics would result in a progressive lowering of service rates, until there comes a point when the load becomes too high (and idle time too low), and so, working harder would significantly increase the idle time compared to the cost. A similar intuition holds for SSF as well. This highlights the importance of accounting for incentives while designing dispatch policies.

Our second result asserts that, there are some rate-based dispatch policies that admit unique symmetric equilibria.

**Theorem 3.2.** Any r-dispatch policy with \( r \in \{-2, -1, 0, 1\} \) admits a unique symmetric Nash equilibrium.

Note that this result implies that Random admits a unique symmetric Nash equilibrium. This highlights the fact that in the presence of strategic servers, Random is “better” than FSF and SSF.

Our third result relates the class of idle-time-based dispatch policies to the class of rate-based dispatch policies.

**Theorem 3.3.** All idle-time-based policies result in the same unique symmetric equilibrium as that of Random.

Note that this result implies that no idle-time-based policy can perform better than Random (which is also a rate-based r-dispatch policy with \( r = 0 \)). In particular, this highlights the fact that LIFS does no better than Random.

Our final result provides a performance comparison among rate-based dispatch policies.

**Theorem 3.4.** Any r-dispatch policy that admits a symmetric Nash equilibrium admits a unique symmetric Nash equilibrium. Further, among all such policies, \( E[T] \) at symmetric equilibrium is increasing in \( r \).

Note that this result (along with Theorem 3.2) implies that an r-dispatch policy with \( r = -2 \) outperforms Random and all idle-time-based policies. This highlights the fact that choosing a dispatch policy while paying attention to server incentives can lead to better performance.

Our results suggest that using smaller \( r \) values lead to better performance by “softening” the bias placed by SSF toward slow servers, but, beyond a certain limit, symmetric equilibria cease to exist. While \( r = -2 \) is the best performing parameter for which we could prove the existence of a symmetric equilibrium for all admissible cost functions, individual cost functions may allow for even smaller values. For example, Figure 1 shows the performance of a 2-server system, for two cost functions, \( c_1(\mu) = \frac{\mu^2}{2} - \frac{\mu}{2} \), and \( c_2(\mu) = \frac{\mu^2}{2} - \frac{1}{\mu} \). In both cases, we see that moving \( r \) down to -10 still yields a symmetric equilibrium.

### 4. FINAL REMARKS

This abstract summarizes a first step toward understanding the interaction of strategic servers and dispatch policy design. We are working to extend the work in many directions. Most obviously, it would be interesting to extend our results to \( m > 2 \) servers. Other promising directions include exploring heterogeneous agents, asymmetric information, alternate payment models, alternate utility functions, more general queueing models, and broader classes of dispatch policies.

### 5. REFERENCES


The Power of Partial Power of Two Choices.

[Extended Abstract]†

Osman T. Akgun
University of California
Berkeley
4141 Etcheverry Hall,
Berkeley, California
94720-1777
akguno@ieor.berkeley.edu

Rhonda Righter
University of California
Berkeley
4141 Etcheverry Hall,
Berkeley, California
94720-1777
rrighter@ieor.berkeley.edu

Ronald Wolff
University of California
Berkeley
4141 Etcheverry Hall,
Berkeley, California
94720-1777
wolff@ieor.berkeley.edu

ABSTRACT

It is well known that the expected waiting time for customers routed to several parallel queues decreases dramatically when customers are routed to the shortest of two randomly chosen queues, rather than being arbitrarily assigned to one of the queues, and that the further improvement when there are three queues to choose from is much less than the improvement when moving from one to two queues (the power of two [5]). We consider the power of two effect when a subset of customers are flexible, and can choose the shortest of two queues, while the remainder are dedicated, and have no routing choice. We show that the stationary expected waiting time is decreasing and convex in the proportion of flexible customers. Our results show that having a small proportion of flexible customers has nearly as much benefit as having full power of two choices.

1. INTRODUCTION

Flexibility is an important tool in many practical systems for improving performance while reducing costs. One notion of partial flexibility in load balancing is power of two choices (see Mitzenmacher [5] and references therein). Here, customers randomly select two potential queues to query for queue lengths, and join the shortest of those two. It has been shown that having two choices has a large impact in reducing waiting times over being routed to a single arbitrary queue, but that more than two choices provide diminishing marginal improvements. We show that when some proportion \( p < 1 \) of customers are flexible, that is, are given a choice of two queues to join, expected waiting times are decreasing and convex in \( p \), again exhibiting decreasing marginal returns to flexibility.

Our model of partial flexibility has application to congestion dependent routing of highway traffic, multi-lingual call centers, make-to-order queues with flexible customers, as well as routing in computing and communication systems.

Our model is a two-server queueing system with
i.i.d. exponentially distributed service times for both servers. The servers follow a nonidling but otherwise arbitrary service discipline (FCFS, LCFS, etc.). Some proportion of arrivals (dedicated arrivals) are obliged to use a particular dedicated server, while others (flexible arrivals) have the ability to use either of the two servers and they join the shortest queue upon arrival. Each arrival is flexible with probability \( p \), independent of the other arrivals, and dedicated arrivals are equally likely to require either server. The overall customer arrival process is independent of the state and satisfies some additional mixing conditions so that the stationary expected waiting time is well defined.

In our earlier work we showed that routing flexible customers to the shortest queue among an arbitrary number of queues, known in the literature as the “Join the Shortest Queue (JSQ)” policy, is optimal in a very strong sense; it minimizes the queue-length vector process in a sample-path weak majorization sense [1]. This is true even within power of two choices. A consequence of this result is that the stationary waiting time is stochastically decreasing in the proportion of flexible customers, \( p \). Here we are interested in the marginal impact of customer flexibility, so we consider the convexity of waiting time in \( p \). Although convexity in \( p \) is intuitive, it does not hold in the strong sense that monotonicity holds, and it is surprisingly difficult to prove. We develop a new approach that combines marginal analysis with coupling to show convexity in the stationary mean waiting time. We consider a tagged customer in steady-state that has lowest preemptive priority relative to the other customers so that the other customers are unaffected by the tagged customer. We show that the derivative of the stationary waiting time with respect to \( p \) (the marginal value of customer flexibility) can be expressed in terms of the difference in expected waiting time between going to the long and the short queue for the tagged customer. We then show, using another coupling argument, that this difference is decreasing in \( p \). Our result also holds when each of two parallel stations may have \( c \geq 1 \) i.i.d. exponential servers, and and when service rates vary randomly as long as the rates for all servers are the same at the same time. Customers may also abandon after an exponentially distributed time.

The JSQ policy has been widely studied in the literature. Its optimality, assuming all customers are flexible, has been shown in a variety of contexts (e.g., Akgun et al. [1] and the references therein). The only work we are aware of that addresses convexity is the paper by He and Down [4]. They show that in heavy traffic with Poison arrivals and multiple parallel servers the full benefit of having some flexible customers can be achieved with an arbitrarily small proportion of customers being flexible.

2. RESULTS

Let \( W(p) \) be the stationary mean waiting time when the proportion of flexible customers is \( p \). The arrival process is state-independent, and each arrival is independently flexible with probability \( p \) and dedicated to queue \( i \) with probability \((1 - p)/2\). Flexible arrivals join the shortest queue. To guarantee that \( W(p) \) is well defined and finite, we require that the interarrival sequence, \( \{T_n, n = 0, \pm 1, \ldots\} \) be strongly mixing and stationary ergodic with \( 1/2\mu < ET \), where \( T \) is a random variable with the marginal distribution of any \( T_n \). Intuitively, in a strongly mixing sequence, the dependencies between the intervals disappear as the time between intervals goes to infinity. Under additional regularity conditions [3, Lemma 7], \( W(0) \) exists and is finite. We have that \( W(p) \) is decreasing in \( p \) [1, Corollary 9], so the stationary waiting times are finite for all \( p \).

We start with a sample path for the model with fixed \( \lambda = \frac{1}{2\mu}, \mu > \frac{1}{2}, \) and \( p \) (call this the \( p \) system) and construct a coupled sample path when the proportion of flexible customers becomes \( p + \epsilon \) (the \( p + \epsilon \) system). All arrivals and (potential) service completions occur at the same times in both systems, and if a service completion is from the shortest (longest) queue in the \( p \) system then it also is from the shortest (longest) queue in the \( p + \epsilon \) system. Let \( \{U_j; j = 0, 1, 2, \ldots\} \) be a sequence of independent and identically distributed uniform random variables on the interval \([0, 1]\) and, for the \( j^{th} \) arrival, let \( A_j = 1\{U_j \in [0, p]\} + 1\{U_j \in [0, p + \epsilon]\} \), so \( A_j \) represents the number of flexible customers for the \( j^{th} \) arrival in both systems. Thus, \( A_j = 2 \) if the arrival is flexible in both systems; \( A_j = 0 \) if it is dedicated in both (we call both of these “regular” customers), and \( A_j = 1 \) if the arrival is an “extra” arrival, i.e., it is flexible for the \( p + \epsilon \) system but not for the \( p \) system. When \( A_j = 0 \), if the dedicated customer goes to the shortest (longest) queue in the \( p \) system, then it does the same in the \( p + \epsilon \) system.
Rather than strict FCFS, we consider the following alternative service discipline that will make only extra customers experience the difference in waiting times for the two systems, i.e., it will maintain the same waiting times for regular customers in both systems. The extra customers in both systems have lowest preemptive priority, i.e., they are always at the back of whichever queue they join, and, among extra customers, the priority is LCFS-PR. Among regular customers the discipline is FCFS. We also have the following switching rule. Suppose a flexible regular customer ($A_j = 2$) joins the shorter queue (just observing the total queue lengths) and realizes that it would have been better off if it had joined the longer queue as it could have preempted the extra customers in the longer queue (i.e., the longer queue has fewer regular customers). Then we switch this customer with the first extra customer in the other queue. This makes the routing of regular flexible customers in the $p$ and $p + \varepsilon$ systems the same without changing the queue lengths. Our alternative service discipline has the same overall mean waiting time as standard FCFS for all customers, because service times are i.i.d. and exponential, independent of the type of the customer. In [2] we show that

$$W'(p) = \frac{dW(p)}{dp} = \lim_{\varepsilon \to 0} \left( \frac{W(p + \varepsilon) - W(p)}{\varepsilon} \right) = -Y(p)/2,$$

where $Y(p)$ is the additional expected waiting time for an extra customer that goes to the longest queue (in the $p + \varepsilon$ system) rather than the shortest queue (in the $p$ system) given that no other extra customers arrive while it is in the system.

An important note here is that we may assume, without loss of generality, that the tagged customer will not be switched if it is the only extra customer in the system at a given time, by assuming that if a flexible customer sees equal queues, it will go to the queue with the tagged customer. This is the only case where a switch after choosing the shortest queue would be beneficial to the regular customer. Then, for a particular sample path in both systems, the tagged customer will remain at the same queue until either it leaves in the $p + \varepsilon$ system, or the two queue lengths, not counting the tagged customer, become the same in both systems. In the former case the waiting time is smaller in the $p + \varepsilon$ system, i.e., $Y(p) \geq 0$ and $W(p)$ is decreasing in $p$.

In the latter case the waiting time for the tagged customer in the two systems will be the same, so, for both cases $Y(p) \geq 0$ and $W(p)$ is decreasing in $p$. Note that although the overall arrival process is general, because dedicated arrivals are equally likely to join either of the two queues and service times are exponential, once the two queue lengths are equal, from that point on the two queues will be stochastically identical. Summarizing, we have the following theorem.

**Theorem 1.** $Y(p) \geq 0$ so $W'(p) = -\frac{Y(p)}{2} \leq 0$.

Note that $W'(p) \leq 0$ follows from our earlier much stronger result [1].

Now let us consider $Y(p)$, the additional stationary expected waiting time a random arrival (the tagged customer) must spend if it goes to the long queue rather than the short queue when the proportion of flexible customers is $p$, it has lowest preemptive priority, all other customers are regular customers and regular flexible customers will join the queue with the tagged customer when the queue lengths are equal.

A similar analysis, considering the impact on the tagged customer of a single customer that is flexible in one system and dedicated in the other, gives the following theorem [2].

**Theorem 2.** $Y(p)$ is decreasing in $p$, so $W(p)$ is convex in $p$.

3. **REFERENCES**


Settling for Less - A QoS Compromise Mechanism For Opportunistic Mobile Networks

Ranjan Pal  
USC, USA  
rpal@usc.edu

Sokol Kosta  
University of Rome, Sapienza  
kosta@di.uniroma1.it

Pan Hui  
T-Labs, Germany  
pan.hui@telekom.de

1. INTRODUCTION

In recent years smartphones have become increasingly popular. In April 2011, Google claimed that around 350,000 Android smartphones are being activated daily. A smartphone is a device equipped with a range of sensors, a gigahertz-range CPU, and high bandwidth wireless networking capabilities. The power and increasing prevalence of smartphones in combination with current research on opportunistic mobile networking have (1) increased the range of applications that could be supported on an opportunistic mobile network and (2) given birth to new fields of research such as mobile crowd computing [5] that are geared towards large-scale distributed computations.

An opportunistic network is created between mobile phones using local peer-to-peer connections. The nodes in such a network are mobile phones carried by human users on the move, and a link between two phones represent the fact that the corresponding phone users are within each other’s wireless communication range. Opportunistic networks are usually intermittently connected and are characterized by social-based mobility and heterogeneous contact rate. Their basic principle of operation is based on the store-and-forward strategy [2].

Keeping in mind the fact that opportunistic networks in the near future will primarily comprise of smartphones as nodes, and would be geared towards servicing numerous applications of varied QoS demands, the opportunistic network research community today still face three basic hurdles to achieving good performance on most applications. User mobility is one such hurdle. In a relatively sparse network, user mobility might lead to network disconnectivity at times, which in turn increases response time of a user application. The second hurdle is the uncertainty in the quality of the wireless transmission channel. Effects like fading, shadowing, and interference might result in data packets being lost during transmission or being transmitted at a low speeds. Finally, individual user selfishness is a psychological hurdle which users in an opportunistic network face. A mobile user would be unwilling to forward packets for someone it does not know due to (1) individual security concerns and (2) it unnecessarily expending battery power and computation resources for an application it has no relation with. Under the above mentioned hurdles, it is not guaranteed that user QoS demands could be satisfied to a certain degree at all times let alone guaranteeing complete user satisfaction. However, in practice, users are generally tolerant on accepting lesser QoS guarantees than what they demand, with the degree of tolerance varying from user to user. The latter fact has been taken into account in some sense in traditional opportunistic networks research, where the primary goal was to make sure that users can somehow get the information through data relaying without thinking of QoS. On the other hand, an opportunistic mobile network of the near future needs to focus on the user tolerance of QoS degradation in order to justify it handling varied applications of different QoS demands.

In this abstract we propose a market based mathematical framework that enables heterogenous mobile users in an opportunistic mobile network to compromise optimally and efficiently on their QoS demands in a manner such that each user is satisfied with its achieved (lesser) QoS, and at the same time the social welfare of users in the network is maximized. Our market based framework is practically implementable and is based on the concept of parameterized supply function bidding in traditional microeconomic theory [3][4]. The contribution made in this abstract is important because (1) the hurdles related to opportunistic mobile networks mentioned in the previous paragraph are not easy to get rid of in a practical sense, and as a result mobile users have to compromise with lesser QoS than they would have ideally liked (2) the mobile users would love to make sure that they can comprise in an optimal and efficient manner, given uncertain network conditions, and (3) In an opportunistic mobile network, the network conditions vary from time to time, and it may not be possible to conjure up network resources on demand to meet user QoS choices; thus there is the need of an efficient technique that matches user demand to supply rather than the other way round. Supply function bidding is one such technique specifically suited for this purpose. In the rest of the abstract we use the terms ‘mobile user’ and ‘user’ interchangeably.

2. SYSTEM MODEL

We consider a mobile network system of $N_t$ mobile users in a time slot $t$. Each time slot $t$ lies within a total time period $T$ and is of the form $[t - 1, t]$. Within each time slot the total number of mobile users is assumed to be constant. We assume that the system is geared towards executing distributed computation tasks, in addition to regular data forwarding as in an opportunistic network. Each user in a time slot could either be (a) someone initiating a computation task, (b) someone doing computations for a task.

\[\text{In general, QoS could be parameters such as response time, number of computations per unit time, allocated bandwidth, etc.}\]

\[\text{For example, the period } T \text{ could be a single day.}\]
at hand, (c) someone just relaying information, or (d) someone doing all of (a), (b), and (c). In every time slot both the task initiators and task executors register with a central market agency. The agency could either be the one who develops the framework for efficient and optimal large-scale distributed computation or a third party. The agency has two functions in every time slot: (1) to accept user QoS demands and supply functions (QoS compromise functions) from task initiators and (2) to assess the aggregate service capacity of the task executors and enable market clearing, i.e., ensure aggregate user QoS compromise equals aggregate service capacity deficit. We also assume that the agency is connected to the mobile users via a control channel for signaling purposes (e.g., via a 3G connection).

2.1 The Basic Idea in a Nutshell

In every time slot the task initiators ‘supply’ (via an iterative bidding process [4]) between themselves and the central agency) their supply functions to the central agency. A supply function is a measure of the amount of QoS a user is willing to compromise in return for a certain amount of benefit the agency would provide to the mobile user for making the compromise. The agency estimates the deficit in the aggregate service capacity (if there is any) that prevents the network from servicing ideal user QoS demands, and chooses a common benefit value that clears the market. This benefit value is passed on to all the task initiators in the time slot who in turn settle for the corresponding compromise level based on their compromise (supply) functions. In this abstract we consider two ways in which mobile users could choose their supply functions: (1) it chooses an optimal function in a ‘price anticipating’ (oligopolistic) market of task initiators and (2) it chooses an optimal function in an ‘price anticipating’ (oligopolistic) market of task initiators.

2.2 QoS Compromise Function

Let \( c_{it}(k_{it}, b_{it}) \) be the QoS compromise function for user \( i \) in time slot \( t \). We parameterize user \( i \)'s compromise function in each time slot as follows.

\[
    c_{it}(k_{it}, b_{it}) = k_{it}b_{it}, \forall t \in N_{i}^{\text{init}} \subseteq N, \tag{1}
\]

where \( N_{i}^{\text{init}} \) consists of those users in time slot \( t \) who initiate the execution of a task. The function \( c_{it}(\cdot) \) is the supply function for user \( i \) and gives the amount of QoS it is committed to compromise. In this abstract we treat QoS as the reciprocal of response time of an application initiated by a user. For example, if a user expects to ideally achieve a response time of 2 time units, its QoS metric would have a value of \( \frac{1}{2} \). However, it could compromise say a response time of 3 additional seconds in which case its achieved QoS is \( \frac{1}{3} \). Thus, it makes a compromise of \( \frac{1}{3} \) QoS units. \( k_{it} \geq 0 \) is the supply function profile [4] for user \( i \) in time slot \( t \). It is a scalar quantity that determines the supply function of a user in time slot \( t \), and is known to the central agency. \( b_{it} \) is the benefit that the central agency provides to all the task initiators.

2.3 Clearing the Market

The central agency clears the market in every time slot by solving the following equation.

\[
    \sum_{i} c_{it}(k_{it}, b_{it}) = \sum_{i} k_{it}b_{it} = d_{t}, \tag{2}
\]

where \( d_{t} \) is the aggregate service capacity deficit in time slot \( t \). Solving the latter equation we get the value of \( b_{it} \) as

\[
    b_{it}(\overline{k}) = \frac{d_{t}}{\sum_{k_{it}} k_{it}}, \forall t, \tag{3}
\]

where \( \overline{k} = (k_{1t}, k_{2t}, ..., k_{N_{i}^{\text{init}}}, t) \) is the vector of support profiles for the task initiators in time slot \( t \).

3. COMPETITIVE MARKET ANALYSIS

We consider a competitive market of task initiators where the latter are ‘benefit’ taking. Given a benefit value \( b_{it} \) in time slot \( t \), each user \( i \) maximizes its profit according to the following optimization problem.

\[
    \arg\max_{c_{it}(\cdot)} C_{it}(c_{it}(k_{it}, b_{it})) - C_{it}(c_{it}(k_{it}, b_{it})), \tag{4}
\]

where \( C_{it}(c_{it}(\cdot)) \) is the disutility or cost incurred by user \( i \) in time slot \( t \) when it compromises \( c_{it}(\cdot) \) QoS units. We assume that \( C_{it}(\cdot) \) is continuous, increasing, and strictly convex with \( C_{it}(0) = 0 \).

In every time slot \( t \), a competitive (Walrasian) equilibrium amongst the task initiators and the central agency is defined as a tuple \( \{k_{it}^{eq}, b_{it}^{eq}\}_{i \in N_{i}^{\text{init}}, t} \) that satisfies the following conditions:

\[
    C_{it}'(c_{it}(k_{it}^{eq}, b_{it}^{eq})) - b_{it}^{eq}(b_{it} - b_{it}^{eq}) \geq 0, \forall b_{it} \geq 0 \tag{4}
\]

\[
    \sum_{t} c_{it}(k_{it}^{eq}, b_{it}^{eq}) = d_{t} \tag{5}
\]

Theorem 1. There exists a competitive equilibrium in the market of task initiators in every time slot, \( t \) that maximizes the following:

\[
    \arg\max_{c_{it}(\cdot)} \sum_{t} -C_{it}(c_{it}) \tag{5}
\]

subject to \( \sum_{t} c_{it}(k_{it}^{eq}, b_{it}^{eq}) = d_{t} \).

Proof Sketch. We get the optimality conditions of the optimization problem in Theorem 1 from equations (4) and (5). The uniqueness of the optimal solution, i.e., the equilibrium solution, follows from the fact that the optimization problem and its dual are strictly convex.

Theorem Implications. The equilibrium solution maximizes the social welfare, i.e., minimizes the sum of the disutility of the task initiators, via the optimization problem in the theorem. Thus, in every time slot, the central agency is able to clear the market by enabling optimal user QoS compromises as well as by ensuring social welfare.

The Iterative Bidding Process. We provide a distributed iterative bidding scheme based on the dual gradient
algorithm in [1] that achieves the market equilibrium in each time slot \( t \).

At the \( j \)-th iteration in time slot \( t \), we execute the following steps:

1. Upon receiving benefit \( b_i(j) \) announced by the central agency, task initiator \( i \) updates its supply function profile, \( k_{it}(j) \) as

   \[
   k_{it}(j) = \left\{ \frac{(H^t_{it})^{-1}(b_i(j))}{b_i(j)} \right\}^+, \tag{6}
   \]

   and supplies it to the central agency. Here ‘+’ denotes the projection onto \( \mathbb{R}^+ \), the set of non-negative real numbers.

2. The central agency updates its benefit according to the following equation

   \[
   b_i(j + 1) = [b_i(j) - \rho(\sum_j k_{it}(j))b_i(j) - d_i]^+, \tag{7}
   \]

   and announces the new benefit to the task initiators.

The above distributed bidding process converges for small enough values of step size, \( \rho \) [1].

4. OLIGOPOLISTIC MARKET ANALYSIS

We consider an oligopolistic market of task initiators where the latter are ‘benefit’ anticipating. The initiators are strategic in the sense that they know that benefit \( b_i \) in each time slot \( t \) is computed according to equation (3) and as a result choose their supply function profile in a manner so as to maximize their net utility functions. The net utility function for each user \( i \) in time slot \( t \) is represented by \( U_{it}(k_{it}, k_{(i-1)t}) \) and is given as

\[
U_{it}(k_{it}, k_{(i-1)t}) = b_{it}(k_{it}, k_{(i-1)t}) - C_{it}(c_{it}(k_{it}, k_{(i-1)t})), \tag{8}
\]

where \( k_{(i-1)t} = (k_{1t}, \ldots, k_{(i-1)t}, k_{it+1}, \ldots, k_{|N|_{init}}) \) is the vector of supply function profile of users other than \( i \). Each user participates in a non-cooperative game of selecting \( k_{it} \)'s, with other task initiators in time slot \( t \), in order to maximize its net utility function. The intersection of the best responses of all the task initiators results in a Nash equilibrium [6].

**Lemma 1.** If \( (k_{it}^{\text{neq}}) \) is a Nash equilibrium of the non-cooperative game at time slot \( t \), then (1) \( \sum_{j \neq i} k_{jt}^{\text{neq}} > 0 \) for any \( i \in N_{\text{init}} \), (2) \( k_{it}^{\text{neq}} < K_{\text{it}}^{\text{neq}} \) for any \( i \in N_{\text{init}} \), and (3) No Nash equilibrium exists when \( |N_{\text{init}}| = 2 \), where \( K_{\text{it}}^{\text{neq}} = \sum_j k_{jt}^{\text{neq}} \).

We omit the proof of the lemma due to lack of space.

**Lemma Implications.** At Nash equilibrium in every time slot, each task initiator compromises at most \( \frac{\Delta}{2} \) amount of QoS units, and at least two task initiators are necessary to reach a Nash equilibrium.

**Theorem 2.** There exists a Nash equilibrium, \( (k_{it}^{\text{neq}}) \), in the market of task initiators in every time slot, \( t \) that maximizes the following:

\[
\arg\max_{0 \leq c_{it} < \frac{\Delta}{2t}} \sum_i -H_{it}(c_{it})
\]

subject to \( \sum c_{it} = d_i \), where

\[
H_{it}(c_{it}) = \left( 1 + \frac{c_{it}}{d_i - 2c_{it}} \right) C_{it}(c_{it}) - \int_0^{c_{it}} \frac{d_i}{(d_i - 2x_{it})^2} C_{it}(x_{it}) dx_{it} \]


**Proof Sketch.** The uniqueness of the Nash equilibrium (optimal) solution follows from the fact that the optimization problem and its dual are strictly convex.

**Theorem Implications.** The equilibrium solution maximizes the social welfare, i.e., minimizes the sum of the disutility of the task initiators, via the optimization problem in the theorem. Thus in every time slot, the central agency is able to clear the market by enabling optimal user QoS compromises, reaching a unique Nash equilibrium, as well as ensuring social welfare.

**Proposition 1.** The Nash equilibrium benefit \( b_i^{\text{neq}} \) is bounded within a factor \( 1 + \frac{\Delta}{2}^2 \) of \( b_i^\ast \), the Walrasian equilibrium benefit where \( r_t = \max_i (H_{it}^{-1}(Y_t)) \) and \( Y_t = \max_i H_{it}^0(\frac{\Delta}{2}) \).

Proposition 1 is an important result and implies that the benefit anticipating and mutually competitive nature of task initiators in an oligopoly market leads to the Nash equilibrium benefit being bounded by the Walrasian equilibrium benefit as Walrasian markets are benefit taking. We omit the proof of the proposition due to lack of space.

**The Iterative Bidding Process.** At the \( j \)-th iteration in time slot \( t \), we execute the following steps:

1. Upon receiving benefit \( b_i(j) \) announced by the central agency, task initiator \( i \) updates its supply function profile, \( k_{it}(j) \) as

   \[
   k_{it}(j) = \left\{ \frac{(H^t_{it})^{-1}(b_i(j))}{b_i(j)} \right\}^+, \tag{9}
   \]

   and supplies it to the central agency.

2. The central agency updates its benefit according to the following equation

   \[
   b_i(j + 1) = [b_i(j) - \rho(\sum_j k_{it}(j))b_i(j) - d_i]^+, \tag{10}
   \]

   and announces the new benefit to the task initiators.

5. CONCLUSION

In this abstract we studied ways to optimally and efficiently entail user QoS compromises in opportunistic mobile networks via market mechanisms. As part of future work, we plan to conduct a simulation study of our proposed theory, and extend our theory to include different forms of parameterized supply functions.

6. REFERENCES

1. INTRODUCTION

Network (positive) externality describes the phenomenon that when people align their behaviors with others, they can incur an explicit benefit [2]. A good example of network externality is the adoption of new technologies. For example, when companies introduce smartphone to the market, the benefit of smartphone to people is determined by how many people are using it. Network externality has been extensively studied. However, many previous results, e.g., [2] only focus on the impact of population size on network externality and only capture the influence of population size on people’s willingness to adopt to a new technology. They did not consider the effect of heterogeneity, for example, different people can cast different influence on others and also different people can be influenced differently. In [3], the authors consider the influence of network externality on security measures deployment. We further enhance and generalize the work by considering the impact of node heterogeneity and differentiation.

2. MODEL

In this work, we present the mathematical model on how security protection can limit the spread of virus. In particular, we include differential treatment of different types of nodes and consider the impact of network externality in our evaluation. Our work is mainly based on the model proposed in [3]. The model includes two parts: the epidemic model and the economic model. The epidemic model is used to characterize the spread of virus or malware in a network. The economic model is used to evaluate the expected payoff of nodes. Based on the epidemic and economic model, nodes can determine whether to invest in security protection or not by evaluating their expected payoff.

Epidemic Model: Let $G = (V, E)$ be an undirected graph with vertex set $V$ and edge set $E$. For $i, j \in V$, if $(i, j) \in E$, then nodes $i$ and $j$ are neighbors and we use $i \sim j$ to denote this relationship. Let $X = \{\text{healthy, infected}\}$ represent the set of states each node can be in. If node $i$ is infected (healthy), then $X_i = 1$ ($X_i = 0$). Each infected node can contaminate its neighbors independently with probability $q$. Once a node is infected, it cannot recover to the healthy state. Note that this is similar to the bond percolation process [4] in which every edge is occupied with probability $q$. Each node has an initial state of being infected or not. Let us denote it by $x_i$ where $x_i = 1$ if it is initially infected and $x_i = 0$ otherwise. Hence, at the steady state, a node is infected either because it is initially infected, or it contracted virus from its infected neighboring nodes. Hence, the final state of node $i$ can be expressed in terms of the following recursive equation:

$$1 - x_i = (1 - x_i) \prod_{j \sim i} (1 - \theta_{ji} x_j) \quad \forall i \in V,$$

where $\theta_{ji}$ is a random variable indicating whether the edge $(i, j)$ is occupied or not. According to previous discussion, $\theta_{ji}$ is a Bernoulli random variable with $Pr(\theta_{ji} = 1) = q$. With Equation 1, the final probability that a node is infected can be derived by using the local mean field technique [3] given the initial probability of infection. Note that the infection can incur certain loss to a node. So a node needs to decide whether to invest in self-protection to decrease the infection probability or not by comparing the expected payoffs using the economic model, which we state below.

Economic Model: Every node $i$ has an initial wealth $w_i \in \mathbb{R}_+$. A node’s utility $u(y)$ is a function of wealth $y \in \mathbb{R}$. In our study, we consider nodes are risk averse, i.e., the utility function is strictly increasing and concave in $y$. However, for simplicity of illustration in this short paper, we assume that it is linear. If node $i$ is infected, then it will incur a loss of $l_i \in \mathbb{R}_+$. In order to reduce the probability of being infected, the node can consider some self-protection measures, such as buying anti-virus software, installing firewall etc. For simplicity of analysis, we assume that the choice of a node $i$ regarding self-protection is a binary decision: either the node invests with a cost of $c_i$, or it does not invest at all. If it decides to invest, it can still be infected with probability $p^-$. Else, it will be infected with $p^+$. Obviously we have $p^- < p^+$. We use $S$ and $N$ to denote the economic state of a node that it invests or does not invest in security protection respectively. A node makes the decision by maximizing its expected utility. In state $N$, the expected utility can be expressed as:

$$p^N u(w - l) + (1 - p^N) u(w),$$

where $p^N$ is the final probability of a node being infected when it initially did not consider security protection, and $l$ is the loss due to infection. The expected utility of a node which initially subscribed to security protection is:

$$p^S u(w - l - c) + p^S u(w - c),$$

where $p^S$ is the final probability of a node being infected when it initially subscribed some self-protection measures with cost of $c$. Note that $p^N$ and $p^S$ are functions of $p^-$ and $p^+$, as well as the infection probability $q$. They can be determined by the epidemic
model and can be derived using Equation (1) with the local mean field technique [1,3].

Each node needs to consider whether it should subscribe to some self-protection measures. The decision is based on the cost of investing in security measure, as well as the risk loss of being infected. The decision is non-trivial because one has to consider the externality effect. In general, a node needs to compare the cost on security investment and the risk. In particular, node $i$ will choose to invest in security protection if and only if

$$c_i < (p^N - p^S)l_i.$$  \hfill (4)

In [3], the authors analyzed the case of homogeneous self-protection cost and risk loss, i.e., nodes, independent of their connectivity, will have the same investment cost and risk loss. However, this is not reasonable in practice, since many nodes have different risk loss. For example, nodes with low degree, representing individual users, have low risk loss, while nodes with high degree, representing large companies, have high cost and security risk if they are crippled by virus. Nodes with high degree are those who have high level of interaction with other nodes. Also, their decisions can have higher influence on others than those nodes with low degrees. Thus, nodes need to be differentiated according to their degree to represent different kinds of users in a network. It is important to incorporate node heterogeneity in the model and analyze the effect. It can also help us to understand the low level of self-protection measure adoption in real life and also different levels of adoption extent in different social classes.

In the following, we consider a Erdős–Rényi random graph $G(n, p)$ with $n$ nodes, where $p = \lambda/n$ is the probability that every possible node pair $(i, j)$, $1 \leq i < j \leq n$, is connected. In the limit of large $n$, the degree of nodes in the random graph follows the Poisson distribution, i.e., $p_k = e^{-\lambda k}/k!$. All results below can be extended to random graphs with general degree distribution [4].

3. DIFFERENTIAL TREATMENT: TWO TYPES CASE

Let us classify nodes into two types according to their degree. Let $k_i$ denote the degree of node $i$. We define a degree threshold $K$. If $k_i \leq K$, then the cost of self-protection is $c_1$ and the loss due to being infected is $l_1$. On the other hand, if $k_i > K$, the cost of self-protection is $c_2$ and the loss due to being infected is $l_2$. It is reasonable to assume that $c_1 \leq c_2$ and $l_1 \leq l_2$. The initial probability of being infected is determined by economic state: $p^-$ for $S$ and $p^+$ for $N$. All edges have the same contraction probability $q$.

Assume that initially $\gamma_1 (\gamma_2)$ fraction of the nodes with degree $k \leq K$ ($k > K$) will invest in self-protection. Using the local mean field technique [1,3], we can calculate the average final probability of nodes being infected, which we denote by $h$.

**Proposition 1.** If $\gamma_1$ fraction of nodes with degree $k \leq K$ and $\gamma_2$ fraction of the nodes with degree $k > K$ invest in self-protection, then $h$, the final average probability of nodes being infected, is given by the unique solution in [0, 1] of:

$$h = 1 - (1 - p^+)e^{-\lambda h(p^1)} - (p^+ - p^-)\sum_{k \leq K} p(k)(1 - qh)^k + \gamma_2 \sum_{k > K} p(k)(1 - qh)^k,$$

where $p(k) = e^{-\lambda k}/k!$ is the probability mass function of the degree distribution of random graph.

Let $p^N_i$ ($p^S_i$) denote the final infection probability of a node which has degree $k \leq K$ and has initiallised subscribed (not subscribed) to the self-protection mechanism. Similarly, let $p^N_S$ ($p^N_S$) denote the final infection probability of a node which has degree $k > K$ and has initially subscribed (not subscribed) to the self-protection mechanism. With Proposition 1, we can derive the following conditional probabilities:

**Corollary 1.** 1. For nodes with degree $k \leq K$,

$$p^N_i = 1 - (1 - p^+ + qh) \sum_{k \leq K} p(k)(1 - qh)^k + \gamma_2 \sum_{k > K} p(k)(1 - qh)^k,$$

$$p^S_i = 1 - (1 - p^-) \sum_{k > K} p(k)(1 - qh)^k.$$  \hfill (6)

2. For nodes with degree $k > K$, we have

$$p^N_i = 1 - (1 - p^+) \sum_{k > K} p(k)(1 - qh)^k - \gamma_2 \sum_{k \leq K} p(k)(1 - qh)^k,$$  \hfill (7)

$$p^S_i = 1 - (1 - p^-) \sum_{k > K} p(k)(1 - qh)^k.$$

$$p^N_i l_i - (p^N_i l_i + c_1) = (p^N_i - l_i p^S_i) l_i - c_1.$$  \hfill (8)

$$p^S_i l_i = (p^N_i l_i + c_1) = f_1(\gamma_1, \gamma_2).$$

Here, $f_1(\gamma_1, \gamma_2) = \gamma_1(\gamma_1 + \gamma_2)^2$ is the probability reduction for nodes being finally infected if they invest in self-protection. Similarly, for $k > K$, we have

$$p^N_i l_i - (p^N_i l_i + c_2) = (p^+ - p^-) f_2(\gamma_1, \gamma_2) l_i - c_2.$$  \hfill (9)

Each node needs to make a decision to perform self-protection or not by maximizing the expected utility. Nodes will invest in self-protection if their utility with investment is greater than that without investment, so

$$\gamma_1 = \Pr((p^N_i - p^S_i) l_i \geq c_1),$$  \hfill (10)

$$\gamma_2 = \Pr((p^N_i - p^S_i) l_i \leq c_2).$$  \hfill (11)

Note that the conditional probabilities $p^S_i$, $p^N_i$, and $p^S_i$, $p^N_i$ are functions of $\gamma_1$ and $\gamma_2$. Equations (6) to (11) form fixed point equations. By Proposition 1 and Corollary 1, we can compare the utilities to determine the fraction of users that will invest in self-protection. For $k \leq K$, we have

$$p^N_i l_i - (p^N_i l_i + c_1) = (p^N_i - l_i p^S_i) l_i - c_1.$$  \hfill (12)

Let $f_1(\gamma_1, \gamma_2) = (p^+ - p^-) \sum_{k > K} p(k)(1 - qh)^k$ (because $h$ is a function of $\gamma_1$ and $\gamma_2$), then Equation (12) becomes:

$$p^N_i l_i - (p^N_i l_i + c_1) = f_1(\gamma_1, \gamma_2) l_i - c_1.$$  \hfill (13)

Here, $f_1(\gamma_1, \gamma_2) = (p^N_i - p^S_i)$ is the probability reduction for nodes being finally infected if they invest in self-protection. Similarly, for $k > K$, we have

$$p^N_i l_i - (p^N_i l_i + c_2) = (p^+ - p^-) f_2(\gamma_1, \gamma_2) l_i - c_2.$$  \hfill (14)

$$p^N_i l_i = (p^N_i l_i + c_1) = f_1(\gamma_1, \gamma_2).$$  \hfill (15)

It is easy to verify that both $f_1(\gamma_1, \gamma_2)$ and $f_2(\gamma_1, \gamma_2)$ are increasing functions in $\gamma_1$ and $\gamma_2$, which indicates that $\gamma_1$ and $\gamma_2$ degenerate to indicator functions. In other words, either no nodes will invest in self-protection, or all of them will invest in self-protection. This also shows the effect of network externality: the value of investing in self-protection increases with the number of nodes doing the investment.

It can be shown that for any $0 \leq \gamma_1 \leq 1$ and $0 \leq \gamma_2 \leq 1$,

$$f_2(\gamma_1, \gamma_2) < f_1(\gamma_1, \gamma_2),$$  \hfill (15)

which indicates that nodes with higher degree are less sensitive to invest in self-protection. In other words, investing in self-protection will lead to lower reduction in the final infection probability for nodes with higher degree.
Nodes can determine whether to make investment or not by comparing the expected profit of investment $f_1(\gamma_1, \gamma_2) l_1$ with the cost $c_1$ for nodes with lower degrees and $f_2(\gamma_1, \gamma_2) l_2$ with $c_2$ for nodes with higher degrees. We proceed to compare $f_1(\gamma_1, \gamma_2)$ with $c_1/l_1$ and $f_2(\gamma_1, \gamma_2)$ with $c_2/l_2$. We have four cases to consider:

**Case 1:** If $f_1(0, 0) > c_1/l_1$, $f_2(0, 0) > c_2/l_2$, then there is a unique Nash equilibrium where all the nodes invest in self-protection. Even if initially none of the nodes invest in self-protection, the profit of investment exceeds the cost regardless of the degree of nodes and eventually, all nodes will purchase self-protection tools.

**Case 2:** If $f_1(0, 0) > c_1/l_1$, $f_2(0, 0) < c_2/l_2$, then all nodes with degree $k \leq K$ will invest in self-protection. This is because the profit of investment for nodes with lower degree exceeds the cost while the profit is smaller than the cost for nodes with higher degree.

- If $f_2(1, 0) > c_2/l_2$, then all nodes with degree higher than $K$ will invest in self-protection. In this case, the profit of investment for nodes with higher degree increases since nodes with lower degrees will do the investment. Hence, the investment in security by nodes with degree $k \leq K$ will incentivize nodes with degree $k > K$ to invest in self-protection.

- If $f_2(1, 0) < c_2/l_2 < f_2(1, 1)$, there exists a tipping point $\gamma^*_2$ such that $f_2(1, \gamma^*_2) = \frac{c_2}{l_2}$. This implies that if we can offer self-protection to $\gamma^*_2$ fraction of nodes with degree higher than $K$ for free, then this will incentivize all nodes with higher degrees to do the investment. The price of anarchy can be expressed as

$$\frac{\sum_{k \leq K} P_k p_k f_1(1,1) l_1 + \sum_{k > K} P_k p_k f_2(1,1) l_2}{\sum_{k \leq K} P_k p_k f_1(0,1) l_1 + \sum_{k > K} P_k p_k f_2(0,1) l_2}.$$

- If $c_2/l_2 > f_2(1, 1)$, all nodes with degree $k > K$ will not perform self-protection.

**Case 3:** If $f_1(0, 0) < c_1/l_1$, $f_2(0, 0) > c_2/l_2$, then all nodes with degree $k > K$ will take self-protection.

- If $f_1(0, 1) > c_1/l_1$, then all nodes with degree lower than $K$ will take self-protection. In this case, the investment in security by nodes with degree $k \geq K$ will incentivize nodes with degree $k \leq K$ to invest in self-protection.

- If $f_1(0, 1) < c_1/l_1 < f_1(1, 1)$, there exists a tipping point $\gamma^*_1$, such that $f_1(1, \gamma^*_1) = \frac{c_1}{l_1}$. The price of anarchy can be expressed as

$$\frac{\sum_{k \leq K} P_k p_k f_1(1,1) l_1 + \sum_{k > K} P_k p_k f_2(1,1) l_2}{\sum_{k \leq K} P_k p_k f_1(0,1) l_1 + \sum_{k > K} P_k p_k f_2(0,1) l_2}.$$

- If $c_1/l_1 > f_1(1, 1)$, all nodes with degree lower than $K$ will not take self-protection.

**Case 4:** If $f_1(0, 0) < c_1/l_1 < f_1(1, 1)$, $f_2(0, 0) < c_2/l_2 < f_2(1, 1)$, then there exists a tipping point $\gamma^*_1$ and $\gamma^*_2$. The price of anarchy can be expressed as

$$\frac{\sum_{k \leq K} P_k p_k f_1(0,0) l_1 + \sum_{k > K} P_k p_k f_2(0,0) l_2}{\sum_{k \leq K} P_k f_1(1,1) l_1 + \gamma^*_1 + \sum_{k > K} P_k f_2(1,1) l_2 + \gamma^*_2}.$$

4. **NUMERICAL RESULTS & CONCLUSION**

In this section, we present numerical results to show how various parameters may affect the adoption of security protection measures. In our experiments, we set the average degree of nodes $\lambda = 5$ and the degree threshold $K = 5$.

First, we fix the initial probability of infection $p^+$ without secure measure and study the effect of $p^-$, the validity of self-protection, on the adoptability of self-protection measures. The result is shown in Figure 1(a). We set $p^+ = 0.9$, contagion probability $q = 0.3$ and vary $p^-$ from 0.1 to 0.8. The figure shows how the reduced infection probability (thresholds) $f_1(\gamma_1, \gamma_2)$ and $f_2(\gamma_1, \gamma_2)$ change with $p^-$. From the figure, we can see $f_1(\gamma_1, \gamma_2) > f_2(\gamma_1, \gamma_2)$, which verifies previous claim. $f_1(\gamma_1, \gamma_2)$ and $f_2(\gamma_1, \gamma_2)$ decrease as $p^-$ grows, which indicates that self-protection measures with higher quality imply more nodes to take the self-protection measures. This is because the network externality effect plays a small role if the self-protection quality is high. Notice that $f_1(0, 1) - f_1(0, 0)$, i.e., the gap between $f_1(0, 1)$ and $f_1(0, 0)$, is greater than $f_2(0, 1) - f_2(0, 0)$. It means that the adoption of self-protection for nodes with lower degree can incentivize higher degree nodes to invest in self-protection more than that higher degree nodes can influence those lower degree nodes. It is somewhat counter intuitive since we expect that nodes with higher degree can inflict more influence. One possible explanation is that nodes with lower degree takes a larger percentage of all the nodes, i.e., $Pr(k \leq K) > Pr(k > K)$.

In Figure 1(b), we investigate the effect of contagion probability $q$ on the thresholds. We set $p^+ = 0.4$, $p^- = 0.1$ and vary $q$ from 0.15 to 0.45. As the figure shows, the thresholds decrease as $q$ grows, i.e., a high contagion probability implies a greater network externality. When the contagion probability is high, taking self-protection will not lead to significant reduction in the final probability of being infected if no one decides to take self-protection. When contagion probability is high, people will decide to invest only if their cost and loss ratio $c/l$ is low enough. Hence, high contagion probability will inhibit nodes to take self-protection.

5. **REFERENCES**


Implications of Peer Selection Strategies by Publishers on the Performance of P2P Swarming Systems

Daniel Sadoc Menasché  
University of Massachusetts  
sadoc@cs.umass.edu

Antonio A. de A. Rocha  
Fluminense Federal University  
arocha@ic.uff.br

Edmundo A. de Souza e Silva  
Federal Univ. of Rio de Janeiro  
edmundo@land.ufrj.br

Don Towsley  
University of Massachusetts  
towsley@cs.umass.edu

Rosa M. Meri Leão  
Federal Univ. of Rio de Janeiro  
rosam@land.ufrj.br

1. INTRODUCTION

In peer-to-peer swarming systems, as peers join a swarm to download a content they bring resources such as bandwidth and memory to the system. That way, the capacity of the system increases with the arrival rate of peers. Furthermore, if publishers are intermittent, increasing the arrival rate of peers can increase content availability [7].

In the presence of stable publishers that have enough service capacity for peers to smoothly complete their downloads [6], increasing the arrival rate of peers decreases the probability that a piece will be unavailable among peers. However, if the capacity of the stable publisher, \( U \) pieces/second, is not large enough, it has been shown that the system might be unstable [3, 5, 14]. Hajek and Zhou [3, 14], following up work by Mathieu and Reynier [5], have shown that if the arrival rate of peers, \( \lambda \), is greater than \( U \), the number of peers increases unboundedly with time. It has also been shown that simple strategies can alleviate, and in some cases resolve, the instability problem. For instance, if peers reside in the system after completing their downloads, on average, the same time that they take to download a piece, then the system is always stable [14]. Nevertheless, as peers have no incentive to stay in the system after completing their downloads, it is important to investigate whether other simple strategies that do not depend on providing incentives for peers to remain online after the download completion can improve system performance and stability.

In a peer to peer system, each peer has to make two decisions before transmitting each piece: 1) which piece to transmit and 2) to whom to transmit it. Although the former question has received some attention in previous works (for instance, it has been shown that rarest-first piece selection and random useful piece selection yield the same stability region [3]), to the best of our knowledge the implications of the peer selection strategy have not been discussed yet (previous works assumed random peer selection [3, 9, 14], a notable exception being [5] – see related work section).

Let the throughput be the rate at which peers leave the system. The goal of this paper is to evaluate the impact of different peer selection strategies on the throughput (hence, stability) of the system. We pose the following questions: a) how to increase the throughput of the system by letting peers strategically select their neighbors? b) how does throughput scale with the number of peers in a closed peer-to-peer swarming system?

We provide the following answers to the above questions. First, we derive an upper bound on the throughput when the stable publisher adopts the most deprived peer selection [1] and rarest-first piece selection, while peers adopt random peer selection and random useful piece selection. The bound is significantly larger than the maximum attainable throughput when both peers and publishers adopt random peer and random useful piece selection. Then, we consider a closed system and we use a simple Markov chain model to study how the throughput of the system scales with the number of peers.

Related Work. The service capacity of peer-to-peer systems was first analyzed by Yang and de Veciana [12], who considered a closed system to analyze the transient increase in throughput after a flash crowd. They also considered an idealized fluid model to study the steady state. The fluid model was further explored by Qiu and Srikant [10], Chow et al. [2] and Zhang et al. [13]. None of these works considered the instability problem that occurs due to the fact that one piece in the system might become rare compared to the others. This problem, referred to as the missing piece syndrome, was first pointed out by Mathieu and Reynier [5]. To the best of our knowledge, [3, 14] and previous works considered only random peer selection [9], files with at most two pieces [8, 11], or considered a different class of peer-to-peer networks as those considered here [4]. Mathieu and Reynier [5] pointed out the potential advantages of most deprived peer selection, but did not pursue its in depth analysis since peers can cheat when announcing their ages. In this paper, in contrast, we analyze different peer selection strategies for peer-to-peer networks that resemble BitTorrent, but assuming peers that do not misbehave. Whereas previous work [4, 8, 9, 11], assume that peers have no information about the number of replicas of each piece in the system, in this paper, inspired by BitTorrent, we leverage the fact that most deprived peer/rarest-first piece selection are practical peer and piece selection mechanisms. As we will show next, it suffices that only publishers adopt such mechanisms in order to improve the throughput of the whole population.

2. MODEL

We consider the model presented by Hajek and Zhu [3], with an important modification: the publisher can adopt the most deprived peer policy. A file, divided into \( K \) pieces, must be distributed to peers that arrive according to a Poisson process with rate \( \lambda \). Let \( C \) be the set consisting of all subsets of \( \{1, 2, \ldots, K\} \). A type \( C \) peer is a peer that has a collection \( C \) of pieces of the file, \( C \in C \).

The publisher has service capacity \( U \) pieces/second. If
the publisher adopts random peer selection, at the end of exponentially distributed intervals with mean $1/U$ it selects a peer uniformly at random to transmit a piece. If the publisher adopts the most deprived peer selection, in contrast, and $U < \lambda$, a fraction $U/\lambda$ of peers receive a piece from the publisher after arriving to the system. These peers are referred to as gifted peers. The arrival rate of gifted peers and non-gifted peers are $U$ and $\lambda - U$, resp. (see Fig. 1).

Peers adopt the random peer, random useful piece selection.

Each peer has an internal clock, which triggers at the end of exponentially distributed intervals with mean $1/\mu$. Every time the clock triggers, the peer selects a target peer uniformly at random to transmit a piece. The piece to be transmitted is selected uniformly at random among those that the target peer does not own. Alternatively, publishers or peers can adopt the rarest-first piece selection policy, according to which they select and transmit the rarest piece among those that the target peer does not own. As soon as peers complete their downloads, they leave the system.

Let $n_C$ denote the number of peers of type $C$. The system described above can be modeled using a Markov Chain with state space $n = (n_C : C \in C)$. Let $e_C$ be a vector of the same length as $n$, with all its elements equal to zero, except the one corresponding to $C$, which equals one. Let $T_{C,n}(\mu)$ denote the state resulting from a peer of type $C$ downloading piece $i$. Since the peer-to-peer system is known to be stable when $\lambda < U$ [3], henceforth we assume $\lambda > U$.

If the publisher adopts the most deprived peer selection and rarest-first piece selection, whereas peers adopt random peer, random useful piece selection, the positive entries of the generator matrix $Q = (q(n, n')) : n, n' \in N^{|C|}$ are

\[ q(n, n + e_C) = U, \] \[ q(n, n + e_g) = \lambda - U \] \[ q(n, T_{C,n}(\mu)) = n_C \left( \mu \sum_{S \in S} n_S /((S - C)) \right) / |n| \] (3)

Eqs. (1) and (2) characterize the arrival rate of gifted and non-gifted peers, respectively. Eq. (3) characterizes piece transmissions between peers adopting random peer and random useful piece selection.

If the publisher and peers adopt random peer, random useful piece selection, the positive entries of the generator matrix $Q$ are given in [14, [11]]. To simplify presentation, in what follows we assume peers adopt random useful piece selection.

3. MOST DEPRIVED PEERS FIRST

In this section we study the system throughput when the publisher adopts the most deprived peer selection strategy and rarest-first piece selection, whereas peers adopt random peer, random useful piece selection.

Figure 1: Piece exchanges among peers.

Figure 2: The effect of the publisher strategy.
the system is unstable if $\lambda > U$.

**Random Peer Selection Versus Random Useful Peer Selection.** Since peers adopt random peer selection, the rate at which a tagged peer $A$ contacts a tagged peer $B$ for transmissions is $\mu/n$. Therefore, each peer is contacted by the rest of the population roughly at rate $\mu$. Note that if there is a large number of peers that have all pieces except a tagged one, most of the contact opportunities will occur among the one-club members, and will consist of useless contacts. In the sequel, we discuss how downlink constraints and reciprocity affect the system throughput when peers adopt random useful peer selection.

**Reciprocity.** The lifetime of peers that receive pieces from the publisher equals $(K - 1)/\mu$. As discussed in the previous paragraph, the use of random peer selection by the peers, which yields such an expected lifetime, is key. Alternatively, other factors can also yield such an expected lifetime. One factor is the limited download capacity of peers. A second factor is the reciprocity that occurs among peers in most peer-to-peer swarming systems. If non-gifted peers, when transmitting content to gifted peers, adopt a tit-for-tat strategy, according to which a peer $A$ only transmits a piece to peer $B$ if peer $B$ transmits a piece to peer $A$, the mean lifetime of gifted peers will be at least equal to $(K - 1)/\mu$, independently of the peer selection strategy adopted. That is because gifted peers can only transmit packets at rate $\mu$, hence receive packets at that rate from one-club members. Note that, to bootstrap peers that do not receive pieces from the publisher, one-club peers still need to optimistically send pieces to resource-less peers.

**Throughput Scaling.** We now study how the throughput of the system scales with the number of peers. To this goal, we consider a Markovian model of a closed system: every time a peer leaves a new one immediately arrives. A detailed description of the Markov Chain and the corresponding Tangram II model are available at http://www-net.cs.umass.edu/~sadoc/p2pthrh/. For a file consisting of two blocks $(K = 2)$ and $\mu = 1$, Figure 3 plots the throughput as a function of the population size, for different publisher capacities $U$ (varied between 0.5 and 1 blocks/s) and publisher strategies. Peers follow random peer, random useful piece selection. Figure 3 shows that the throughput obtained when publishers adopt rarest piece/most deprived peer selection is greater than the throughput obtained with each of the other two strategies. It also shows that for large population sizes, the throughput of rarest first/random peer and random useful piece/random peer are roughly the same.

4. CONCLUSION

During the past decade, peer-to-peer systems have received considerable attention for their popularity and scalability. Nonetheless, it has been recently shown that such systems are not always stable [3,5,14]. In this paper we considered publishers that adopt the most deprived peer/rarest piece selection. First, we presented a bound on the achievable system throughput. The bound is proportional to the number of pieces in the file and simulations provide evidence that it is achievable in practice. Second, we presented numerical results obtained with a Markovian model of a closed system. These results also indicate considerable gains when publishers adopt the most deprived peer/rarest piece selection mechanism.

5. REFERENCES

1. INTRODUCTION

In the past ten years, peer-to-peer (P2P) networks have challenged the traditional client/server networking paradigm. From the performance point of view, the salient feature of P2P networks is their scalability, which originates from the fundamental idea that the peers not only act as clients downloading content from other peers but also serve the other peers by uploading the downloaded contents.

Qiu and Srikant [7] developed a deterministic fluid model to analyze the performance of BitTorrent-like P2P file sharing systems under a steady flow arrival scenario. Among the key parameters are the arrival rate of new peers, $\lambda$, the efficiency of P2P file sharing, $\eta$, and the abort rate of leechers, $\theta$. For the file sharing application, it is generally accepted that $\eta \approx 1$ whenever the number of pieces is sufficiently high. More fine-grained models for P2P file sharing systems have later been developed, e.g., in [3, 2, 5], where the models are analyzed from the stability point of view. Another approach is presented in [4], which uses $\text{M}/\text{G}/\infty$ queues to model the self-scaling property of P2P file sharing systems.

Inspired by [7], Parvez et al. [6] developed a fluid model to analyze the performance of P2P video-on-demand systems. In [1], we presented a slightly different model. An essential difference between our model and that of Parvez et al. is in the handling of the playback phase and the modeling of selfishness. The model in [6] ignores the playback phase and simply assumes that any seed, whether it has played back the media file or not, departs with a constant rate. Instead of that, we include the playback phase explicitly in [1].

While in the context of file sharing a peer may download the pieces in any order, the pieces of a video file have to be retrieved (almost) in their sequential order to enable the on-line playback as required by the video-on-demand application. Therefore, it is clear that $\eta < 1$ in this case. One of our main conclusions in [1] was that the system operates properly (guaranteeing sufficient playback quality) whenever the efficiency parameter $\eta$ exceeds the following threshold:

$$\eta > \frac{1}{z} \frac{1}{\mu - \lambda},$$

where $z$ refers to the playback time of the video, $k$ to the number of permanent seeds, and $\mu$ to the upload rate of a single peer (in file transfers per time unit).

Previous P2P video-on-demand models [6, 1], however, omit the fact that the leechers may become impatient during the transfer phase and leave the system before the video file is completely downloaded. In this paper, we study how impatience affects the performance and scalability of BitTorrent-like P2P video-on-demand systems. We find out that a different approach is needed for modeling. Instead of a deterministic fluid model, we propose and develop an approximative stochastic queueing model describing the steady-state behavior of the system, the accuracy of which is verified by simulations. Based on this model, we come, maybe a bit surprisingly, to a conclusion that the most stringent conditions concerning the playback quality are related to the case with the least amount of impatience.

2. MODEL AND ANALYSIS

Let $m$ denote the size of the video file to be shared (in bits). The video is played back at a constant rate $w$. New peers arrive at rate $\lambda$. Each peer is connected to the network over an asymmetric access link with download capacity $d$ and upload capacity $u$, where $d > u$. The corresponding download and upload rates are: $c = d/m$ and $\mu = u/m$. As justified in [1], we assume that $d > w$.

The life span of a peer consists of two phases, the file transfer phase and the playback phase, which are overlapping. The video transfer and playback proceed in a parallel manner until the entire video is transferred. The transfer rate should be greater than the playback rate so that the video can be played back without any breaks or delays. In such a case, the playback phase extends beyond the transfer phase.

During the transfer phase, the peer is called a leecher. According to the fundamental P2P principle, leechers help each other. Let $\eta \in [0, 1]$ denote the efficiency of this operation. An altruistic leecher becomes a seed as soon as its own video file transfer is completed. Let $\zeta \in [0, 1]$ denote the fraction of altruistic peers, which continue to upload to leechers even after their own transfer phase. Non-altruistic peers are assumed to leave the system immediately after the transfer.
phase, while altruistic peers stay in the system until the end of the playback phase but no longer, which is a conservative assumption. If the video is played back without any breaks or delays, the length of the playback phase equals \( z = m/w \).

In addition to dynamic seeds, we allow permanent seeds, the number of which is denoted by \( k \).

**Fluid model:** As a new parameter (when compared to [1]), we introduce the abort rate of leechers, \( \theta \in [0, \infty] \). A rather straightforward extension of our previous deterministic fluid case with

\[
\begin{align*}
\dot{x}(t) &= \lambda - \theta x(t) - y(t), \\
\dot{y}(t) &= \zeta y(t) - \frac{y(t)}{\nu} - \frac{y(t)}{\zeta}.
\end{align*}
\]

(2)

where \( x(t) \) and \( y(t) \) denote the number of leechers and (non-permanent) seeds, respectively, at time \( t \). However, as we demonstrate in our numerical examples, the fluid model does not describe the steady-state behavior of the system with satisfactory accuracy for \( \theta > 0 \). Therefore, we have to take a new approach for modeling.

**Stochastic approach:** Below we develop an approximative stochastic queueing model to describe the steady-state behavior of the system more precisely. For that, we assume that new peers arrive according to a Poisson process with rate \( \lambda \), and leechers leave the system before the transfer completion randomly at rate \( \theta \). A leecher that completes the transfer phase leaves the system immediately thereafter with probability \( 1 - \zeta \). Otherwise it stays in the system as a seed until the end of the playback phase.

First we consider the hypothetical pure download constrained case with \( \mu \to \infty \). In this case, the time \( S_z \) that a peer is a leecher is clearly distributed as \( \min\{A, 1/c\} \), where \( A \) is an independent exponential random variable with mean \( 1/\theta \). It follows that the number of leechers, \( X(t) \), behaves like the number of customers in an \( M/G/\infty \) queue (cf. [4]) with steady-state mean value

\[
x_d := E[X] = \lambda E[S_z] = \frac{\lambda}{\theta}(1 - e^{-\theta/c}).
\]

(3)

In addition, we observe that new seeds constitute a Poisson process with rate proportional to \( \lambda P\{A > 1/c\} = \lambda e^{-\theta/c}/\zeta \), and the length of the remaining playback phase after the completion of the video file transfer is exactly \( z - 1/c \). Thus, the number of (non-permanent) seeds, \( Y(t) \), behaves like the number of customers in another \( M/G/\infty \) queue with steady-state mean value

\[
y_d := E[Y] = \lambda e^{-\theta/c}/\zeta(z - 1/c).
\]

(4)

Consider now the other extreme, i.e., the pure upload constrained case with \( c \to \infty \). Here we have to make a further simplification by assuming that the random variation of the effective transfer capacity per leecher, \( \mu(\eta + (Y(t) + k)/X(t)) \), can be neglected and its expectation can be approximated by

\[
\tilde{\mu} := \mu(\eta + (E[Y] + k)/E[X]),
\]

(5)

where the steady-state mean values \( E[X] \) and \( E[Y] \) are given below in (6) and (7), respectively. Under these simplifying assumptions, we may proceed as above by just substituting \( \tilde{\mu} \) for \( \mu \) everywhere. Thus, we get the following implicit formulas for the steady-state mean values:

\[
x_u := E[X] = \frac{\lambda}{\theta}(1 - e^{-\theta/\tilde{\mu}}),
\]

(6)

\[
y_u := E[Y] = \lambda e^{-\theta/\tilde{\mu}}\zeta(z - 1/\tilde{\mu}).
\]

(7)

The explicit values for \( x_u \) and \( y_u \) can be determined numerically from equations (5), (6), and (7).

According to our numerical experiments, the candidates derived above closely approximate the corresponding mean values determined from simulations when the efficiency parameter \( \eta \) is high enough but shows a qualitatively different behavior when \( \eta \) is below a certain threshold. The critical value \( \eta_0 \) is determined by requiring that the (approximate) transfer rate in the upload constrained case equals the playback rate, i.e., \( \tilde{\mu} = 1/z \). It follows that

\[
\eta_0 = \frac{1}{z} \frac{1}{\mu} - \frac{k\theta z}{\lambda(1 - e^{-z/c})},
\]

(8)

If the efficiency parameter is below this threshold, \( \eta < \eta_0 \), the transfer rate for a leecher stays below the playback rate resulting in playback quality problems. Peers have to stay longer in the system than the actual playback time so that \( Y(t) = 0 \) and the system is upload constrained. Our observation in this case is that the number of leechers and seeds are well estimated by

\[
x_0 := x_u|_{\zeta = 0}, \quad y_0 := y_u|_{\zeta = 0} = 0.
\]

(9)

The threshold \( \eta_0 \) is monotonously decreasing with \( \theta \) implying that the most stringent conditions for the efficiency parameter \( \eta \) (concerning the playback quality) are related to the case with the least amount of impatience, \( \theta = 0 \). As a consequence, we find that the playback quality is scalable (i.e., independent of the arrival rate) whenever \( \eta > w/u \). A necessary condition for this kind of scalability is clearly \( u > w \).

3. **NUMERICAL RESULTS**

In this section, we validate the accuracy of our approximative model against simulations. Unlike the analysis, which is based on the two extreme cases with \( c \to \infty \) or \( \mu \to \infty \), simulations allow any finite values for \( c \) and \( \mu \). In the simulations, peers arrive according to a Poisson process and start downloading the video file with a given fixed size at a rate determined by dynamically evolving \( \phi(t) \) (cf. (2)) which is assumed to be evenly shared between all leechers.

We consider a scenario where the parameters correspond to a typical YouTube setting. The users are viewing a video file consisting of 800 pieces each 32kB in size. The video coding rate is \( w = 300 \) kbit/s, and thus the viewing time is \( z = 682 \) s. The upload and download bandwidths of the users are \( u = 512 \) kbit/s and \( d = 1024 \) kbit/s. Also, we assume that \( k = 1 \). New users arrive with rate \( \lambda = 0.2 \) peers/s, and leechers leave due to impatience with rate \( \theta = 0.001 \). Unless stated otherwise, \( \eta = 0.8 \), which is a reasonable estimate of the efficiency that can be achieved in the present setting by using a windowed BitTorrent protocol [1].
First we compare the fluid model and the stochastic approach against simulations. The results are given in Figure 1, which depicts as a function of time the mean number of leechers $x(t)$ and seeds $y(t)$ when the system is initially empty. In the upper panel with $\zeta = 0.9$, the system is download constrained, while in the lower panel with $\zeta = 0.3$ the system is upload constrained. As it can be seen, especially in the lower panel, the fluid model is not able to accurately characterize the steady-state performance, while the accuracy of the approximative queueing model is very good.

In Figure 2, we set $\eta = 0.4$, which is below our fundamental quality threshold (8) with the given parameters. In this case, when a leecher downloads the video, it typically takes longer than $z$ seconds and the leecher leaves the system immediately after the download. It can be observed that the steady state is well estimated by equation (9).

Finally, we study the impact of the impatience parameter $\theta$ on the steady state. We set $\eta = 0.8$ and $\zeta = 0.9$, and vary $\theta \in (0,0.01)$. The results are given in Figure 3. For small values of $\theta$, the system is initially download constrained and the values of $x$ and $y$ decrease monotonously as $\theta$ increases. However, the steady state solution switches to being upload constrained roughly at $\theta = 0.0050$ at which point there is a discontinuity also in the steady state solution. Comparing with the simulated results, we observe that the correspondence is very good when $\theta < 0.0040$ or $\theta > 0.0055$. In the middle, the simulations indicate a smoother behavior than the approximative stochastic model predicts.

In the future, we plan to compare the results against traces from a BitTorrent simulator implementing a windowing algorithm as already done for the case $\theta = 0$ in [1].

4. REFERENCES


