P2P content sharing in spontaneous multi-hop wireless networks

Mohamed Karim Sbai
EPI Planete, INRIA, France
Email: mksbai@sophia.inria.fr

Emna Salhi
EPI Planete, INRIA, France
ENSI, Tunisia
Email: esalhi@sophia.inria.fr

Chadi Barakat
EPI Planete, INRIA, France
Email: cbarakat@sophia.inria.fr

***Abstract*—Spontaneous multi-hop wireless networks provide a new environment for sharing content among communities of end users. Nodes interested in some content can share it on a multi-hop basis by relying on the collaboration of intermediate nodes. Organizing the communication overlay in such an environment is then a challenging task especially given the mobility of the nodes, the interference, and the scarcity of resources. In this work, we propose and implement a general, stand-alone and efficient P2P content sharing protocol for wireless ad hoc networks. We consider general and realistic scenarios ranging from a single overlay to multiple overlays coexisting in the same network, and for every single overlay, we consider different peer densities in the network. The main challenges are overlay disconnections, routing overhead, low sharing opportunities and instability of distant paths. Our protocol proposes efficient solutions for these problems leading to the best download times while guaranteeing the maximum sharing opportunities among the peers. We validate our protocol through extensive simulations and experimentations on the ORBIT platform.**

I. Introduction

The proliferation of wireless devices (Laptops, PDAs, Smartphones, etc) motivates end users to connect to each other to form spontaneous communities. A multi-hop wireless network of devices, where the end-to-end communication is rendered possible by the help of ad hoc routing protocols, can be a good opportunity to share some contents (data, audio, video, etc) among the members of the same community without using any established infrastructure. But, the resources of a wireless ad hoc network are very limited and shared among the devices, which play the role of both end-users and routers. Considering this constrained nature, any content sharing application must optimize the use of the resources and distribute the content replication load equally among the set of nodes. The classical client/server architecture is not the most appropriate for wireless ad hoc networks because of the burden of the multi-hop communication and the local congestion around the server. This latter node may become the main point of failure of the application. On the other hand, application-level multicast solutions are known not to distribute the load equally among nodes and hence lack incentives for collaboration. Reliability is also an issue given the difficulty to retransmit lost packets on an end-to-end basis over the multicast tree. Considering this, content sharing applications based on the peer-to-peer (P2P) paradigm are good candidate solutions to run over spontaneous wireless ad hoc networks for the following reasons. First, they have become, in a few years, the most popular applications in the Internet and users are familiar with their functionalities and features. Moreover, a P2P file sharing solution like BitTorrent for example [1] decentralizes the data transfer plane using the multi-sourcing concept and provides enough incentives to encourage fair sharing. It is thus important to have the same principles applied in a wireless environment where nodes tend to save capacity and energy. Furthermore, multi-hop wireless communications consume resources in intermediate nodes and so there is a strong need for reducing the routing overhead by limiting the communications to neighboring peers rather than retrieving content over long network paths.

Whereas efficient content retrieval and localization techniques for wireless ad hoc networks have been widely studied in the literature [5][6], the data plane of the content sharing problem is still in its first steps. The majority of previous studies focus on the particular case of a single sharing session running over a fixed and dense wireless network. They consider that all nodes are interested in sharing the same content and they take the BitTorrent protocol [1] as reference. For instance, [2] aims to ameliorate the global download time by reducing the routing overhead. The proposed idea is to make peers only concentrate on their nearby neighbors. We show in [3] that if this is done, the replication burden is unequally distributed among peers and that there is a poor transmission parallelism in the network. This is contradictory to the goals of BitTorrent and does not respect the constraints imposed by wireless ad hoc networks. Instead, we propose to replicate pieces of the content across the network at low rate to increase the diversity of information and improve the parallelism. Although these policies register better download times and point to some new directions, they are limited to some specific cases that need to be generalized to illustrate clearly the

This work has been supported by the ITEA European project on experience sharing in mobile peer communities (Expeshare).
978-1-4244-5489-1/10/ $26.00

relationship between content replication, user performance, fairness and overhead on the underlying network. On one side there is a need to diversify the content in the network to improve user perceived quality and enforce fairness, and on the other side, this diversification is costly because of the multi-hop routing. We studied in [3] the optimal balance between sharing and diversification efforts. The study in [3] is carried out by simulations and is limited to the case of dense P2P networks with one file to share. Our main observation was that leechers (peers downloading the content) should concentrate their sharing effort on their physical neighborhood whereas seeds of the content should take into consideration the diversification of the content pieces across the network. The diversification area by one seed must be taken wider than the sharing area but no more than the distance allowed by a multi-hop TCP communication. In this paper, we generalize this first result to the case of sparse P2P networks and multiple shared files in parallel. We further consolidate this result by the means of extensive experimentations over the ORBIT platform [10]. A sparse P2P network is a sharing overlay where nodes are not necessarily all interested in the content. Some nodes are not peers but only forward packets at the routing level. The density of the overlay is the percentage of peers over the total number of nodes.

First, we consider the case of a single Torrent in the network while varying the density of peers. We study the impact of limiting the scope of sharing on the download time and the connectivity of the sharing overlay. We notice that the routing overhead is decreased at the expense of the guarantee of finishing the download. Indeed, in the most sparse cases, peers will be isolated and cannot finish downloading the content. Therefore, we propose to organize peers in a minimum spanning tree which guarantees the connectivity of the overlay and limited routing distances between peers. The NS-2 simulations [8] and the experimentations on the ORBIT platform [10] both show an improvement in the download time and the completion ratio when this strategy is deployed. Then, we move to study the impact of limiting the scope of sharing on the diversity of pieces in the network. In fact, reducing the neighborhood over the spanning tree impacts negatively this piece diversity. Pieces of the content propagate in one way from the initial seed to the edge peers resulting in low sharing ratios. By adding a diversification effort to seeds of the content, we improve considerably the sharing ratio while preserving the low routing overhead of limited P2P neighborhood. Again this will be confirmed by NS-2 simulations and ORBIT experimentations. Finally, we study the tuning of the diversification effort. Mainly, we design an algorithm to adapt the scope of the diversification area of a seed to the changes in the network settings, to congestion and to mobility of nodes. On one hand, we show that when many concurrent Torrents run in the same network, the gain brought by the diversification is negligible since the network is very congested. On the other hand, we show that there is no need for diversification in a mobile scenario since nodes will exchange different pieces of the content while moving. Simulations and experimentations show that our algorithm adapts the diversification scope in both these cases.

The remainder of this paper is organized as follows. Section II presents the background, the motivations and the main findings of our work and describes our validation methodology. Section III studies the impact of reducing the neighborhood on both connectivity and the routing overhead. It shows the benefits of using our minimum spanning tree strategy in case of sparse Torrents. Section IV investigates the gain obtained by diversifying the pieces of the content in the network. In Section V, we propose an adaptive algorithm that selects automatically the optimal scope of diversification in case of multiple Torrents and in case of different mobility speeds. The conclusions and the perspectives of this paper are dressed in Section VII.

II. Background, Motivations, Contributions and Methodology

In this section, we describe the background, the motivations and the main finding of our work and the methodology followed in our investigation and validation.

A. Background

BitTorrent [1] is a scalable and efficient P2P content replication protocol. Peers interested in sharing the same content play the role of servers for the other peers. Indeed, each peer shares some of its upload capacity with other peers in order to increase the global system capacity. Peers cooperating together to download a content form a sharing overlay called *Torrent*. To facilitate the replication of the content in the network and to ensure multi-sourcing, content is subdivided into a set of pieces. A peer having all pieces of the content is called a *seed*. When the peer is downloading pieces, it is called a *leecher*. Among the members of the Torrent, neighbors are those with whom a peer can open a TCP connection to exchange data and information. Only four simultaneous outgoing *active* TCP connections are allowed by the protocol. These neighbors are called effective neighbors. They are selected according to the *choking algorithm* of BitTorrent. This algorithm is executed periodically and aims at identifying the best uploaders. Once the choking period expires, a peer chooses to unchoke the 3 peers uploading to him at the highest rate. This strategy, called *tit-for-tat*, ensures reciprocity and enforces collaboration among peers. Now to discover new upload capacities, a peer chooses randomly a fourth peer to unchoke. All other neighbors are left choked. When unchoked, a peer selects a piece to download using a specific piece selection strategy. This strategy is called *local rarest first*. Indeed, when selecting a piece to download, a peer chooses the piece with the least redundancy in its neighborhood. Rarest first is supposed to increase the diversity of pieces [7].

Each BitTorrent's client periodically contacts a central server called *Tracker* to get an up-to-date list of the members of the Torrent. The peers in this list form the neighborhood of the peer. These peers are selected randomly and independently of any information on their locations and the performance of connections among them. For the Internet, this can be a good strategy but in the case of a wireless ad hoc network, the selection of the neighbors must be modified to adapt to the constraints of such networks.

B. Motivations and summary of contributions

Spontaneous wireless multi-hop networks are an adequate field for content sharing among communities of users. Indeed, users can connect to each other in order to share data and multimedia files without being connected to any infrastructure network. To ensure this connection at the data transfer layer, they need to agree on a content distribution protocol. The classical data transfer methods namely the client/server and the application level multicast methods are not the most suitable for wireless ad hoc networks for many reasons. First, they yield important overheads on the underlying wireless network as the communication scheme is not designed for networks where resources are limited and shared. Moreover, the load of data transfers are not fairly distributed among the set of nodes since the nodes that are nearer to the source of the content will send more packets than other nodes that are far from it. The target of these methods is to have a hierarchy of nodes where some of them sacrifice some of their capacities to serve others without any incentives built in the protocol. Hence, a suitable content sharing paradigm must minimize the consumption of network resources and must divide the burden of sharing data equally among the set of nodes by thinking about the topology of the network and giving enough incentives for fair sharing. Furthermore, it must maximize the global capacity of the system by using the ability to have parallel communications in different areas of muti-hop wireless networks.

Having these goals in mind and starting from the well-known P2P file sharing paradigm in the Internet where a peer uploads to other peers as much as it receives, we adapt in this work this paradigm to the constraints and the nature of wireless ad hoc networks. Our objective is to come up with a general, stand-alone and efficient solution for content sharing in wireless ad hoc networks. The construction of the content sharing overlay in the Internet version of BitTorrent is done independently of the underlying topology and can engender a big routing overhead in a wireless ad hoc environment. In this work, we study the best neighborhood selection strategy that suites the wireless multi-hop environment. In a first part of this work, we study the case of a single sparse Torrent in the network while varying the density of peers. In fact, only a subset of the nodes of the wireless ad hoc network can be interested in downloading/uploading the same content. Hence, these nodes will be peers and the remaining ones will be simple nodes forwarding packets to others at layer 3. The percentage of peers in the network is called in our study the Torrent density. We take the well-known BitTorrent protocol as a baseline and show through simulations and experimentations that the best neighborhood selection strategy, which guarantees both the best download times and the best sharing opportunities among peer nodes, is the strategy that limits the sharing scope of the classical Internet version of BitTorrent. However, this amelioration of the download time is at the expense of fewer peers that complete the download, particularly in the case of very low Torrent densities. In fact, limiting the sharing to nearby peers engenders less routing overhead. Unfortunately, when the Torrent is very sparse, peers that are far from each other will be isolated in separate islands and will never finish the download. The neighboring sharing area should be extended in this latter case to ensure connectivity of peers. A tradeoff then emerges between reducing the sharing area to reduce overhead and increasing it to ensure connectivity.

The solution we propose in this paper is to organize peers in a minimum spanning tree and define the neighborhood of a peer as being its neighborhood over the logical tree rather than its physical neighborhood. We choose the minimum spanning tree for two main reasons. First, this tree is a structure that limits the number of hops between peers, which translates to less routing overhead. Second, this tree has the advantage of being a structure that connects all the peers together, hence all peers can complete the download of the file. We evaluate the gain brought by constructing the P2P neighborhood over this minimum spanning tree connecting peers through simulations and experimentations.

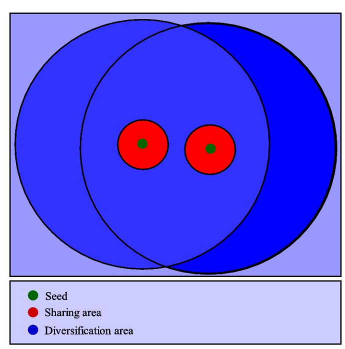

Fig. 1. Sharing and diversification areas

In a second part of this work, we study the impact of diversifying the content beyond the neighborhood scope of a seed. In fact, reducing the neighborhood over the spanning tree impacts negatively the piece diversity in the network (See

The scope of a peer, whether a seed or a leecher, is by definition the peers located within a certain physical hops around it

Figure 1). Pieces of the content propagate in one way from the initial seed to the edge peers resulting in low sharing ratios. By applying the piece diversification strategy, which we designed in [3] for the case of very dense networks to our spanning tree strategy, we improve considerably the sharing ratio while preserving the low routing overhead of limited P2P neighborhood. Following this strategy, seeds of the content will devote some of their upload capacities to send pieces of the content to peers outside the limited sharing area. The fourth outgoing connection of a seed is then aimed to select a random leecher in the diversification area. This area is limited to some hops around the seed. Whenever many seeds ensure this diversification effort in the same area, this can again engender a considerable routing overhead. We propose to share the diversification effort among the seeds of the same area. A seed selects a peer in the diversification area during one choking slot out of a number of slots equal to the number of sources existing in its diversification area. Applying this diversification strategy, the sharing opportunities among peers are boosted without overloading the network with diversification traffic. Hence, we can expect both better download times and sharing ratios.

In a third part of this work, we study the tuning of the scope of diversification. As the optimal scope of diversification depends on network settings, the speed of nodes and the amount of concurrent traffic, we design an adaptive algorithm that seeds can use to select the appropriate scope. This algorithm decides on the diversification scope upon observing the number of pieces the seed can successfully send to distant peers. Our main finding is that when the number of Torrents becomes important, connections between distant peers suffer from very bad performances. In particular, will suffer any connection the seed will open with distant peers with the hope to diversify the content. Hence, the diversity of content obtained by this effort will be negligible. We show that, in this case of many concurrent Torrents, it is better to concentrate on nearby neighbors and to reduce the scope of the diversification area to a few number of hops around the seed. On contrary, when the number of Torrents is small, pieces manage to get through the network and so the diversification area is increased to converge to one that suits the current topology of the network. We validate our adaptive algorithm in the multi-Torrent case through both simulations and experimentations and show that the best download times and sharing opportunities are recorded. Finally, we discuss through simulations the case of a mobile network. We mainly show that mobility increases the diversity of pieces and decreases the stability of long paths. Our adaptive diversification mechanism coupled with our minimum spanning tree strategy guarantee the best performances in terms of download time and sharing ratio in this case too. Similar to the case of many Torrents, the diversification area will be reduced because of path instability, but with the main difference that the diversification obtained by mobility will increase sharing opportunities.

C. Scenarios and Methodology

To ensure the validation of our proposed protocol and to compare different possible approaches in different scenarios, we conduct extensive simulations and extensive experimentations. By varying the settings of our scenarios we aim at making our results the most general possible.

- **Simulations:** we extend the NS-2 network simulator [8] by adding a general and tunable content sharing module, which is based on the algorithms of the well-known Internet protocol BitTorrent. Using our implementation, one can change different strategies of BitTorrent mainly the neighbor selection strategy and the choking algorithm. In addition to the data transfer plane, our module implements a peer discovery mechanism. This mechanism emulates for the BitTorrent client the existence of a centralized tracker providing it with the list of Torrent members. Furthermore, our module profits from the existing NS-2 modules to ensure wireless communication and multi-hop routing of packets. The wireless ad hoc network that we are simulating consists of 50 nodes randomly distributed in a $500m \times 80m$ square area. In our simulations, we discard all the realizations where the topology is not connected at the physical level. Nodes connect to each other using the 802.11 MAC Layer with the RTS/CTS-Data/ACK mechanism enabled. The data rate is set to 1 Mb/s and the wireless range to 50m. Without loss of generality, the ad hoc routing service is ensured thanks to the DSDV proactive protocol. At the beginning of each simulation and for each Torrent a random node is chosen as the seed of the corresponding content and another random set of nodes are selected as leechers. The number of peers is a parameter of the simulations and can be computed from the Torrent density. We vary the number of Torrents when we evaluate the multiple-Torrents scenario. Each content is a 10 Mbytes data file that is subdivided into 100 pieces. All peers start downloading the file at the same time (a flash crowd scenario). The BitTorrent choking period is set to 40s.
- **Experimentations:** In parallel to the NS-2 simulations, we conduct extensive experiments over the ORBIT wireless testbed [10]. We use and modify LibTorrent the open-source library and implementation of the BitTorrent protocol [13] to compare the performance of the different variants of the protocol implementing different neighborhood selection strategies. In each experiment, we randomly select 100 nodes among the 400 nodes of the ORBIT testbed. Each node in this testbed is a PC equipped with Atheros AR5002X Mini PCI 802.11a/b/g wireless card attached to an omnidirectional antenna. We configure the wireless interface card to operate in 802.11b ad hoc mode, and set the transmission power level at 20 dBm, and the bit-rate at 11Mbps. Each node in the testbed runs Linux Debian kernel v2.6.22, Mad-Wifi v0.9.3.3 [11], the OOLSR open source implementation of

the OLSR routing protocol [12] and the modified BitTorrent protocol. At the beginning of each experiment and for each Torrent considered, one of the nodes of the constructed ad hoc network plays the role of the initial seed and a randomly selected sub-set of the remaining nodes play the role of leechers. Each content is a 100 MB file. The number of peers participating in the sharing session for each file is determined by the Torrent density, a parameter of the experimentation.

As the membership management is out of the scope of this paper, each of the peers of an experiment is deterministically informed about the other peers participating in the same sharing session as him. We leave the implementation and experimentation of a more intelligent membership management plane for future research. For instance, one possible solution for this membership management is proposed in [4]. The solution is designed to be fully decentralized, network-friendly, and aware of the constrained nature of wireless ad hoc networks, namely the frequent topology changes and the network partitioning problem due to the mobility of nodes. [4] proposes to organize peers interested in sharing the same content in a shared minimum spanning tree over which global information on the Torrent, as the IDs of peers, are exchanged. All peers of the Torrent are involved in the construction and the maintenance of the tree using only their local routing information provided by the underlying routing protocol. Note that the minimum spanning tree construction of this latter solution is relevant to our present work, where a minimum spanning tree is used to define logical neighborhood for the sparse Torrent scenario.

III. Connectivity versus routing overhead

In this section, we focus on the data plane of content sharing in spontaneous multi-hop wireless networks. We mainly consider the case of a single Torrent in the network. As one of the main objectives of a content sharing application is to minimize the download time of peers, we compare the average download time of peers that have completed the download for different neighbor selection strategies by considering different Torrent densities. Table I defines the parameters used in the description of the strategies.

TABLE I
Parameters of the neighbor selection strategies

Parameter	Description
RS	the scope of the neighborhood of a peer at the routing level
OS	the scope of the neighborhood of a peer at the spanning tree level (i.e. overlay)
$OS1$	the routing distance to the node located at $OS = 1$ that is leading to the peer.
DR	the scope of the diversification area at the routing level

The compared strategies are the following:

- $RS = i$**:** This means that the scope of the neighborhood of a peer is limited to i routing hops. In other words, a peer cannot communicate with other peers located at a distance more than i hops. The classical BitTorrent strategy is topology unaware. It can be seen as a one where the neighborhood scope is taken equal to the maximum number of hops in the network. This maximum is equal respectively to 12 hops and 15 hops in our simulations and experimentations.
- $OS = 1$ **or** $RS \leq OS1$ **:** This strategy considers that the peers interested in sharing the same content are organized in a minimum spanning tree in terms of the number of hops. The construction of the minimum spanning tree can be done for example by the membership management protocol proposed in [4]. According to this strategy, a peer selects all peers located at one logical hop in the tree as its neighbors and adds all peers located at a routing distance (layer 3 distance) shorter than the number of routing hops to 1-hop neighbors in the spanning tree.

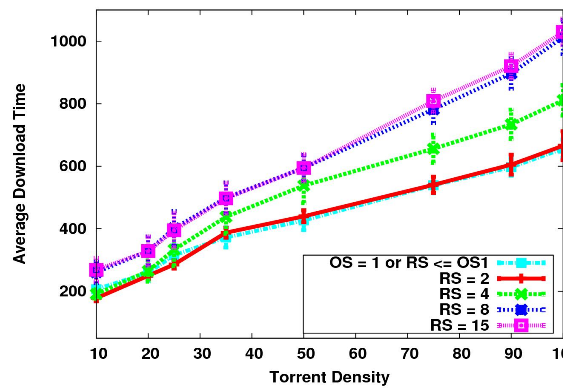

Fig. 2. ORBIT experimentations: Download time vs. Torrent density

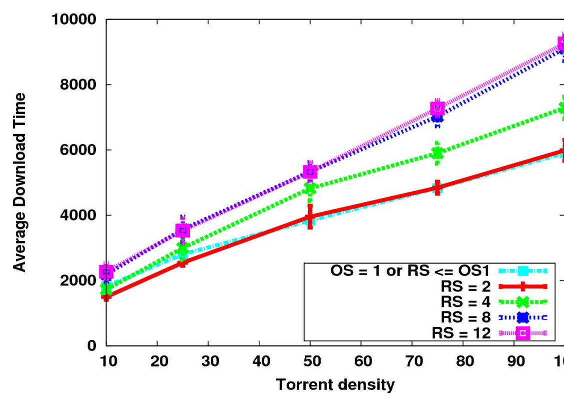

Fig. 3. NS-2 simulations: Download time vs. Torrent density

In Figures 2 and 3, we plot respectively the simulation results and the experimentations results for the average download time as a function of the Torrent density for the different strategies mentioned earlier. From these figures, one can make the following observations:

- The average download time increases with the increase of the Torrent density for all considered strategies. The maximum download time is reached for the 100% Torrent density case.

In fact, the stress on the underlying network increases with the number of peers.

- The classical strategy of BitTorrent ($RS = 12$ in simulations and $RS = 15$ in experimentations) yields the highest download times particularly when the overlay is dense. This is mainly due to the important routing overhead it engenders. In fact, following this strategy, all nodes communicate with each other independently of their locations. Intermediate nodes will be relaying packets at the routing layer without profiting of them at the content sharing layer. From here came our idea to test other strategies where the scope of the neighborhood is reduced.
- For a reduced scope strategy like $RS = 2$ (in both simulations and experimentations), one can see that the download time decreases dramatically. This can be explained in the dense cases by the reduced routing overhead. But, in the sparse cases, this can also mean that the overlay is disconnected and that only a sub-set of peers can finish downloading the content. Globally, it is always beneficial to reduce the distance between P2P neighbors provided that the overlay stay connected.
- A minimum spanning tree is a logical structure, faithful to network topology, that minimizes the distances between peers while guarantying the connectivity. By applying the $OS = 1$ or $RS \leq OS1$ strategy, one can obtain the best download time as the figures show. This strategy indeed adapts the neighborhood selection to the density of the overlay. It is almost equivalent to RS = 2 in the dense cases but it has the further advantage that it connects far away peers in the sparse cases, hence ensuring that all peers get the file.

To underline this latter point, we plot in Figure 4 (or in Figure 5) for the same experiments (or simulations) the average completion ratio as a function of the Torrent density . One can easily notice that the completion ratio of the classical version of BitTorrent is always equal to 100% as all peers are neighbors of each other. This is at the expense of larger download times as we have already seen. Now, when we limit the scope of the neighborhood, mainly in the case of sparse Torrents, the completion ratio decreases subsequently. Peers start to be disconnected from each other and from the initial seed of the file. The minimum spanning tree strategy has the main advantage of keeping the overlay connected which results in a completion ratio reaching 100%. On the other hand, the tree strategy optimizes the download time by limiting the routing overhead as we have seen.

IV. Piece diversification increases sharing opportunities

In the previous section, we concluded that the best way to select neighbors is to follow the minimum spanning tree strategy as this ensures the connectivity of the overlay and yields the best download times. In this section, we consider another important metric, the sharing ratio, which measures

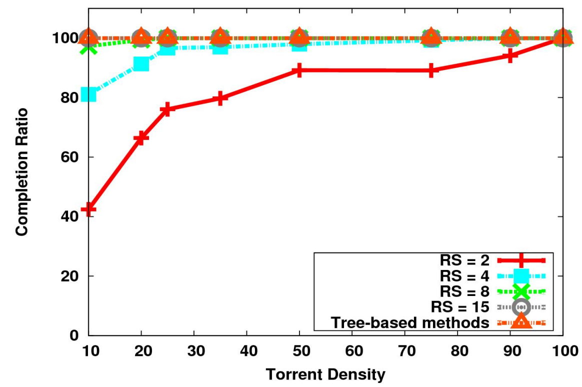

Fig. 4. ORBIT experimentations: Completion Ratio vs. Torrent density

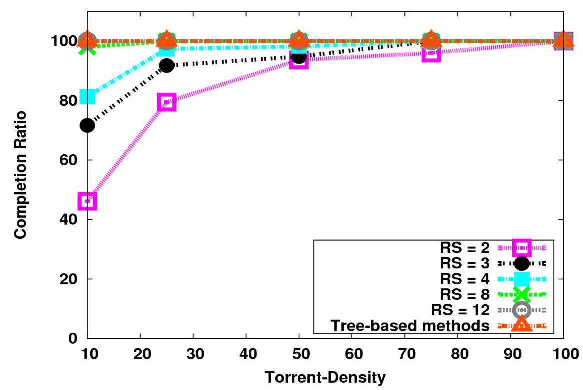

Fig. 5. NS-2 simulations: Completion Ratio vs. Torrent density

the degree of cooperation of peers. An ideal content sharing protocol must divide the sharing load fairly among the set of peers. Differently speaking, a peer should upload the same amount of data it downloads. We define the sharing ratio between a couple of peers i and j as follows:

$$R_{ij} = \frac{min(D_{ij}, D_{ji})}{max(D_{ij}, D_{ji})} \tag{1}$$

, where D_{ij} is the amount of data that peer i downloads from peer j during the Torrent lifetime. This ratio measures the magnitude of the reciprocity of data between the two peers. A value nearing null means a one-way propagation of data. The fair sharing ideal case is obtained when the sharing ratio is equal to 1. Figures 6 and 7 plots the average sharing ratio per peer as a function of the Torrent density for respectively the same experimentations and simulations as in the previous section. From these figures, one can easily notice that when the scope of the neighborhood is reduced, the lowest sharing ratios are recorded. This is mainly due to the fact that the pieces of the content would propagate in one way from the initial seed to far away peers. Hence, there would be not enough diversity of pieces to engender fair exchanges. In this case, the load of sharing is not equally distributed among nodes and the network parallelism is not fully used.

To counter this problem, we introduce a new choking

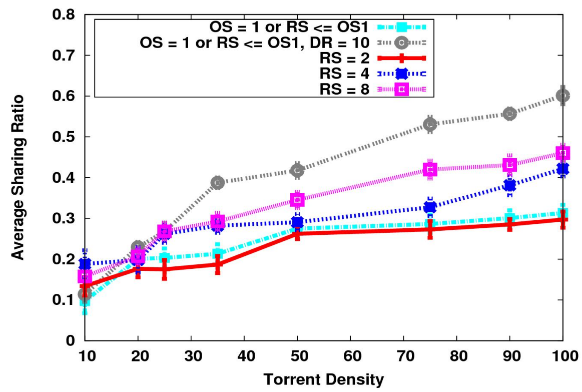

Fig. 6. ORBIT experimentations: Sharing Ratio vs. Torrent density

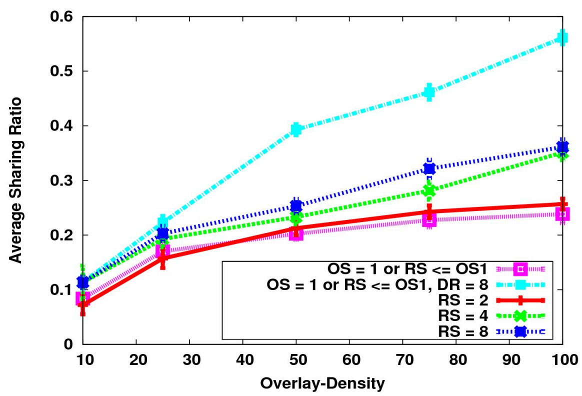

Fig. 7. NS-2 simulations: Sharing Ratio vs. Torrent density

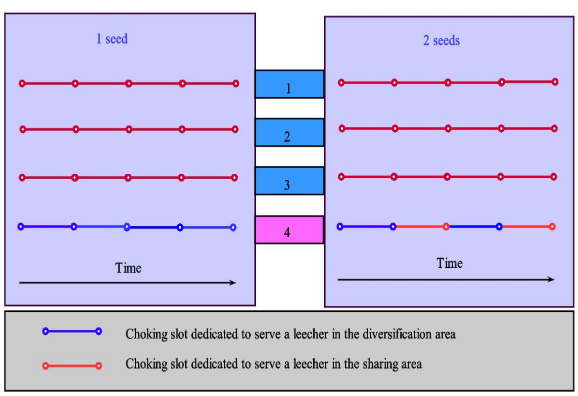

Fig. 8. Scheduling the 4th connection of a seed

strategy that allows better sharing opportunities while keeping a very low routing overhead. We propose that the fourth connection of a seed is used from time to time to serve peers that belong to an area wider than the simple close sharing neighborhood. This area is called *the diversification area* and is delimited by the diversification routing scope (DR)(see Figure 1). When it is the turn of a diversification slot on the optimistic unchoking connection, a peer is selected randomly in this diversification area and a piece is given to it following the rarest first algorithm of BitTorrent. The scheduling of the diversification slots depends only on the number of seeds in the same diversification area (see Figure 8). Note that only seeds carry out diversification since there are no incentives for leechers to do that. Moreover, seeds reduce the rate of diversification slots when other seeds appear in the same diversification area so to share the effort. Using this new strategy, one can gain in sharing ratios since more piece diversity is injected into the network. This is confirmed in Figures 6 and 7 (the line with DR set). To further validate this claim, we plot in Figure 9 the piece diversity metric as a function of time for the spanning tree strategy, with and without the diversification mechanism. The diversity metric is defined as the average number of original pieces possessed by each peer compared to its neighbors, normalized by the total number of pieces. The figure shows that the piece diversity is very low when there is no diversification and is much higher when seeds apply the diversification strategy. Clearly, the diversity metrics drops to zero when the Torrent approaches its end.

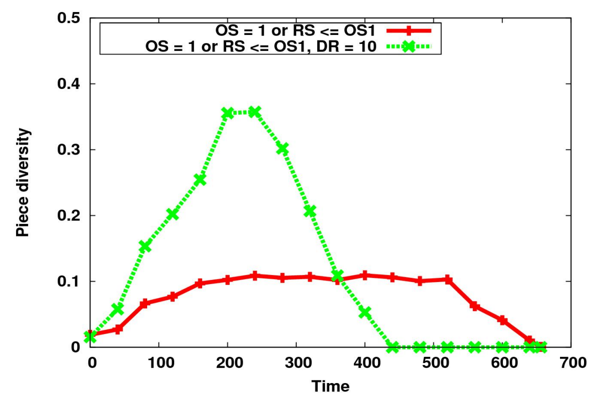

Fig. 9. ORBIT experimentations: Piece diversity

Using the spanning tree strategy with the diversification mechanism engenders good sharing opportunities. However, one needs to verify if the routing overhead is not important and the download time is still low. In Figures 10 and 11, we show the download time as a function of Torrent density for both strategies, with and without diversification. Surprisingly, it shows that there is even a considerable gain in the download time caused by the gain in piece diversity. This can be explained by the increase in network parallel transmissions due to the increase in piece diversity. Peers across the network now have data to exchange with each other instead of waiting to get them from their upstream peers.

V. Adaptive selection of optimal diversification scope

Up to now, we have not justified the selection of the scope of the diversification area. The results shown in Section IV are in fact those of the best scopes ($DR = 8$ in our simulations and $DR = 10$ in our experiments). Figures 12 and 13 plot the download time as function of the diversification scope for the case of 100% Torrent density. For small value of this

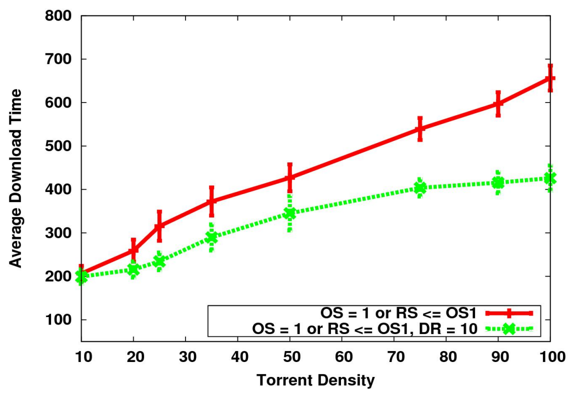

Fig. 10. ORBIT experimentations: Download time vs. Torrent density

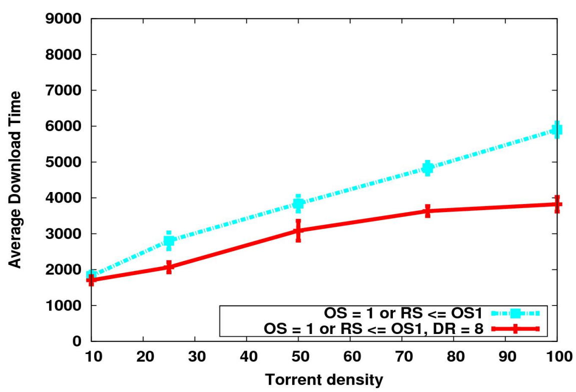

Fig. 11. NS-2 simulations: Download time vs. Torrent density

scope, not enough diversity is introduced into the network. For large values of DR, the connections to far away peers fail to send complete pieces and hence the gain in diversity is null in addition to wasting resources. A close investigation has allowed to conclude that the best value of the scope is the maximum number of hops that allows the transfer of a complete piece in one slot. Its automatic selection and adaptation to network congestion and path instability will be discussed in the following paragraphs.

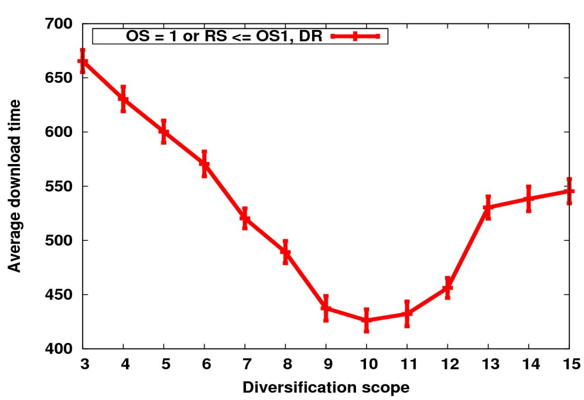

Fig. 12. ORBIT experimentations: Download time vs. Scope of diversification

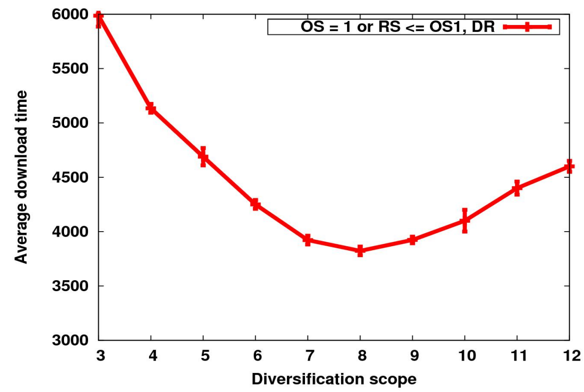

Fig. 13. NS-2 simulations: Download time vs. Scope of diversification

A. Adapting the diversification scope to network congestion

In this paragraph, we study the impact of having many Torrents together in the same network. Our aim is to evaluate the impact of the inter-Torrent congestion on the neighborhood selection strategy, mainly on the diversification effort. We vary the number of Torrents concurring in the network from 1 to 5 setting each Torrent's density to 50%. We plot in Figure 14, the average download time per peer versus the number of Torrents for the two versions of the spanning tree strategy: with and without diversification. The diversification scope is set to 10 routing hops, which is the best one in the case of a single Torrent (Figure 12). One can notice that when there is a few number of Torrents (1 or 2), the diversification yields a decrease in the download time. Whereas, for a larger number of Torrents, the download time worsens and it is better not to diversify. This is because when the network gets congested by other Torrents, pieces can no longer be sent over several hops as in the case of one Torrent. Insisting on sending them wastes resources and impairs the download time. It is better to artificially decrease the diversification area of seeds in this case. Furthermore, all the nodes will be busy sending or receiving some pieces and the network will be fully used by the different Torrents, so there will be no gain from enforcing parallel network utilization by means of piece diversity.

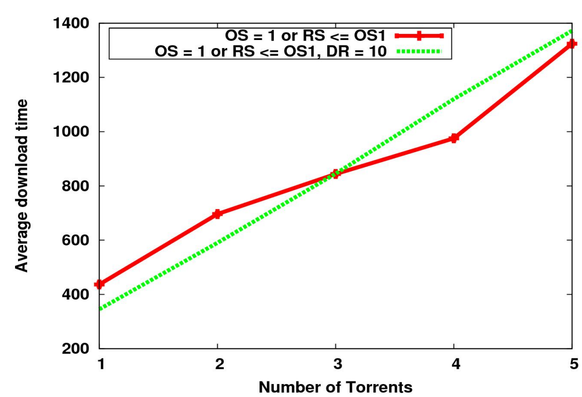

Fig. 14. ORBIT experimentations: Download time vs. Number of Torrents

As we want our protocol to be general and adaptive, we propose to adjust the scope of the diversification area automatically as a function of network conditions. We measure in each diversification slot the number of pieces sent to the selected peer. If no complete piece has been sent, this means that the path is congested and that one needs to decrease the diversification scope. And if one or more pieces can be sent, this means that one can increase the diversification scope, hopefully reaching farther peers. The algorithm that we propose to adjust this diversification scope is the following:

- At the beginning of the sharing session, DR is set to an average value equal to $\frac{(3+h_m)}{2}$, where h_m is the maximum number of hops in the network. It is a first guess of the diversification scope. It is taken equal to an average value since the scope of diversification can range from 3 hops to $\frac{h_m}{2}$ hops.
- Each time a peer sends diversification packets to a node located at h hops ($h <= DR$), it updates the value of DR depending on whether it can send a complete piece to the destination or not.
- In case more than one complete piece can be sent, DR is increased to $\alpha \times DR + (1-\alpha) \times (DR+1)$, where α is an empirical value chosen in our experiments equal to $\frac{3}{4}$.
- In case no complete piece can be sent, DR is decreased to $\alpha \times DR + (1-\alpha) \times (h-1)$, where α is the same empirical value chosen in our experiments equal to $\frac{3}{4}$.

The objective of any adaptation of this type is to absorb transitory network congestion while allowing fast convergence to the appropriate diversification scope. Figure 15 plots the download time as function of the number of Torrents for the spanning tree strategy with our adaptive scope of diversification. One can easily notice that it converges to the best download times of the two previous versions (with and without diversification in Figure 14). Hence, this new adaptive strategy is aware of the inter-Torrent congestion problem and can easily absorb other transitory congestions or troubles in the path. Figure 16 plots the diversification scope of our algorithm in the steady state (after convergence) as a function of the number of Torrents. Clearly, the diversification scope has a decreasing trend with the number of Torrents. In particular, we notice that for the case of one Torrent, our adaptive algorithm converges to a $DR = 10$, which has been shown in Figure 12 to be the best value to use. For two Torrents, it is better to shrink slightly the diversification area to have less congestion and routing overhead. If one continues increasing the number of Torrents, the diversification scope will converge to its minimum value 2, which is equivalent to no diversification. As a conclusion, we propose to build the neighborhood using the spanning tree strategy with an adaptive scope of diversification. This strategy provides the best performances in both the single Torrent and the multiple Torrent cases.

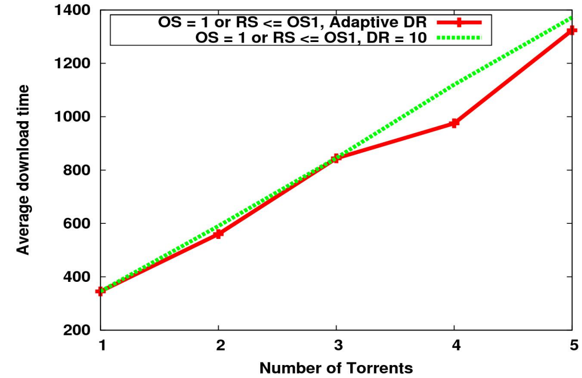

Fig. 15. ORBIT experimentations: Download time vs. Number of Torrents

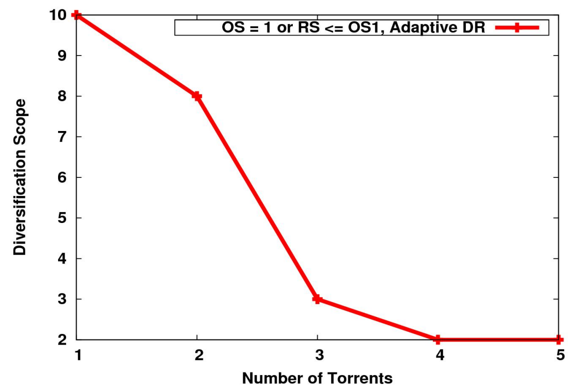

Fig. 16. ORBIT experimentations: Diversification scope vs. Number of Torrents

B. Adapting the scope of diversification to mobility of nodes

In the previous sections, we supposed that the network is fixed. In case of mobility of nodes, two main factors must be considered. On one hand, the mobility increases naturally the diversity of pieces since the neighborhood of a peer is changing while moving. In this case, one can hope that there will be enough sharing opportunities and hence will be no need for sending pieces to far away nodes to boost diversity. On the other hand, as long paths suffer from high failure rates in mobile ad hoc networks, our adaptive diversification algorithm should converge to the case of no diversification when mobility increases. To validate these claims, we simulate a mobile network scenario where nodes move following the well-known Random Waypoint [9] model. In this model, each node moves along a zigzag line from one waypoint to another. The speed of nodes is taken equal to 2m/s, a typical pedestrian speed. For the pause time, different values were considered, we show here the results for 2s. The remaining parameters are set as described in Section II-B. The dynamic minimum spanning tree is computed by our membership management protocol described in [4]. The download time and sharing ratio in case of a 100% dense Torrent for different strategies are summarized in table II. Clearly and as expected, the first and third strategies lead to close performances, which means that artificial piece diversification is not needed in this mobile scenario (since it is inherent) and that the adaptive algorithm captures this operation mode. The spanning tree strategy with the adaptive scope of diversification is then the right strategy to use in a mobile environment as well.

TABLE II
IMPACT OF MOBILITY OF NODES

Strategy	Download time	Sharing ratio
$OS = 1$ or $RS <= OS1$	321s	0.71
$OS = 1$ or $RS <= OS1$, $DR = 8$	435s	0.57
$OS = 1$ or $RS <= OS1$, Adaptive DR	335s	0.68

VI. SUMMARIZING THE PROPOSED CHOKING ALGORITHM

```
Initializing diversification scope
DR = (MaxHops + 3) / 2

Upon each choking slot:

 For all peers:
Sharing Area = { leechers with a routing scope less than maximum
hops to neighbors located at one hop on the spanning tree.}
Diversification Area = {leechers not located in Sharing Area and
located at a number of hops less than the diversification scope}

 Leecher behavior:
1- select 3 neighbors from Sharing Area that are the best
uploaders in the previous choking slot
2- select 1 neighbor randomly from Sharing Area

 Seed behavior:
1- select 3 neighbors randomly from Sharing Area
2- If it is a diversification slot:
   select 1 neighbor randomly from Diversification Area
   Else
   select 1 neighbor randomly from Sharing Area
   End if
3- Update DR observing the rate of sending complete pieces
to peers in Diversification Area
```

Fig. 17. Pseudo-code of the final chking algorithm

Figure 17 gives a pseudo-code of the proposed choking algorithm. It shows two main behaviors: the behavior of leechers and the behavior of seeds. A leecher runs the classical BitTorrent's choking algorithm in its sharing area. This area is delimited by nodes located at one-hop in the minimum spanning tree ($RS <= OS1$). Whereas a seed, from time to time, uses its fourth connection to serve nodes located at an area wider than its sharing area. This area is called the diversification area. The effort of diversification is divided between seeds. A seed considers one slot among a number of slots equal to the number of seeds in its diversification area as a diversification slot. The scope of the diversification (DR) area is adapted by observing the rate of sending complete pieces to nodes in the diversification area.

VII. CONCLUSIONS AND PERSPECTIVES

Spontaneous multi-hop wireless networks offer good opportunities to share content among communities of users. In this work, we propose a general, stand-alone and efficient content sharing protocol for wireless ad hoc networks. The provided solution is a BitTorrent variant that adapts itself to the constrained nature of ad hoc networks and to the underlying topology. Considering the spectrum from sparse to dense Torrents, our first objective was to design a neighborhood selection mechanism that reduces the routing overhead and hence the resources consumption, while guaranteeing the connectivity of the sharing overlay. Furthermore, we added a diversification mechanism that increases the sharing opportunities among the nodes and improves further the download time. We concluded by considering the realistic scenario of multiple simultaneous Torrents and designing an adaptive diversification algorithm that takes into consideration the network load caused by other Torrents. Extensive simulations and ORBIT experimentations were carried out to support the study and prove the outperformance of our solution compared to a standard BitTorrent-like solution. We mainly gain in download time and record the best sharing ratios. As a future work, we will be looking if synergy between Torrents can help ameliorating the performance of content sharing. More mobility scenarios will be considered coupled with a modeling effort to prove the optimality of the diversification area estimator.

ACKNOWLEDGEMENTS

We are thankful for Byrav Ramamurthy from the University of Nebraska-Lincoln and for the anonymous reviewers for their valuable feedback.

REFERENCES

[1] BitTorrent protocol. http://wiki.theory.org/BitTorrentSpecification.
[2] Michiardi P., Urvoy-Keller G., ”Performance analysis of cooperative content distribution for wireless ad hoc networks”,in WONS 2007, Obergurgl, Austria.
[3] Mohamed Karim Sbai, Chadi Barakat, ”Revisiting content sharing in wireless ad hoc networks”, in proceedings of the fourth workshop on self-organizing systems (IWSOS), Zurich, December 2009
[4] Mohamed Karim Sbai, Emna Salhi, Chadi Barakat, ”A membership management protocol for mobile P2P networks”, in proceedings of the ACM Mobility Conference, Nice, September 2009.
[5] A.Klemm, C.Lindermann, O.Waldhorst, ”A special-purpose peer-to-peer file sharing system for mobile ad hoc networks”, VTC 2003.
[6] S.M. Das, H.Pucha, Y.C. Hu. ”Ekta: an efficient peer-to-peer substrate for distributed applications in mobile ad hoc networks.” TR-ECE-04-04, Purdue University,2004.
[7] A. Legout, G. Urvoy-Keller, P. Michiardi, ”Rarest First and Choke Algorithms Are Enough”, in IMC 2006, Rio de Janeiro.
[8] NS: The Network Simulator, http://www.isi.edu/nsnam/ns/
[9] T. Camp, J. Boleng, and V. Davies, A survey of mobility models for ad hoc network research, WCMC, vol. 2, no. 5, pp. 483502, 2002.
[10] Rutgers ORBIT project team. http://www.orbit-lab.org
[11] MadWifi Project Team. http://madwifi.org
[12] OOLSR project, http://hipercom.inria.fr/olsr/
[13] LibTorrent's web site, http://libtorrent.rakshasa.no/