Distributed Computing manuscript No.
(will be inserted by the editor)

Stable and Scalable Universal Swarms

Ji Zhu · Stratis Ioannidis · Nidhi Hegde · Laurent Massoulié

Received: date / Accepted: date

Abstract Hajek and Zhu [3] recently showed that the BitTorrent protocol can become unstable when peers depart immediately after downloading all pieces of a file. In light of this result, Zhou *et al.* [16] propose bundling swarms together, allowing peers to exchange pieces across different swarms, and claim that such "universal swarms" can increase BitTorrent's stability region. In this work, we formally characterize the stability region of universal swarms and show that they indeed exhibit excellent properties. In particular, bundling allows a single seeder with limited upload capacity to serve an arbitrary number of disjoint swarms if the arrival rate of peers in each swarm is lower than the seeder upload capacity. Our result also shows that the stability region is insensitive to peers' upload capacity, piece selection policies and number of swarms.

1 Introduction

BitTorrent is one of the most popular peer-to-peer protocols, used by millions of Internet users to share files online. In simple terms, peers interested in downloading a single file from a distinguished user, termed the *seeder*, form

Ji Zhu
University of Illinois at Urbana-Champaign, 1308 W. Main St. Urbana, IL 61801
E-mail: jizhu1@illinois.edu

Stratis Ioannidis
Technicolor, 735 Emerson St., Palo Alto, CA 94301
E-mail: stratis.ioannidis@technicolor.com

Nidhi Hegde
Technicolor, 1 rue Jeanne d'Arc, Issy les Moulineaux, 92443, France
E-mail: nidhi.hegde@technicolor.com

Laurent Massoulié
MSR-INRIA, 1 rue Honor d'Estienne d'Orves, Palaiseau, 91120, France
E-mail: laurent.massoulie@inria.fr

a so-called *swarm*. Peers in a swarm exchange file *pieces* (or *chunks*) with each other. Each peer thereby acts as both a client and a server, contributing to the aggregate upload capacity of the swarm. A natural question about BitTorrent amounts to determining its *stability region*: assuming the seeder's upload capacity is U pieces per second, what is the largest arrival rate of peers λ that can be supported without the swarm growing to infinity? Intuitively, as every incoming peer increases the swarm's aggregate upload capacity, one would expect BitTorrent to support high arrival rates.

Determining the stability region of BitTorrent has been an open problem for more than ten years. It was resolved recently by Hajek and Zhu [3], who showed that the swarm remains stable if $\lambda < U$, and unstable if $\lambda > U$. In light of this, a series of recent works have focused on extending the stability region of BitTorrent [4, 10, 14, 16, 17]. Our work builds upon an approach by Zhou *et al.* [16]. The authors propose bundling multiple autonomous swarms together into a *universal swarm*: peers immediately depart upon retrieving the file they desire but, while in the system, they store and exchange pieces with peers belonging to *different swarms*. Intuitively, such inter-swarm exchanges utilize bandwidth that would otherwise remain idle. On account of this, Zhou *et al.* conjecture that a universal swarm with a single seeder has an increased stability region compared to autonomous swarms. Sharing pieces with different swarms may introduce a trade-off between the area of the stability region and the average *sojourn time*, *i.e.*, the time peers stay in the system (though the latter is finite only when the system is stable): by consuming bandwidth for pieces they are not interested in, in some scenarios peers may take longer to retrieve the file they desire.

In this paper, we establish that universal swarms have excellent stability properties. In addition, we show that they can be designed so that peers *do not* experience increased delays. In particular, we make the following contributions. First, we formally characterize the stability region of universal swarms; we derive necessary and sufficient conditions under which such swarms remain stable. Second, we show that by bundling swarms together BitTorrent's stability region increases significantly, and the system scales gracefully in the number of bundled swarms. In particular, the stability region under limited seeder upload capacity *is insensitive to the number of swarms, as well as the peers' uploading capacity and piece selection policies*. Third, the increased stability region of universal swarms comes at the cost of increased delays, which scale with the total number of bundled swarms. Finally, to address the problem of increased delays, we propose a modified system design that provably extends the stability region *and* does not affect delays.

Our stability result (Theorem 2) is general: in contrast to previous works [3, 12, 16, 17], it covers all work-conserving piece selection policies, determining which pieces peers exchange upon contact with each other. As a consequence, our result has interesting implications for a single swarm as well. Moreover, to the best of our knowledge, we are the first to observe and exploit the fact that swarms become *meta-stable* when the swarm seeder and peers prioritize rare pieces.

The remainder of the paper is organized as follows. We review related work in the next section. Our model of universal swarms is introduced in Section 3. Earlier results and our new results on stability constitute Sections 4 and 5 respectively. We present detailed proofs of our main results in Sections 6 and 7. We study sojourn time numerically in Section 8 and conclude in Section 9.

2 Related Work

The stability of BitTorrent is determined by a phenomenon called the *missing piece syndrome*, described in detail in Section 4. The first rigorous treatment of this phenomenon, and how it affects stability, was given by Hajek and Zhu [3] in the context of files comprising an arbitrary number of pieces. The missing piece syndrome was also studied independently by Norros *et al.* [12], in the case where a two-piece file is shared. Hajek and Zhu further observed that the phenomenon persists—and the stability region remains the same—when using source coding as well as when prioritizing the transmission of rare pieces.

Mathieu and Reynier [7] first observed the phenomenon of the missing piece syndrome through simulations and provide conditions for the persistence of the so-called *starvation* state. They further show advantages of selection strategies that are piece discriminant. Menasché *et al.* [8] also investigate the impact of peer and piece selection strategies. In particular they derive upper bounds on throughput when the seed adopts the most-deprived-peer selection and rarest-first piece selection while the peers adopt random peer selection and random-useful piece selection. The authors provide simulation results for the evolution of the number of peers in the system. While these papers consider the impact of selection strategies on various metrics, they do not provide a rigorous treatment of the stability region.

Menasché *et al.* [11] further consider another metric for peer-to-peer systems, self-sustainability, defined as the fraction of time the swarm is self-sustainable, where all content is collectively held by peers, that is, help from the seed is not needed. They explore how this metric varies with content popularity, service capacity of users, and size of file. Self-sustainability is tied to the notion of piece availability and aims at characterizing a notion of the dependence on a seed. As such, it doesn't capture stability, since it doesn't necessarily prevent the missing piece syndrome. This notion of self-sustainability relates to the meta-stability phenomenon we discuss in Section 8. Indeed, although prioritizing rarest pieces and favoring self-sustainability may not necessarily affect the stability region, it can make the formation of one-club, a large set of peers missing a piece, a very rare event.

One way to extend the stability region of BitTorrent is to require that peers remain in the system after they have downloaded all pieces, effectively turning from leechers to seeders. In follow-up work, Zhu and Hajek show that this indeed extends the stability region by a factor that depends on the mean additional time peers spend in the system [17]. By setting the latter to a high enough value, the system remains stable for arbitrary arrival rates. Núñez-

Queija and Prabhu [13] study the impact of this additional time on the average broadcast time, that is the time to distribute a file to all peers in a closed system, where there are no peer arrivals, only departures. They show that in a closed system of N peers, the mean broadcast time is $O(\log N)$ with altruistic nodes that stay for the duration of at least one more contact, while it is $O(N)$ otherwise, implying a phase transition phenomenon for the scaling law. Requiring that peers spend additional time in the system assumes they are altruistic. Universal swarms also presume altruism, though of a different kind: though peers immediately leave upon retrieving a file, they contribute their storage and bandwidth resources to other swarms while in the system, by storing and uploading pieces of files they may not be interested in.

Oğuz *et al.* [14] propose an alternative approach to extend BitTorrent's stability region. The authors consider schemes where (a) newly arriving peers do not accept pieces unless they are sufficiently rare, and (b) peers missing a single piece do not download it before uploading a piece that is sufficiently rare. Crucially, the rarity of a piece is estimated in a distributed fashion, through sampling a small number of peers. The resulting system is stable under arbitrary arrival rates. It also assumes peers are altruistic, as they do not download useful pieces at all opportunities to do so.

Zhou *et al.* [16] propose universal swarms as a means of extending the stability region of BitTorrent. They study stability under a model in which (a) files consist of only one piece and (b) swarms are *seedless* [6]: no seeder exists, and all peers arrive already endowed with certain pieces. The authors observe an extension of the stability region under these assumptions. Menasché *et al.* [9] also consider a limited notion of universal swarms by allowing bartering, where peers download file pieces not of interest to them, but to be used in a future exchange. Their model also doesn't include a seeder, and peers arrive endowed with pieces, but fractions of files may be exchanged. Their analysis is limited to reciprocity only through a single relay that they call indirect reciprocity. The paper focuses on analyzing the loss of efficiency, measured in terms of the number of transmissions in the system. In the present work, we consider the more realistic setup of [3, 14, 17] ; that is, peers arrive empty, while pieces are injected by a seeder, and files consist of multiple pieces. Further, we provide a rigorous treatment of the stability region of such a multi-swarm system.

Bundling across swarms has also been studied in other contexts. Menasche *et al.* [10] show that bundling swarms together improves file availability, while Wu *et al.* [15] show that bundling improves the performance of live streaming. In contrast, our work presents a rigorous analysis of the stability of universal swarms under various sharing policies. Finally, a measurement study by Han *et al.* [4] establishes the prevalence of file bundling in BitTorrent: $50 - 80\%$ of swarms contain bundled files, with content such as movies, TV series and music being bundled more often than games. These real-world measurements attest to the relevance and practical significance of studying universal swarms.

3 System Description

Consider a BitTorrent-like file-sharing system, consisting of multiple swarms. Suppose all files are divided into *pieces* of equal size. Denote by $\mathcal{F} = \{1, 2, \ldots, K\}$ the set of all pieces of all files. A distinguished peer, the seeder, is always present and holds $\mathcal{F}$. A *file* C is a non-empty subset of $\mathcal{F}$: *i.e.*, $\emptyset \neq C \subseteq \mathcal{F}$. We refer to the set of peers interested in downloading a file C as a *swarm*. Abusing notation, we sometimes refer to the swarm of peers interested in C as "swarm C" (note that $|C|$ is the number of pieces in the file requested by peers in swarm C, not the number of peers in swarm C). Peers in swarm C arrive according to a Poisson process with rate λ_C, independent across swarms. Note that different swarms are *not* required to be disjoint subsets of $\mathcal{F}$, so the model captures a scenario where arriving peers are interested in multiple files.

Each peer maintains a cache to store pieces it downloads. Assume peers arrive with empty caches, and each peer's cache is large enough to hold $\mathcal{F}$ (*i.e.*, all file pieces). Peers in swarm C depart *immediately* upon retrieving all pieces in C. Partition peers into *types* according to (a) the swarm they belong to and (b) the set of pieces in their cache. Hence, a peer in swarm C holding $S \subseteq \mathcal{F}$ is denoted to be of type $\langle C, S\rangle$. Assume the seeder is of type $\langle\{\perp\}, \mathcal{F}\rangle$ for some piece $\perp \notin \mathcal{F}$. Denote $n_{\langle C,S\rangle}$ to be the number of type $\langle C, S\rangle$ peers and $\mathbf{n} = (n_{\langle C,S\rangle})$ to be the vector of numbers of peers in all types. The seeder uploads pieces at instants of a Poisson process of rate U. At each such instant, the seeder contacts a peer selected uniformly at random among all peers across all swarms, and replicates a piece in $\mathcal{F}$ to this peer. Similarly, at instances that follow a Poisson process of rate $\mu > 0$, each peer contacts another peer (also selected uniformly among all peers) and replicates a piece from its cache.

The piece replicated when a source (either a peer or the seeder) contacts a receiver is determined by the source's piece selection policy. We consider a broad class of work-conserving piece selection policies satisfying the following:

Assumption 1 *If a source in type $\langle C, S\rangle$ contacts a receiver in type $\langle C', S'\rangle$, no piece is replicated if $S \subseteq S'$. Otherwise exactly one piece in $S \setminus S'$ is replicated, with piece $i \in S \setminus S'$ replicated with probability $h_{\langle C,S\rangle}(i, \langle C', S'\rangle, \mathbf{n}) \in [0, 1]$, determined by the types of the source and the receiver, the piece id, and the current state $\mathbf{n}$. Function $h_{\langle C,S\rangle}$, also referred to as the policy, satisfies:*

$$\sum_{i \notin S \setminus S'} h_{\langle C,S\rangle}(i, \langle C', S'\rangle, \mathbf{n}) = 0, \tag{1}$$

$$\sum_{i \in S \setminus S'} h_{\langle C,S\rangle}(i, \langle C', S'\rangle, \mathbf{n}) = 1 \text{ if } S \not\subseteq S'. \tag{2}$$

Intuitively, Equation (1) implies the source only uploads pieces in its storage the receiver does not already have. Equation (2) implies that, if the source has a piece that the receiver does not have, the source sends some piece to the receiver.

Suppose a type $\langle C, S\rangle$ source contacts a type $\langle C', S'\rangle$ receiver. A few examples of work-conserving policies satisfying Assumption 1 are as follows:

Random Novel [RN]: If $S \setminus S' \neq \emptyset$, the source replicates a piece chosen uniformly from $S \setminus S'$.
Rarest First [RF]: Define the *availability* of a piece $i \in \mathcal{F}$ to be the number of peers holding it. The source replicates the piece in $S \setminus S'$ that has the least availability, with ties broken randomly.
Priority Rarest First [PRF]: The source prioritizes pieces within the swarm of the receiver: if $(S \setminus S') \cap C' \neq \emptyset$, it replicates the piece in $(S \setminus S') \cap C'$ that has the least availability; if $(S \setminus S') \cap C'$ is empty but $S \setminus S'$ is not, the source reverts to RF. **Priority Random Novel [PRN]** can be defined similarly.

Notice that, under Assumption 1, sources of the same type apply the same policy. The piece selection policy of the system is denoted by a tuple of $h_{\langle C,S\rangle}$ indexed by each $\langle C, S\rangle$, where all sources in type $\langle C, S\rangle$ apply the policy $h_{\langle C,S\rangle}$ in the tuple. Different policies h can co-exist across types: *e.g.*, the seeder may implement a random novel policy, while peers implement priority rarest first.

Contrary to random novel, the RF and PRF policies depend on the system state $\mathbf{n}$, and require knowledge of a global property; as such, they are harder to implement in a distributed fashion. In BitTorrent implementations, peers roughly track the rarest block in a distributed fashion, monitoring their neighborhoods. Focus on the problem of how to gather information on piece distribution is beyond this paper's scope. We therefore simply assume that the system state $\mathbf{n}$ is available for all peers, through, e.g., a centralized monitor, so that peers can apply RF, PRF or other piece selection policies depending on the system state $\mathbf{n}$. It is important to keep in mind however that, in practise, such policies would need to be implemented in a distributed fashion.

We use the following additional definitions. Define

$$\lambda_{total} := \sum_{C:C\in\mathcal{C}} \lambda_C$$

to be the total arrival rate,

$$n_S := \sum_{C:C\in\mathcal{C}} n_{\langle C,S\rangle}$$

to be the number of peers holding the set of pieces S,

$$\mathcal{C} := \{C : C \in 2^{\mathcal{F}} \setminus \{\emptyset\}, \lambda_C > 0\}$$

the set of swarms,

$$\mathcal{T} := \{\langle C, S\rangle : C \in \mathcal{C}, S \in 2^{\mathcal{F}} \setminus \{\mathcal{F}\}, C \not\subseteq S\}$$

the set containing all peer types for peers in the system,

$$\tilde{\mathcal{T}} = \mathcal{T} \cup \{\langle\{\perp\}, \mathcal{F}\rangle\}$$

the extended set of peer types, and, finally, $\mathcal{D} = \mathbb{N}^{|\mathcal{T}|}$ the set of all possible vectors $\mathbf{n}$.

Our main stability result (Theorem 2) assumes that the seeder applies RN, while peers apply *any* piece selection policies satisfying Assumption 1.

3.1 Markov Process Description and Stability

The system evolution is described by a Markov process $\{\mathbf{n}(t)\}_{t\in\mathbb{R}_+}$ with state space $\mathcal{D}$, where state $\mathbf{n} = (n_{\langle C,S\rangle})$ is the vector of numbers of peers in all types, $\mathbf{n}(t)$ denotes the state $\mathbf{n}$ at time t, and $\mathcal{D}$ is the set of all possible values of state $\mathbf{n}$. The transition rates of the process depend on how pieces are uploaded.

Assume that the seeder implements RN, and peers apply policies as in Assumption 1. Given a state $\mathbf{n}$, let $T_C(\mathbf{n})$ be the new state resulting from the arrival of a new peer in swarm C. Given $\langle C,S\rangle \in \mathcal{T}$ such that $i \notin S$, and a state $\mathbf{n}$ such that $n_{\langle C,S\rangle} \geq 1$, let $T_{\langle C,S\rangle,i}(\mathbf{n})$ denote the new state resulting from a type $\langle C,S\rangle$ peer downloading piece i. The positive entries of the generator matrix $Q = (q(\mathbf{n},\mathbf{n}') : \mathbf{n},\mathbf{n}' \in \mathcal{D})$ of the Markov process $\mathbf{n}(t)$ are given by:

$$q(\mathbf{n}, T_C(\mathbf{n})) = \lambda_C$$

$$q(\mathbf{n}, T_{\langle C,S\rangle,i}(\mathbf{n})) = \frac{n_{\langle C,S\rangle}}{n}\Big[\frac{U}{K-|S|}+\mu\sum_{v\in\mathcal{T}} n_v h_v(i,\langle C,S\rangle,\mathbf{n})\Big]$$

We follow the usual definitions of stability and instability for Markov processes [2]:

Definition 1 The system is *unstable* if it is transient and the number of peers converges to infinity with probability one; and the system is *stable* if it is positive recurrent and it has a finite mean number of peers in equilibrium.

4 Single Swarm and Missing Piece Syndrome

Hajek and Zhu study the above system in the case of a single, autonomous swarm, *i.e.*, a system in which all peers are interested in downloading the file $C = \mathcal{F}$. They determine the stability region under the RN policy:

Theorem 1 (Hajek and Zhu [3]) *Consider a single swarm of peers requesting all pieces in $\mathcal{F}$, in which both the seeder and peers follow the random novel piece selection policy. The system is stable if $\lambda_{\mathcal{F}} < U$, and unstable if $\lambda_{\mathcal{F}} > U$.*

Hajek and Zhu further establish that the so-called missing piece syndrome [3, 12] is the reason of instability when $\lambda_{\mathcal{F}} > U$. This phenomenon arises when there is a large number of peers in the system that store all pieces in $\mathcal{F}$ except for one missing piece. When this set of peers, termed the *one-club*, is large enough, most of the contacts of new peers arriving in the system will be with such peers. The new peers thus quickly retrieve all pieces except the missing piece, thus joining the one-club set. Since peers holding the missing piece are few, departures from the one-club are mostly due to uploads by the seeder; as a result, the departure rate of the one-club is close to the seeder upload rate U. Since $\lambda_{\mathcal{F}} > U$, the rate of growth of peers in the one-club is positive, causing the size of this set to increase to infinity and resulting to instability.

Theorem 1 has an immediate corollary in the case of multi-swarm systems. In particular, suppose that each swarm operates in an *autonomous mode*, independently and in isolation of other swarms. More specifically, peers in swarm

$C \in \mathcal{C}$ contact and exchange pieces only with other peers in the same swarm. In addition, the seeder divides its upload capacity across different swarms (possibly unevenly), serving each with an appropriate fraction of its total capacity. Finally, pieces that peers in swarm C store and exchange are pieces in set C. Theorem 1 directly applies to each such system and, thus, it is easy to verify the following:

Corollary 1 *Consider a multi-swarm system operating in autonomous mode. The seeder can allocate its upload capacity so that the system is stable if* $\sum_{C\in\mathcal{C}} \lambda_C < U$*; the system is unstable for all allocations of the seeder's upload capacity if* $\sum_{C\in\mathcal{C}} \lambda_C > U$*.*

Note that the corollary assumes a static allocation of a seeder's rate to each swarm. Though formally studying the capacity of a dynamic allocation is beyond the scope of this work, we observe through simulations that an allocation that is proportional to the size of each swarm does not improve stability or prevent the missing piece syndrome (see Fig. 2(a)). Recall also that our results presented assume random peer selection by peers and the seeder, either over each individual swarm or over all peers in all swarms. It is possible that prioritizing contacts with certain peers can also improve stability, though the study of such a system is beyond the scope of this paper.

In what follows, we study multi-swarm systems that operate as described formally in Section 3. To distinguish this system, we refer to it as a multi-swarm system in *universal mode* or, simply, a *universal swarm.*

5 Stability Region of a Universal Swarm

The following is the main result regarding the stability region of the Markov process defined in Section 3.1:

Theorem 2 *If the seeder implements RN and peers implement work-conserving piece selection policies in Assumption 1, the system is*

i) unstable if $\max_{i:i\in\mathcal{F}} \sum_{C:i\in C} \lambda_C > U$,
ii) stable if $\max_{i:i\in\mathcal{F}} \sum_{C:i\in C} \lambda_C < U$.

Beyond considering universal swarms, Theorem 2 extends Theorem 1 to the case where peers implement arbitrary piece selection policies. In fact, an immediate corollary of Theorem 2, applied to the single swarm setup, is that the stability region of Theorem 1 extends to such policies as well. The theorem assumes that the seeder uses the RN policy; our numerical evaluations in Section 8 suggest that other seeder policies (*e.g.*, RF) also yield the same stability region.

Note that the theorem implies that bundling swarms together yields a significant increase in the stability region. Observe that, when the files $C \in \mathcal{C}$ are disjoint, Theorem 2(ii) becomes $\max_{C\in\mathcal{C}} \lambda_C < U$. This defines a larger stability region than the one of autonomous mode, given by Corollary 1. In particular, by bundling swarms together, the stability region scales extremely

well as the number of swarms increases: a single seeder can support an *unbounded number* of swarms with constant arrival rate, with no effect on the stability region! However, as discussed in Section 8, bundling swarms together comes at the cost of increased delays. Hence, the number of swarms cannot be arbitrarily large in practice. In Section 8.4, it is shown that this can be addressed by a *hybrid system* that, by alternating between the universal and autonomous mode, maintains the same stability region as a universal swarm while also ensuring small delays for large numbers of swarms.

The stability region is insensitive to peer piece selection policies. However, policies can be quite different w.r.t. other performance metrics. For example, as shown in Section 8, policies can differ drastically in how quickly they stabilize the system when operating within the stability region, as well as in how quickly the missing piece syndrome manifests under the condition of Theorem 2(ii).

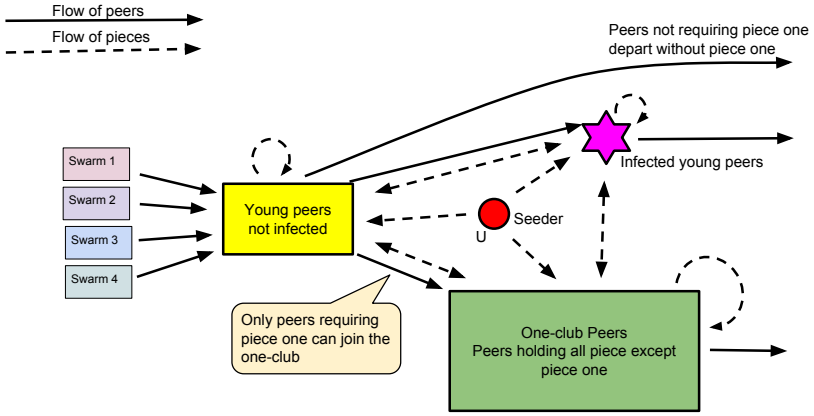


Fig. 1 Flow of peers and pieces in the universal swarm where one-club exists.

The intuition behind Theorem 2 is still related to the missing piece syndrome, as discussed in Section 4. That is, whether or not a large one-club — peers that have all pieces except the missing one — can diminish. As the number of peers in the one-club grows, chances are that they will grow unboundedly and will act as helpers for the peers downloading the other files. Assuming disjoint swarms and an infinite one-club in swarm C, the service capacity to all other swarms is infinite. Therefore, the other swarms are stable, but the one for which $\lambda_C > U$ is unstable. Conversely, if the arrival rate of peers requesting the missing piece is smaller than U, the server will eventually make the one-club vanish. Meanwhile, the other swarms will be stable as the large number of peers in the one-club will act as helpers to the other swarms, and the system will be stable.

Figure 1 shows the flow of peers and the flow of pieces when a large one-club exists in a universal swarm system. In Figure 1, *young peers* refer to peers missing at least one piece other than piece one, *infected peers* refer to peers having piece one, and one-club peers refer to peers having all pieces in $\mathcal{F}$ except piece one. A peer arriving in any swarm becomes a young peer immediately. Young peers receive large service capacity from the one-club, and have three choices. 1) With high probability, young peers requiring piece one quickly download other pieces from one-club peers and thereby join the one-club. 2) With low probability, several young peers download piece one from the seeder or other infected peers, thereby get infected. 3) Young peers not requiring piece one can get all required pieces from the one-club, thereby depart quickly. Only infected peers and the seeder can make the one-club vanish. However, infected peers are very rare because they can quickly depart by downloading all pieces they require from the one-club which provides high service capacity, so the rate for peers to depart from the one-club is close to the upload rate of the seeder. If the arrival rate to the one-club, which is close to the sum of the arrival rates of swarms requiring piece one, is larger than the upload rate of the seeder, there is a positive chance that the one-club will grow unboundedly.

The proof of Theorem 2(i), which is provided in Section 6, is based on the intuition above. It works by rigorously bounding the average arrival rate and departure rate of a large one-club in a long term. Notice that the proof of Theorem 2(i) is quite similar to the proof of instability of a single swarm in [3, Section 3]. They follow a similar argument both based on the intuition of the missing piece syndrome. The major difference is that in the proof of Theorem 2(i), only peers requiring piece one can join the one-club, while peers not requiring piece one can quickly depart, as shown in Figure 1; but in the proof of [3] all peers can join the one-club. Because of that difference, 1) the proof of Theorem 2(i) is more general than the proof in [3]. Actually the former also works in proving the instability result in [3]. 2) Parameters chosen in the proof of Theorem 2(i) differ greatly from that in [3]. 3) A careful argument has to be provided in the proof of Theorem 2(i) to show that the number of young peers is stochastically bounded, because of the departure of both infected peers and peers not requiring piece one. Though the frameworks of the two proofs are the same, it is hard to reuse the proof in [3] in this paper.

6 Proof of Theorem 2(i)

In this section the proof of Theorem 2(i) is provided. If $K = |\mathcal{F}| = 1$, Theorem 2 follows because $\mathbf{n}$ is an $M/M/1$ queue, with arrival rate $\lambda_{\mathcal{F}}$ and service rate U.

In the following, we assume that $K \geq 2$ and that, *w.l.o.g.*, $\lambda_1 := \sum_{C:1\in C} \lambda_C > U$. Notice that $\mathbf{n}$ is irreducible; as such, to show it is transient, it suffices to show that there exists a transient state [2, Proposition 6.3.5]. Disucssion below shows that the initial state where many peers are missing piece one is

a transient state: starting from this state, the number of peers converges to infinity with a positive probability. Transience directly implies that the number of peers converges to infinity with probability one, as (a) there is a finite number of states where the number of peers is bounded by a constant, and (b) the probability for $\mathbf{n}$ to stay in any finite set forever is zero. To construct a transient initial state, assume:

Assumption 2 *Select positive values* $\epsilon, \xi, \rho, \epsilon_o, B, N_o$ *so that*

$$3\epsilon < \lambda_1 - U, \xi < 0.5, \epsilon > 4K\xi U, \tag{3}$$

$$\rho := 2\xi(K-1) < 0.5, \epsilon_o < \xi(\lambda_1 - U - 3\epsilon), \tag{4}$$

$$e^{\lambda_{total}[2(K-1)/\mu+1]}2^{-B} \leq 0.1(1-2^{-\epsilon_o}), \tag{5}$$

$$64K^2\xi U \leq 0.2B(\epsilon - 4K\xi U), \tag{6}$$

$$\lambda_{total} \leq 0.2B\epsilon, U \leq 0.2B\epsilon, \tag{7}$$

$$N_o > 6B, B+1 < \xi(N_o - 3B - 1). \tag{8}$$

Notice that there exists an infinite set of values satisfying Assumption 2. One straightforward way of selecting values satisfying Assumption 2 is: 1) fix ϵ; 2) select ξ small enough based on (3); 3) fix ϵ_o and ρ small enough according to (4); 4) select B large enough to satisfy (5), (6) and (7) at the same time; 5) select N_o large enough to satisfy (8).

Partition the set of peers in two classes: *one-club peers*, which are the peers having all pieces in $\mathcal{F} \setminus \{1\}$, and *young peers*, which are peers missing at least one piece from $\mathcal{F} \setminus \{1\}$. Refer to peers holding piece 1 as *infected peers*; note that infected peers are necessarily young.

Definition 2 Define the following random processes:

- A_t : cumulative number of arrivals of peers wanting to download piece one, up to time t
- N_t : number of peers at time t
- Y_t : number of young peers at time t
- D_t : cumulative number of uploads of piece one by infected peers, up to time t
- Z_t : cumulative number of uploads of piece one by the seeder, up to time t

Construct the following initial state:

Assumption 3 *At* $t = 0$, $N = N_o$ *and all* N *peers are one-club peers.*

Let τ be the extended stopping time defined by $\tau = \inf\{t \geq 0 : Y_t \geq \xi N_t\}$, with the usual convention that $\tau = \infty$ if $Y_t < \xi N_t$ for all t.

Lemma 3 *Under Assumptions 2 and 3,*

$$P\{A_t > -B + (\lambda_1 - \epsilon)t \ \ \textit{for all } t \in [0, \tau)\} \geq 0.9, \tag{9}$$

$$P\{Z_t < B + (U + \epsilon)t \ \ \textit{for all } t \in [0, \tau)\} \geq 0.9, \tag{10}$$

$$P\{Y_t < B + \epsilon_o t \ \ \textit{for all } t \in [0, \tau)\} \geq 0.9, \tag{11}$$

$$P\{D_t < B + \epsilon t \ \ \textit{for all } t \in [0, \tau)\} \geq 0.9. \tag{12}$$

Intuitively but not strictly speaking, in Lemma 3, (9) indicates that the probability of having constantly large number of arrivals as a rate close to λ is large; (10) indicates that the probability for the seeder to keep uploading as a rate larger than U is small; (11) indicates that the probability for the number of young peers to keep bounded is large, because ϵ_o can be arbitrarily small; (12) indicates that the probability of having very small number of uploads of piece one is large, because ϵ can be arbitrarily small.

The proof of Lemma 3 is too long so we attach it at the end of this section. In the following we continue to show that the state described in Assumption 3 is a transient state and thereby Theorem 2(i) works.

Lemma 3 implies Lemma 4:

Lemma 4 $P\{\tau = \infty \quad and \quad \lim_{t\to\infty} N_t = +\infty\} \geq 0.6$

Proof Let $\mathcal{Z}$ be the intersection of the four events on the left sides of (9)-(12). Then $P(\mathcal{Z}) \geq 0.6$. Note that $N_0+A_t-D_t-Z_t$ is no larger than the number of peers wanting to download piece one at t. So, on $\mathcal{Z}$, for all $t \in [0, \tau)$,

$$N_t \geq N_0 + A_t - D_t - Z_t > N_0 - 3B + (\lambda_1 - U - 3\epsilon)t. \tag{13}$$

The second inequality above is the result of applying inequalities in (9), (10) and (12) directly. Notice that τ cannot be finite on $\mathcal{Z}$. Otherwise, at time τ, $Y_\tau < B+1+\epsilon_o\tau$ because of the inequality in (11) and because Y_t can increase with size at most one at time τ; $N_\tau > N_0 - 3B - 1 + (\lambda_1 - U - 3\epsilon)\tau$ because of (13) and because N_t can decrease with size at most one at time τ. Thus, notice that $Y_\tau \geq \xi N_\tau$ from the definition of τ, and combine inequalities about ξ in (4) and (8),

$$\begin{aligned}\xi \leq \frac{Y_\tau}{N_\tau} &< \frac{B+1+\epsilon_o\tau}{N_0 - 3B - 1 + (\lambda_1 - U - 3\epsilon)\tau} \\ &\leq \max\left\{\frac{B+1}{N_0 - 3B - 1},\ \frac{\epsilon_o}{\lambda_1 - U - 3\epsilon}\right\} < \xi,\end{aligned}$$

a contradiction. Thus, on $\mathcal{Z}$, $\tau = \infty$. So $N_t \to \infty$ because of (13), and $\forall t$, $Y_t/N_t < \xi$. Thus, $\mathcal{Z}$ is a subset of the event on the left side of Lemma 4, and so Lemma 4 follows. □

Lemma 4 indicates that the state in Assumption 3 is a transient state, therefore we can conclude here that Theorem 2(i) follows, as discussed in the beginning of this section. To finalize the proof of Theorem 2(i), the proof of Lemma 3 is attached in the following.

6.1 Proof of Lemma 3

Define an alternative process to be the same process as the original process $\mathbf{n}$, but it terminates at time $\tau = \min\{t \geq 0 : Y_t \geq \xi N_t\}$. It is sufficient to prove

the lemma for the alternative process. In the alternative process, A is a Poisson process with rate λ_1, and Z is stochastically dominated by a Poisson process with rate U. Thus, both (9) and (10) follow from (7), and the consequence of Kingman's moment bound [5] described as Proposition 5. Notice that the jump size in a Poisson process is constant one. So $m_1 = m_2 = 1$ when Proposition 5 is applied to generate (9) and (10).

Proposition 5 (Corollary 6.1 in [3]) *Let C be a compound Poisson process with $C_0 = 0$, with jump times given by a Poisson process of rate α, and jump sizes having mean m_1 and mean square value m_2. Then for all $B > 0$ and $\epsilon > \alpha m_1$, $P\{C_t < B + \epsilon t \text{ for all } t\} \geq 1 - \frac{\alpha m_2}{2B(\epsilon - \alpha m_1)}$.*

The inequality (11) follows from (5), Lemma 7 and Lemma 6:

Lemma 6 (Lemma 6.2 in [3]) *Let M denote the number of customers in an $M/GI/\infty$ queue, with arrival rate λ and mean service time m. Suppose that $M_0 = 0, B, \epsilon > 0$, then $P(M_t \leq B + \epsilon t \text{ for all } t) \geq 1 - \frac{e^{\lambda(m+1)} 2^{-B}}{1 - 2^{-\epsilon}}$.*

Lemma 7 *The process Y is stochastically dominated by the number of customers in one $M/GI/\infty$ queue initially empty, arrival rate λ_{total}, and service times follow a Gamma distribution with shape parameter $K - 1$ and scale parameter $2/\mu$.*

Proof Construct one $M/GI/\infty$ queue on the same probability space as the alternative system, so that $Y_t \leq$ the number of peers in the $M/GI/\infty$ queue. Let the $M/GI/\infty$ queue have the same arrival process as the alternative system– *i.e.*, a Poisson process of rate λ_{total}. As $\xi < 0.5$, for any young peer, the intensity of downloads from the one-club is always greater than or equal to $\mu/2$ for the alternative system. Suppose thus that each young peer has an internal Poisson clock, which ticks at rate $\mu/2$, and is such that whenever the internal clock of a young peer ticks, that young peer downloads a piece from the one-club. It is required that a peer remains in the $M/GI/\infty$ system until its internal clock ticks $K - 1$ times. This gives the desired service time distribution, and the service times of different peers in the $M/GI/\infty$ are independent. A young peer may leave the group of young peers (depart or join the one-club) sooner than it leaves the $M/GI/\infty$ system, because a young peer in the alternative system can possibly download pieces at times when its internal clock doesn't tick. But if a peer is still a young peer in the alternative system, it is in the $M/GI/\infty$ system. □

To prove (12), consider the stochastic system described in Table 1, which is called the *comparison system*. It should be clear to the reader that both the alternative system and the comparison system can be constructed on the same underlying probability space; in particular, such a construction can be done so that any infected peer in the alternative system at a given time is also in the comparison system. To enforce this, when a peer becomes infected in the alternative system, require that (a) it also arrives to the comparison system, (b) it discards all pieces it may have downloaded before becoming

Table 1 Specification of comparison system

Alternative system	Comparison system
The seeder creates infected peers at stochastic intensity $\leq \xi U_s$.	The seeder creates infected peers at stochastic intensity ξU_s.
An infected peer creates infected peers at intensity $\leq \xi\mu$.	An infected peer creates infected peers at intensity $\xi\mu$.
An infected peer uploads piece one to one-club at intensity $\leq \mu$.	An infected peer uploads piece one to one-club at intensity μ.
Any infected peer downloads at most $K-1$ additional pieces, at intensity $\geq \mu/2$.	A new infected peer downloads exactly $K-1$ additional pieces, at intensity $\mu/2$.

infected, and (c) it subsequently ignores all opportunities to download except those occurring at the times its internal Poisson clock with ticking rate $\mu/2$ ticks. Because infected young peers may stay longer in the comparison system than in the alternative system, some of the peers in the comparison system correspond to peers that already departed from the alternative system. There can also be some infected peers in the comparison system that never existed in the alternative system because of the higher arrival rate of infected peers in the comparison system.

In all cases, any infected peer in the alternative system is also in the comparison system. That is, any of the following events occurring in the alternative system also occurs in the comparison system: (a) the seeder creates an infected peer, (b) an infected peer creates an infected peer, and (c) an infected peer replicates piece one to a one-club peer. Events of types (b) and (c) correspond to the two possible ways that infected peers can upload piece one. Therefore, this property implies Lemma 8, where $\widehat{D}$ is the cumulative number of uploads of piece one by infected peers, up to time t, in the comparison system:

Lemma 8 *The process* $(D_t : t \geq 0)$ *is stochastically dominated by* $(\widehat{D}_t : t \geq 0)$.

Lemma 9 $(\widehat{D}_t : t \geq 0)$ *can be stochastically dominated by a compound Poisson process* $(\widetilde{D}_t : t \geq 0)$*, with arrival rate of batches* $= \xi U$*, and first and second moments of batch sizes bounded by* $4K$ *and* $64K^2$*, respectively*

Proof Identify two kinds of infected peers in the comparison system–the *root peers*, which are those created by the seeder, and the infected peers created by other infected peers. Assume each root peer signs uniquely on its piece one received from the seeder, and the signature is inherited by all copies of piece one replicated from the root peer. Partition the uploads of piece one according to their signatures. Let $(\widetilde{D}_t : t \geq 0)$ denote a new process which results when all of the uploads of piece one signed by a root peer (in the comparison system) are counted at the arrival time of the root peer. Since $\widetilde{D}$ counts the same events as $\widehat{D}$, but does so earlier, $\widehat{D}_t \leq \widetilde{D}_t$ for all $t \geq 0$. It is sufficient to prove (12) with D replaced by $\widetilde{D}$.

The random process $\widetilde{D}$ is a compound Poisson process. Jumps occur at the arrival times of root peers, which form a Poisson process of rate ξU. Let J be the size of the jump of $\widetilde{D}$ associated with a typical root peer. Then

$J = J_1 + J_2$, where (a) J_1 is the number of infected peers holding piece one signed by the root peer, not counting the root peer itself, and (b) J_2 is the number of uploads of piece one to the one-club by the root peer and these J_1 peers. The family of peers with piece one signed by the root peer is naturally the result of a branching process. So J_1 is the total number of individuals in the branching process, not counting the root. At the same time J_2 is related to the sum of lifetime of all individuals spent in the branching process. To evaluate J_2, it is convenient to consider the busy period of one $M/GI/1$ queueing system, where the arrival of customers in a busy period is naturally a branching process and the service time for one customer in the queue can be viewed as the lifetime for each individual in the branching process. So, consider one $M/GI/1$ queueing system with arrival rate $\xi\mu$ and service times following the distribution of a Gamma random variable $\widetilde{X}$ with shape parameter $K-1$ and scale parameter $2/\mu$. Then the sum of all the times that the root peer and these J_1 peers are in the comparison system is the same as the duration, L, of a busy period of the $M/GI/1$ queue. The random variable J_1 has the same distribution as the number of customers in a busy period of the $M/GI/1$ queue, not counting the customer who started the busy period. Note that ρ in (4) is the load factor for the $M/GI/1$ queue: $\rho = \xi\mu E[\widetilde{X}]$. Based on results about busy periods of $M/GI/1$ queueing system (see [3, Lemma 6.3]) and Assumption 2, bounds on first and second moments of J can be derived follow a similar argument as that in [3, Page 259]: $E[J] = \frac{1+\mu E[\widetilde{X}]}{1-\rho} - 1 \leq 4K$, and $E[J^2] \leq 2\{E[J_1^2] + E[J_2^2]\} \leq 64K^2$. Thereby Lemma 9 follows. □

Hence, (12) with D replaced by $\widetilde{D}$ follows from Proposition 5 and (6). Lemma 3 therefore follows. □

7 Proof of Theorem 2(ii)

In this section, we prove Theorem 2(ii), i.e., the stability of the universal swarm system. We begin by formally describing the Lyapunov function used in proving the stability, then step by step showing that it is a valid Lyapunov function. The stability is established by Foster's criterion [1]. Specifically, the result below is applied, itself a standard version of the criterion:

Lemma 10 *The Markov process* $\mathbf{n}(t)$ *is stable, if there exists* $W(\mathbf{n})$ *which is a non-negative function depending on state* $\mathbf{n}$*, satisfying:*

i) Set $\{\mathbf{n} : W(\mathbf{n}) \leq c\}$ *is a finite set for any* $c \geq 0$.
ii) There exist constants $n_o \geq 0, \xi > 0$*, such that for any state* $\mathbf{n}$ *where the number of peers* $n \geq n_o$*, we have* $QW(\mathbf{n}) \leq -\xi n < 0$*, where* Q *is introduced in Section 3.1 denoting the infinitesimal generator matrix.*

In that case we say that W *is a valid Lyapunov function.*

Intuitively, Lemma 10 tells that the Markov process $\mathbf{n}(t)$ is stable if there is a Lyapunov function which is non-negative and which has negative drift when

the number of peers, n, is large enough. In Lemma 10, the Markov process $\mathbf{n}(t)$ and the generator matrix Q are both introduced in Section 3.1. W is a function on $\mathbf{n}$. The drift vector QW is the multiplication of the generator matrix Q with the vector composed of applying W on each possible value of state $\mathbf{n}$. Notice that QW is a vector with an infinite dimension and can also be viewed as a function on $\mathbf{n}$. Applying drift vector QW is a standard way of describing the drift of a continuous time stochastic process. For readers not familiar with standard definitions and methods applied for continuous time stochastic processes, more details are provided in Chapter 6 in [2].

In the following we first describe the Lyapunov function used to prove Theorem 2(ii), then show that it satisfies the above Foster criterion.

Definition 3 $\Delta := U - \max_{i:i\in\mathcal{F}} \sum_{C:i\in C} \lambda_C > 0$.

Definition 4 $\mathcal{E}_C := \{\langle C', S'\rangle : C' \not\subseteq C, S' \subseteq C\}$ is set of types of peers which may hold the set of pieces C in the future, $E_C := \sum_{\langle C',S'\rangle\in\mathcal{E}_C} n_{\langle C',S'\rangle}$ is the number of peers with types in $\mathcal{E}_C$.

Definition 5 $\mathcal{H}_C := \bigcup_{i\in\mathcal{F}\setminus C} \{\langle C', C\cup\{i\}\rangle : C' \not\subseteq C\cup\{i\}\}$ is set of types of peers which hold one more piece than C, $H_C := \sum_{\langle C',S'\rangle\in\mathcal{H}_C} n_{\langle C',S'\rangle}$ is the number of peers with types in $\mathcal{H}_C$.

Definition 6 $r \in (0, \frac{1}{2})$, $d \in (1, \infty)$, $\beta \in (0, \frac{1}{2})$, $\alpha := 8K(\lambda_{total} + \Delta/2)/U$ are four constants, and

$$\phi(x) := \begin{cases} (2d + \frac{1}{2\beta} - x) & 0 \le x \le 2d \\ \frac{\beta}{2}(x - 2d - \frac{1}{\beta})^2 & 2d < x \le 2d + \frac{1}{\beta} \\ 0 & x > 2d + \frac{1}{\beta} \end{cases}$$

Definition 7 The Lyapunov function W is defined as $W := \sum_C r^{|C|} T_C$, where

$$T_C := \begin{cases} \frac{1}{2}E_C^2 + \alpha E_C \phi(H_C), |C| \le K-2, \\ \frac{1}{2}E_C^2, |C| = K-1 \end{cases}$$

Function ϕ is affine in the range $[0, 2d]$, the component $\frac{1}{2}E_C^2$ is quadratic, and the component $\alpha E_C \phi(H_C)$ vanishes in most cases. Thus the Lyapunov function W is quadratic in the state $\mathbf{n}$ in a corresponding range. However, the proof needs to go beyond quadratic functions in order to apply the Foster criterion.

In the following, we show that for suitable choices of constants r, d, β the function W in Definition 7 satisfies the Foster's criterion Lemma 10. Further define:

Definition 8 $M_\phi := 3d + \frac{1}{\beta}$. Note $M_\phi \ge \max_x \phi(x)$ and $M_\phi > \min\{x : \phi(x) = 0\} + d > 1$.

Definition 9 Define $D_{\langle C,S\rangle}$ to be the transition rate for type $\langle C, S\rangle$ peers to download pieces.
Define $D_S := \sum_{C:C\in\mathcal{C},C\not\subseteq S} D_{\langle C,S\rangle}$.
Define $D_{total} := \sum_{S:S\subsetneq\mathcal{F}} D_S$.
Note that $D_{\langle C,S\rangle}, D_S, D_{total}$ are functions of the state $\mathbf{n}$.

The following two lemmas obviously hold:

Lemma 11 *Function ϕ verifies $\phi'(x) = -1$ for $0 \le x \le 2d$ and $\phi'(x) = 0$ for $x \ge 2d + 1/\beta$. Moreover, ϕ' increases linearly from -1 to 0 in $[2d, 2d + 1/\beta]$.* $\forall x \ge 0, \phi'(x) \in [-1, 0]$.

Lemma 12 $D_S \le U+\mu \min\{n_S, n-n_S\}$, $D_{total} \le U+n\mu$, $D_S \ge (U + H_S\mu)\, n_S/n$.

In the proof, the following two classes of states are considered separately. The variable σ is to be selected within $\sigma \in (0, 1/2)$. The two classes overlap and their union includes every non-zero state:

Definition 10 Class I is the set of states $\mathbf{n}$ such that $n_S/n > 1 - \sigma$ for some $S \subsetneq \mathcal{F}$; Class II is the set of states $\mathbf{n}$ such that there exist $C_1, C_2 \subsetneq \mathcal{F}$, $C_1 \ne C_2$, so that, $n_{C_1}/n > \sigma/2^K$ and $n_{C_2}/n > \sigma/2^K$.

For a specific σ, a state $\mathbf{n}$ can either be Class I or Class II, or both. The main idea of the proof is to show that W is a valid Lyapunov function for an appropriate choice of (r, d, β, σ). The given parameters of the network, $\lambda_C(C \in \mathcal{C}), U$ and μ, are treated as constants. Functions on the state space may or may not depend on the variables r, d, β and σ. It is convenient to adopt the big theta notation $\Theta(\cdot)$, with the understanding that it is uniform in these variables; this is summarized in the following definitions.

Definition 11 Given functions f and g on the state space $\mathcal{D}$, say $f = \Theta(g)$ if there exist $k_1, k_2, n_0 > 0$, not dependent on (r, d, β, σ), such that $k_1|g(\mathbf{n})| \le |f(\mathbf{n})| \le k_2|g(\mathbf{n})|$ for all $\mathbf{n}$ such that $n > n_0$.

For example, $2 = \Theta(1), \lambda_{total} n = \Theta(n)$, $d = \Theta(d)$, $1 \le \Theta(n)$ and $\Theta(n) - \Theta(n)/2 = \Theta(n)$. Notice that d and $\Theta(1)$ cannot be compared. Similarly, adopt notions of "small enough" and "large enough" that are uniform in (r, d, β, σ):

Definition 12 Say that "condition A is true if $x > 0$ is *small enough*" if there exists a constant $k > 0$, not depending on (r, d, β, σ), such that A is true for any $x \in (0, k)$. Similarly, say that "condition A is true if $x > 0$ is *large enough*" if there exists a constant $k > 0$, not depending on (r, d, β, σ), such that A is true for any $x \in (k, \infty)$.

Identify an approximation to the drift of W. Notice that the infinitesimal generator Q is linear, so $Q(W) = \sum_C r^{|C|} Q(T_C)$, with $Q(T_C) = \frac{1}{2}Q(E_C^2) + \alpha Q(E_C\phi(H_C))$.

Definition 13 Define $\mathcal{Q}W$, an approximation of $Q(W)$, as $\mathcal{Q}W := \sum_C r^{|C|}\mathcal{Q}T_C$, with
$\mathcal{Q}T_C := E_C Q(E_C) + \alpha E_C Q(\phi(H_C))$ if $|C| \le K - 2$,
$\mathcal{Q}T_C := E_C Q(E_C)$ if $|C| = K - 1$.

Our proof relies on Lemmas 13, 14, 15, and 16, which are conditioned on (r, d, β, σ), to bound terms in $Q(W)$ and $\mathcal{Q}W$. These lemmas with their proofs are provided below. Lemma 17 directly applies results in Lemmas 13, 14, 15, and 16 to prove that W is a valid Lyapunov function.

Definition 14 is provided to simplify the proofs of Lemmas 13, 14, 15, 16 and 17.

Definition 14 Define Γ_{J_1,J_2} for $J_1, J_2 \in \mathcal{T}, J_1 \neq J_2$ to be the aggregated transition rate for type J_1 peers to become type J_2 peers.

Define $\Gamma_{\mathcal{X}_1,\mathcal{X}_2} := \sum_{J_1:J_1\in\mathcal{X}_1} \sum_{J_2:J_2\in\mathcal{X}_1} \Gamma_{J_1,J_2}$, where $\mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset$.

Define $D_{\mathcal{H}_S} := \sum_{J:J\in\mathcal{H}_S} D_J$, for $\mathcal{H}_S$ in Definition 5.

Define $A_{\mathcal{H}_S}$ to be the total transition rate for peers to join the group of peers with types in $\mathcal{H}_S$.

Let $\mathcal{P}_S := \{\langle C, S\rangle : C \not\subseteq S, C \in \mathcal{C}\}$ be the set of types of peers holding piece set S.

Notice that Γ, D, A are all functions of $\mathbf{n}$.

Lemma 13 *Bound for the approximation error:*

$$|Q(W) - \mathcal{Q}W| \leq M_\phi(D_{total} + 1)\Theta(1). \tag{14}$$

Proof Compare $Q(W)$ and $\mathcal{Q}W$ term by term. Consider terms $Q(T_C)$ and $\mathcal{Q}T_C$. First, assume $|C| \leq K - 2$. Because α is fixed, $|Q(T_C) - \mathcal{Q}T_C| \leq a_1 + \alpha(a_2 + a_3)$, with

$$\begin{aligned}
a_1 &= \left|\frac{1}{2}Q(E_C^2) - E_C Q(E_C)\right| \leq \lambda_{total} + D_{total},\\
a_2 &= \left|Q(E_C\phi(H_C)) - Q(E_C)\phi(H_C) - E_C Q(\phi(H_C))\right|,\\
a_3 &= |Q(E_C)\phi(H_C)| \leq M_\phi(\lambda_{total} + D_{total}).
\end{aligned}$$

The only way E_C and $\phi(H_C)$ can simultaneously change is that some peer with type in $\mathcal{E}_C$ becomes a peer with type in $\mathcal{H}_C$, causing E_C to decrease by one, and $\phi(H_C)$ to decrease by at most one, so $a_2 \leq \Gamma_{\mathcal{E}_C,\mathcal{H}_C}$. Notice the fact that $\Gamma_{\mathcal{E}_C,\mathcal{H}_C} \leq D_{total}$, it follows that $\forall |C| \leq K - 2$, $|Q(T_C) - \mathcal{Q}T_C| \leq M_\phi(D_{total} + 1)\Theta(1)$. Secondly, assume $|C| = K - 1$. Then, $|Q(T_C) - \mathcal{Q}T_C| = a_1 \leq \lambda_{total} + D_{total} \leq M_\phi(D_{total} + 1)\Theta(1)$. There are only finitely many terms of T_C in W (2^K in total), and notice that $r < 1$, Lemma 13 follows. □

Lemma 14 *If d is large enough, $\forall C \subsetneq \mathcal{F}$, $Q(E_C) \leq \Theta(1)$, $Q(\phi(H_C)) \leq M_\phi\Theta(1)$, and $\mathcal{Q}T_C \leq M_\phi\Theta(E_C) \leq M_\phi\Theta(n)$.*

Proof The upper bound for the drift of E_C is obvious: $Q(E_C) \leq \lambda_{total} = \Theta(1)$. Next consider $Q(\phi(H_C))$. Because ϕ is a decreasing function, only the rate for H_C to decrease contributes to the positive part in the drift of $\phi(H_C)$, so to consider an upper bound of $Q(\phi(H_C))$ it suffices to consider the rates of transitions which reduce H_C. There is only one way for H_C to decrease: peers with types in $\mathcal{H}_C$ to download a novel piece – with aggregate rate $D_{\mathcal{H}_C}$. Each peer with type in $\mathcal{H}_C$ downloading a novel piece can cause $\phi(H_C)$ to increase

at most one. Thus, an upper bound for the drift of $\phi(H_C)$ is $Q(\phi(H_C)) \leq D_{\mathcal{H}_C} \leq U + H_C\mu = \Theta(1) + H_C\Theta(1)$, by Lemma 12.

Choose d large enough, i.e. $d > 1$, such that $M_\phi > 2d + 1/\beta + 1$. Thus $Q(\phi(H_C))$ vanishes when $H_C > M_\phi$, because $\phi(H_C)$ vanishes when $H_C > 2d + 1/\beta$ and the decreasing of H_C when state changes is bounded by $1 < d$. Hence $Q(\phi(H_C)) \leq M_\phi\Theta(1)$, because $M_\phi > 1$.

Finally, the bound on $\mathcal{Q}T_C$ follows from the other two bounds already proved. □

Lemma 15 *If d is large enough, $\sigma M_\phi, \beta$ are small enough, for any $S \subsetneq \mathcal{F}$ and any nonzero state $\mathbf{n}$ such that $n_S/n > 1 - \sigma$, $\mathcal{Q}T_S \leq -\frac{1}{2}\Delta E_S$.*

Proof Assume that $n_S/n > 1-\sigma$, where $0 < \sigma < \frac{1}{4}$ is to be specified. Consider two cases. Suppose first $|S| = K - 1$. Peers with cache S only miss one piece $i \in \mathcal{F} \setminus S$, and $\mathcal{Q}T_S = E_S Q(E_S) \leq E_S\big[\sum_{C:i\in C} \lambda_C - U(1-\sigma)\big] \leq -\frac{1}{2}\Delta E_S$, if σ is set to be small enough: $\sigma < \Delta/(2U)$. Lemma 15 follows.

Suppose secondly that $|S| \leq K - 2$, by [17, Lemma 19],

$$\begin{aligned} Q(\phi(H_S)) &\leq \phi'(H_S)Q(H_S) + \beta/2(A_{\mathcal{H}_S} + D_{\mathcal{H}_S}) \\ &= \phi'(H_S)(A_{\mathcal{H}_S} - D_{\mathcal{H}_S}) + \beta/2(A_{\mathcal{H}_S} + D_{\mathcal{H}_S}). \end{aligned}$$

Substitute the above inequality into $Q(E_S) + \alpha Q(\phi(H_S))$, which is one component in Definition 13, apply Lemma 12,

$$\begin{aligned} &Q(E_S) + \alpha Q(\phi(H_S)) \leq \bar{\varsigma} - \varpi, \text{ where} \\ &\begin{cases} \bar{\varsigma} := \lambda_{total} - \frac{1}{2}D_S + \alpha\phi'(H_S)A_{\mathcal{H}_S} \\ \varpi := \frac{1}{2}D_S + \alpha\phi'(H_S)D_{\mathcal{H}_S} - \alpha\beta/2(A_{\mathcal{H}_S} + D_{\mathcal{H}_S}) \end{cases}. \end{aligned} \tag{15}$$

Two bounds on $\bar{\varsigma}$ and ϖ respectively can be established. The first bound is $\bar{\varsigma} \leq -\Delta/2$. To prove the bound, suppose first that $H_S < d$: Then, $\phi'(H_S) = -1$ and because the seeder applies RN, $A_{\mathcal{H}_S} \geq \Gamma_{\mathcal{P}_S,\mathcal{H}_S} \geq U(1-\sigma)/K \geq U/(4K)$. So $\bar{\varsigma} \leq \lambda_{total} - \alpha U/(4K) \leq -\Delta/2$. Suppose secondly that $H_S \geq d$: Then, $D_S \geq d\mu(1-\sigma) \geq \frac{1}{2}d\mu$. So $\bar{\varsigma} \leq \lambda_{total} - \frac{1}{2}d\mu \leq -\frac{1}{2}\Delta$, for d large enough such that $d > 2(\lambda_{total} + \Delta/2)/\mu$.

The second bound is $\varpi \geq 0$. To prove the bound, let ω_S be the number of peers holding pieces not in S. For $\sigma < \frac{1}{2}$,

$$D_S \geq (U + \omega_S\mu)(1-\sigma) \geq \frac{1}{2}(U + \omega_S\mu). \tag{16}$$

Notice that pieces novel to peers with types in $\mathcal{H}_S$ are not contained in S. The number of peers that can upload pieces to peers with types in $\mathcal{H}_S$ is no larger than ω_S, so by Lemma 12

$$D_{\mathcal{H}_S} \leq (U + \omega_S\mu)\sigma \leq 2\sigma D_S. \tag{17}$$

In addition, $A_{\mathcal{H}_S} = \Gamma_{\mathcal{P}_S,\mathcal{H}_S} + \Gamma_{I_S,\mathcal{H}_S}$, where $I_S := \{\langle C', S'\rangle : |S'| = |S|, S' \neq S\}$. The number of peers with types in I_S is no larger than ω_S. Therefore

$\Gamma_{I_S,\mathcal{H}_S} \leq U + \omega_S\mu \leq 2D_S$ by (16). Followed by $\Gamma_{\mathcal{P}_S,\mathcal{H}_S} \leq D_S$, $A_{\mathcal{H}_S} \leq 3D_S$. Combine $A_{\mathcal{H}_S} \leq 3D_S$ with Lemma 12, and (17),

$$\varpi \geq \frac{1}{2}D_S - 2\alpha D_{\mathcal{H}_S} - \frac{\alpha\beta}{2}A_{\mathcal{H}_S} \geq (\frac{1}{2} - 4\alpha\sigma - \frac{3}{2}\alpha\beta)D_S.$$

The inequality $\varpi \geq 0$ follows by setting $4\alpha\sigma < \frac{1}{4}$ and $\frac{3}{2}\alpha\beta < \frac{1}{4}$.

Therefore, $\bar{\varsigma} \leq -\frac{1}{2}\Delta$ and $\varpi \geq 0$ imply that, when $n_S/n > 1-\sigma$ and $|S| \leq K-2$, $\mathcal{Q}T_S = [Q(E_S) + \alpha Q(\phi(H_S))]E_S \leq -\frac{1}{2}\Delta E_S$ and Lemma 15 follows. □

Lemmas 14 and 15 imply Lemma 16:

Lemma 16 *If d is large enough, $\beta, rM_\phi, \sigma M_\phi r^{-K}$ are small enough, (a) on Class I, $\mathcal{Q}W \leq -r^K\Theta(n)$; (b) on Class II, $\mathcal{Q}W \leq -r^K\sigma^3\Theta(n^2) + M_\phi\Theta(n)$.*

Proof First, consider Lemma 16(a). There are only finitely many types, fix a set $S \subsetneq \mathcal{F}$ and consider the set of Class I states $\mathbf{n}$ for which $n_S/n > 1-\sigma$. Because $\sigma \in (0, \frac{1}{2})$, $E_S > \frac{1}{2}n$. By Definition 3, $\triangle > 0$. By Lemma 15,

$$\mathcal{Q}T_S \leq -\frac{1}{4}\triangle n = -\Theta(n). \tag{18}$$

Consider two conditions: (a) for type C with $|C| > |S|$, E_C may be larger than E_S. Lemma 14 and (18) imply $r^{|C|}\mathcal{Q}T_C \leq rM_\phi r^{|S|}\Theta(n) < 2^{-K-1}r^{|S|}\big|\mathcal{Q}T_S\big|$, if rM_ϕ is selected to be small enough; (b) for type C with $|C| \leq |S|$ but $C \neq S$, $E_C \leq \sigma n$; note that $r^{|C|}r^K \leq r^{|S|}$, Lemma 14 and (18) imply $r^{|C|}\mathcal{Q}T_C \leq r^{|C|}M_\phi\Theta(E_C) \leq \sigma M_\phi r^{-K}r^{|S|}\Theta(n) < 2^{-K-1}r^{|S|}\big|\mathcal{Q}T_S\big|$, if $\sigma M_\phi r^{-K}$ is selected to be small enough.

Notice that $E_S \geq n/2$, Lemma 15 implies that

$$\mathcal{Q}W = r^{|S|}\mathcal{Q}T_S + \sum_{C:C\neq S} r^{|C|}\mathcal{Q}T_C \leq r^{|S|}\mathcal{Q}T_S + \frac{1}{2}r^{|S|}\big|\mathcal{Q}T_S\big| \leq -\frac{1}{8}r^{|S|}\triangle n \leq -r^K\Theta(n),$$

which proves Lemma 16(a).

Second, consider Lemma 16(b). Suppose $C_1 \not\subseteq C_2$ and consider the set of Class II states $\mathbf{n}$ such that $n_{C_1}/n > \eta, n_{C_2}/n > \eta$, where $\eta = \sigma/2^K$. In such states: $D_{C_2} \geq n_{C_2}n_{C_1}\mu/n \geq \mu\eta^2 n \in \sigma^2\Theta(n)$. Notice that $E_{C_2} \geq n_{C_2} \geq \eta n$,

$$E_{C_2}Q(E_{C_2}) = E_{C_2}(\lambda_{\mathcal{E}_{C_2}} - D_{C_2}) \leq -\sigma^3\Theta(n^2) + \Theta(n).$$

Lemma 14 indicates $E_{C_2}Q(\phi(H_{C_2})) \leq M_\phi\Theta(n)$, so

$$\mathcal{Q}T_{C_2} = E_{C_2}Q(E_{C_2}) + \alpha E_{C_2}Q(\phi(H_{C_2})) \leq -\sigma^3\Theta(n^2) + M_\phi\Theta(n).$$

Obviously the above inequality works also for the case $|C_2| = K-1$ where the ϕ term does not exist. Followed by the above inequalities and the result $\forall C$, $\mathcal{Q}T_C \leq M_\phi\Theta(n)$ indicated by Lemma 14, over the set of all Class II states,

$$\mathcal{Q}W \leq r^{|C_2|}\mathcal{Q}T_{C_2} + \sum_{C:C\neq C_2}\mathcal{Q}T_C \leq -r^K\sigma^3\Theta(n^2) + M_\phi\Theta(n),$$

which proves Lemma 16(b). □

Notice that the conditions of Lemmas 13, 14, 15, and 16 are consistent with each other, Lemma 17 follows.

Lemma 17 *There exists (r, d, β, σ) satisfying all conditions of Lemmas 13, 14, 15, and 16, such that W is a valid Lyapunov function.*

Proof On Class I, $D_{total} = D_S + \sum_{C:C\neq S} D_C \leq 2U + 2(n - n_S)\mu \leq 2U + 2\sigma n\mu = \Theta(1) + 2\sigma\Theta(n)$ by Definition (9). So Lemma 13 implies that on Class I, $|Q(W) - \mathcal{Q}W| \leq \sigma M_\phi \Theta(n) + M_\phi \Theta(1)$. Combined with Lemma 16(a), on Class I, $Q(W) \leq -r^K\Theta(n) + M_\phi\Theta(1)$ if $\sigma M_\phi r^{-K}$ is small enough. On Class II, $D_{total} \leq U + n\mu = \Theta(n)$, so Lemma 13 implies that $|Q(W) - \mathcal{Q}W| \leq M_\phi\Theta(n)$. On Class II, Combined with Lemma 16(b), $Q(W) \leq -r^K\sigma^3\Theta(n^2) + M_\phi\Theta(n)$. Thus, if (r, d, β, σ) satisfies conditions of Lemmas 13 ,14, 15, 16, and $\sigma M_\phi r^{-K}$ is small enough, $\exists \varsigma > 0$ such that $Q(W) \leq -\varsigma n$ whenever n is larger than some constant. Thus W is a valid Lyapunov function. □

Lemma 17 is the final stage, where intermediate lemmas and definitions are utilized to conclude that W is a valid Lyapunov function. Combine Lemma 10 and 17, we conclude that Theorem 2(ii) works.

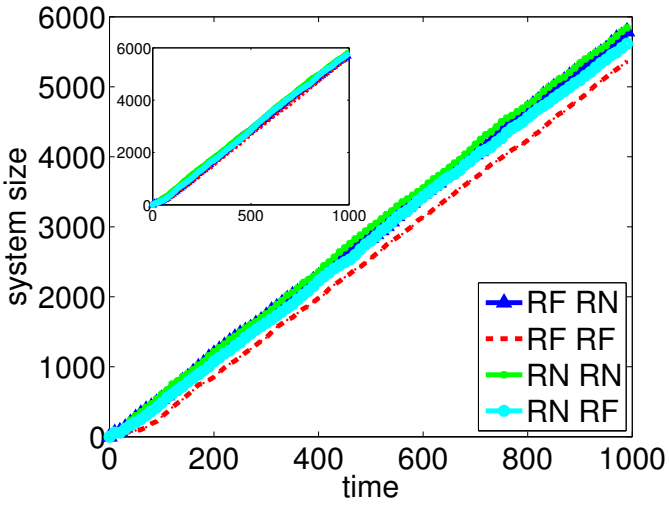
8 Evaluation

In this section, we numerically evaluate the performance of universal swarms through simulations. We study the behavior of the universal swarm system under different piece selection policies, and describe the average peer sojourn time under varies system settings. In what follows, denote by RF, RN, PRN and PRF the piece selection policies rarest first, random novel, priority random novel and priority rarest first. Note that PRF and PRN reduce to RF and RN when the system operates in autonomous mode.

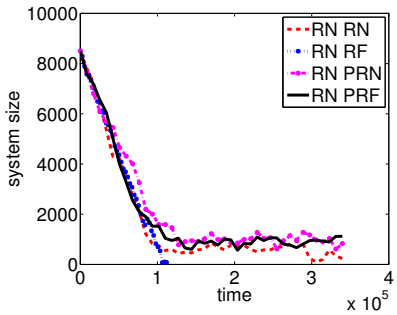
8.1 Autonomous vs. Universal Swarms

We begin by numerically illustrating the results of Theorems 1 and 2, which is achieved below by studying the evolution of the system size n in autonomous and universal mode for a system comprising 3 swarms, each requesting a different 3-piece set. The seed rate is $U = 3.1$ and the arrival rate in each swarm is $\lambda = 3.0$; note that Theorems 1 and 2 imply that the autonomous mode is unstable while the universal system is stable, in this regime.

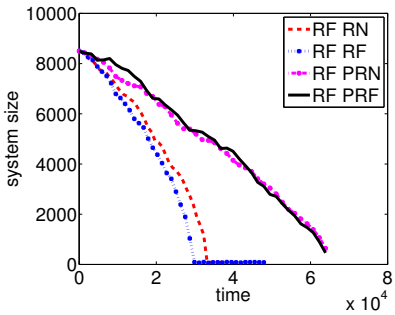
Figure 2(a) shows the evolution of the system size in autonomous mode, when the seeder statically allocates 1/3 of its upload rate to each swarm, for different combinations of policies at the seeder and the peers. All simulations start from an empty system. Even though applying RF at both the seeder and the peers leads to a slightly smaller system size, the missing piece syndrome manifests in all four cases. We repeat these experiments with the seeder allocating its rate dynamically, so that each swarm receives pieces at a rate



(a) Autonomous mode, static and dynamic (inset) allocation.



(b) Universal mode, RN at seeder.



(c) Universal mode, RF at seeder.

Fig. 2 System size VS time. ("RN RF" means RN at seeder and RF at peers, other legends follow similarly.)

proportional to its size. The results (inset of Figure 2(a)) show that instability persists in this setup too.

We repeat these experiments in universal mode, starting the system from an initial state comprising 8500 peers forming a *one-club*: all peers belong to the same swarm, and store in their cache all nine pieces *except for one common piece they request*. Figure 2(b) shows the system evolution when the seeder applies RN; indeed the system stabilizes after 10^5 time units, confirming Theorem 2. The system stabilizes faster (in the order of 10^4 time units) when the seeder applies RF, as seen in Figure 2(c), with RF at both seeder and peers stabilizing the system the fastest (in roughly $3 \cdot 10^4$ time units). Interestingly, prioritizing pieces at peers (through either PRN or PRF) leads

to *slower* stabilization: this is precisely because these policies reduce piece diversity.

8.2 Instability and Meta-Stability in Universal Mode

Next consider the same experiments as above with $U = 2.9$. As the arrival rate at each swarm is $\lambda = 3$, Theorem 2 stipulates that when the seeder applies RN the system is unstable. A question to address here is how quickly the missing piece syndrome manifests in this case, depending on the piece selection policies.

To evaluate this, the following experiments are constructed. Start our simulations with initial system size n_0, where the initial state comprises all peers forming a one-club (*i.e.*, storing all pieces but one). Then terminate the simulation when either the system size increases to the threshold $\max(2000+n_0, 2n_0)$ or the simulation time reaches 10^7, whichever occurs first. First conduct this experiment with an empty initial state $n_0 = 0$; if the experiment does not reach the threshold in 10^7 units, increase n_0 by 100 and repeat the experiment. This way, the *critical one-club size* is identified: if the system reaches a state with a one-club of that size, it becomes unstable.

Our simulation results for the case where the seeder applies RN are summarized in the top half of Table 2. It can be seen that the missing piece syndrome indeed manifests at the critical initial conditions, with the one-club comprising more than 90% of the peer population at termination time. When peers use any policy other than RF, the critical one-club size is 0. In contrast, when peers use RF, the syndrome manifests only when $n_0 = 500$; indeed, using RF improves the diversity of pieces in the system, which in turns makes reaching a critical one-club size more difficult. This behavior becomes even more striking when the seeder uses RF: as shown in the bottom half of Table 2, piece diversity is so high that critical one-club sizes lie between 2 and 8 thousand peers.

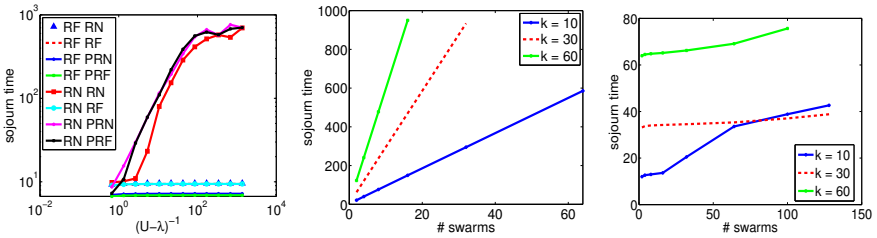
Crucially, in all simulations starting from an initial size below the critical value, an interesting observation is that the system size actually *decreases* to a size below 200 and lingers around this value for the entire 10^7 time units! This implies that, though the system is clearly not stable in any of the cases in Table 2, applying RF at the seeder or peers yields *meta-stability*: although there exists a critical one-club size, its value is so high that it is quite hard to reach from the "typical" size at which the system operates most of the time ($\sim$200 peers in our simulations). A natural question to ask is what is the critical value n_0, as well as what is the "typical" size at which a meta-stable system operates most of the time; these question are revisited below in Section 8.4.

8.3 Average Sojourn Time

Next we turn the attention to the sojourn time. First study a universal system comprising 3 swarms with 3 pieces each. The seed rate is fixed at $U = 3.0$ and

Table 2 Critical One-Club Size

Policy		Critical	Final	Final One	Sim.
Seed	Peer	n_0	Size	Club Ratio	Duration
RN	PRN	0	2000	95.6%	13181
	PRF	0	2000	95.4%	17211
	RN	0	2000	93.3%	13603
	RF	500	2500	94.7%	22655
RF	PRN	2100	4200	98.1%	74655
	PRF	2000	4000	98.0%	51415
	RN	8000	16000	99.4%	283197
	RF	8100	16200	99.4%	323738



(a) Approach boundary from below. (b) Universal, $U = 3.0$, $\lambda = 1.5$. (c) Hybrid, $U = 3.0$, $\lambda = 1.5$.

Fig. 3 Average sojourn time for universal swarms.

the swarm arrival rate varies as $\lambda = U(1 - \frac{1}{2^i})$, for $i = 1, \ldots, 10$, remaining within the stability region but approaching U from below. Figure 3(a) plots the average sojourn time for different piece selection policies as a function of $1/(U - \lambda)$. It is observed that, as λ approaches U, the sojourn time under the RN policy at the seeder increases considerably, with the exception of the RN-RF case, *i.e.*, when peers use RF. In all four cases for which the seeder uses RF, the sojourn time remains practically constant as λ approaches U. This is consistent with the fact that, by meta-stability, when the seeder uses RF the system size remains small most of the time even if $\lambda > U$; as such, there is no sharp increase in the sojourn time as λ approaches U from below.

Next, we study how the sojourn time scales with the number of swarms L. Figure 3(b) plots the average sojourn time vs L for the case where each swarm comprises peers requesting a k-piece file, for $k \in \{10, 30, 60\}$. The total number of pieces is $K = kL$. Across all values of k, the average sojourn time increases linearly as L increases. Similarly, the sojourn time also increases proportionally to k. Thus, the increased stability offered by bundling swarms together comes at the cost of increased delays; we address this in the next section by showing that delays can be suppressed for a wide range values of L by using a *hybrid* approach, alternating between universal and autonomous mode.

8.4 Stable, Low Sojourn Universal Swarms

Our simulations suggest that, in a meta-stable swarm, there are two important system sizes: the *operating* size $n_{\rm op}$, which is the size around which the system stays most of the time, and the *critical size* n_0, which is the size of a one-club that, once attained, leads the system to instability. If the two sizes are sufficiently far apart from each other, the system will exhibit meta-stability: when $N \approx n_{\rm op}$, it will take a long time for N to reach n_0, from which the missing piece syndrome manifests.

Calculating exactly $n_{\rm op}$ and n_0 is quite challenging, and is beyond the scope of this paper. Nevertheless, we derive below some simple estimates of $n_{\rm op}$ and n_0 when (a) the system comprises of a single swarm, (b) $\lambda > U$, and (c) both the seeder and peers use RF. In particular, the following lemma holds:

Lemma 18 *In a single swarm system, the operating and critical sizes can be approximated by the following quantities:*

$$n_{\rm op} \approx \frac{\lambda K}{\mu}, \text{ and } n_0 \approx \frac{\lambda(K-1)}{\mu}\left[\frac{1}{2}\frac{U(K-1)}{\lambda - U} - 1\right]. \tag{19}$$

Proof Consider a single swarm, where peers arrive with rate λ and wish to download K pieces. Assume that the seeder has upload rate $U < \lambda$, so that the system is unstable, and peers have upload rate μ. When the system is in the operating state, it is expected that the diversity of pieces is high enough, so that every contact a peer makes leads to a piece download. Under this assumption, as a peer wishes to download K pieces, the expected sojourn time is K/μ. By Little's law, as the arrival rate is λ, an approximation of the operating size is therefore $n_{\rm op} \approx \frac{\lambda K}{\mu}$.

Estimating the critical size of a one-club requires a more involved argument. Suppose that a one-club with size B has formed. For B large, peers outside the one-club download pieces from the one-club at a rate close to μ. As such, the expected time it takes a new peer to be converted to a one-club peer is approximately $(K-1)/\mu$; hence, the number of *young peers*, *i.e.*, peers outside the one-club, is approximately $\lambda(K-1)/\mu$. Young peers can become *infected*, *i.e.*, obtain the piece the one-club peers are missing. As the seeder uses the RF policy, the rate with which the seeder infects young peers is $U\frac{\lambda(K-1)/\mu}{B+\lambda(K-1)/\mu}$. Ignoring the fact that young peers may also infect other young peers, and assuming that an infection occurs at an instant sampled uniformly at random within a young peer's lifetime, each infected peer stays for $(K-1)/2\mu$ time units before it departs, in expectation. Then, the drift of the one-club is roughly $\Delta B = \lambda - U - U\frac{\lambda(K-1)/\mu}{B+\lambda(K-1)/\mu}\frac{(K-1)}{2\mu}\mu$. Requiring the drift to be zero and solving for B, Equation (19) follows. □

Using these two estimates, we propose a *hybrid system* that attains the increased stability region of the universal swarm, while also ensuring that the sojourn times remain small for a wide range values of L. The hybrid system alternates between the autonomous mode, whereby swarms operate in isolation while sharing a U/L portion of the seeder's capacity, and the universal

mode, where swarms are bundled together. In particular, consider a system with L swarms, each requesting a file of $k = K/L$ pieces. The system alternates between the two modes according to the following rules: (a) If in autonomous mode, the system switches to universal mode if any single swarm has size $\geq n_{\text{op}} + \max(n_0, 2n_{\text{op}})$; (b) if in universal mode, the system switches to autonomous mode if each piece requested by a swarm is held by at least $\max(n_{\text{op}}/10, 1)$ peers within the swarm. Values n_{op}, n_0 are computed by (19), assuming an upload rate U/L and a number of pieces k. Intuitively, the universal mode is applied when there is strong evidence that the missing piece syndrome is manifesting, as the swarm size becomes greater than $n_{\text{op}}+n_0$. The system reverts to an autonomous mode when there is enough diversity in each swarm—each piece is held by at least the one tenth of the peer population.

The hybrid system switches to the universal mode when the system size becomes large, so it exibits the increased stability region of universal swarms described in Theorem 2. Figure 3(c) shows the sojourn time of a hybrid system as L increases. In contrast to Figure 3(b), for $k = 30$ and $k = 60$, the sojourn time stays close to the value attained when $L = 1$ ($\sim$33 and $\sim$64 time units, respectively). For $k = 10$, the sojourn time starts increasing linearly after $L = 12$.

These improved sojourn times appear precisely because of the meta-stability. Indeed, swarms operate fine most of the time without the intervention of other swarms, and this is why they experience the same delay as if $L = 1$. As $U/L < \lambda$, the autonomous mode is unstable; however, at the few (and rare) occasions when the missing piece syndrome manifests, bundling swarms together ensures the system quickly stabilizes and reverts to its operating size.

The knee observed for $k = 10$ suggests that this behavior cannot be sustained for arbitrarily large L. Equation (19) can help us give an approximate answer to how large L can be. Indeed, for the system to be meta-stable, the critical one-club size must be significantly larger than the operating size. Requiring that $n_0 > 2n_{\text{op}}$, so that the missing piece syndrome rarely manifests, and taking $K/(K-1) \approx 1$, gives the following heuristic for metastability when L=1: $K\frac{U}{\lambda - U} > 6$. Consider now $L > 1$ swarms in autonomous mode, each requesting $k = K/L$ pieces. Each swarm gets a U/L upload rate in autonomous mode. Then, the above condition becomes $L < \frac{U}{6(\lambda - \frac{U}{L})}k \approx \frac{U}{6\lambda}k$. In other words, the hybrid system can support a number of swarms L with small delay so long as L is of the order of k, the number of pieces in each swarm. As the number of pieces in a file typically numbers in the thousands, this implies that the above system can sustain low sojourn times for a large number of swarms.

9 Conclusion

We have formally characterized the stability region of universal swarms. Our simulations reveal an interesting relationship between stability and piece selection policies. Though piece selection policies may share the same stability

region they can differ in their ability to resist the missing piece syndrome. This intuition helps us design a hybrid system that achieves simultaneously a large stability region and low sojourn times; establishing its properties analytically, and investigating other hybrid designs, remains an open question.

Acknowledgements This work was supported by the National Science Foundation under Grant NSF CCF 10-16959, and was partially funded by the European Commission under the FIRE SCAMPI (FP7- IST-258414) project.

References

1. Foster, F.: On the stochastic matrices associated with certain queuing processes. The Annals of Mathematical Statistics **24**(3), 355–360 (1953)
2. Hajek, B.: An exploration of random processes for engineers (December 20, 2011). URL `http://www.ifp.illinois.edu/~hajek/Papers/randomprocesses.html`
3. Hajek, B., Zhu, J.: The missing piece syndrome in peer-to-peer communication. Stochastic Systems **1**(2), 246–273 (2011)
4. Han, J., Chung, T., Kim, S., Kwon, T.T., Kim, H.c., Choi, Y.: How prevalent is content bundling in bittorrent. In: ACM SIGMETRICS (2011)
5. Kingman, J.: Some inequalities for the queue gi/g/1. Biometrika **49**(3-4), 315–324 (1962)
6. Massoulié, L., Vojnović, M.: Coupon replication systems. In: ACM SIGMETRICS (2005)
7. Mathieu, F., Reynier, J.: Missing piece issue and upload strategies in flashcrowds and p2p-assisted filesharing. In: AICT-ICIW'06, pp. 112–112. IEEE (2006)
8. Menasché, D.S., de A Rocha, A.A., de Souza e Silva, E.A., Towsley, D., Meri Leäo, R.M.: Implications of peer selection strategies by publishers on the performance of p2p swarming systems. ACM SIGMETRICS **39**(3), 55–57 (2011)
9. Menasché, D.S., Massoulié, L., Towsley, D.: Reciprocity and barter in peer-to-peer systems. In: IEEE INFOCOM, pp. 1505–1513. IEEE (2010)
10. Menasche, D.S., Rocha, A.A., Li, B., Towsley, D., Venkataramani, A.: Content availability and bundling in swarming systems. In: ACM CoNEXT (2009)
11. Menasché, D.S., Rocha, A.A., de Souza e Silva, E.A., Leao, R.M., Towsley, D., Venkataramani, A.: Estimating self-sustainability in peer-to-peer swarming systems. Performance Evaluation **67**(11), 1243–1258 (2010)
12. Norros, I., Reittu, H., Eirola, T.: On the stability of two-chunk file-sharing systems. Queueing Systems **67**(3), 183–206 (2011)
13. Núnez-Queija, R., Prabhu, B.: Scaling laws for file dissemination in p2p networks with random contacts. In: IWQoS 2008, pp. 75–79. IEEE (2008)
14. Oğuz, B., Anantharam, V., Norros, I.: Stable, distributed p2p protocols based on random peer sampling. In: IEEE Allerton, pp. 915–919 (2012)
15. Wu, D., Liu, Y., Ross, K.W.: Queuing network models for multi-channel P2P live streaming systems. In: IEEE INFOCOM (2009)
16. Zhou, X., Ioannidis, S., Massoulié, L.: On the stability and optimality of universal swarms. In: ACM SIGMETRICS (2011)
17. Zhu, J., Hajek, B.: Stability of a peer-to-peer communication system. IEEE Transactions on Information Theory (2012)