A Survey on Energy Efficiency in P2P Systems: File Distribution, Content Streaming and Epidemics

SIMONE BRIENZA, University of Pisa
 SENA EFSUN CEBECI, Koç University
 SEYED SAEID MASOUMZADEH, University of Vienna
 HELMUT HLAVACS, University of Vienna
 ÖZNUR ÖZKASAP, Koç University
 GIUSEPPE ANASTASI, University of Pisa

Several Peer-to-Peer (P2P) protocols and applications have been developed in order to allow file distribution/sharing, video and music streaming, data and information dissemination. These P2P systems are regularly used by a large number of users, both in desktop and mobile environments and generate a remarkable portion of the overall Internet traffic. However, many common P2P protocols and applications were designed neglecting the energy problem. In fact, they often require always-on devices in order to work properly, thus producing significant energy wastes. The problem is even much more relevant in the mobile context, since the battery lifetime of mobile devices is limited. Therefore, energy efficiency in P2P systems is a highly debated topic in the literature. New P2P approaches – more energy efficient than traditional client/server solutions – have been proposed. In addition, several improvements to existing P2P protocols have been introduced in order to reduce their energy consumption. In this paper, we present a general taxonomy to classify state-of-the-art approaches to the energy problem in P2P systems and applications. Then, we survey the main solutions available in the literature, focusing on three relevant classes of P2P systems and applications, namely *file sharing/distribution*, *content streaming* and *epidemics*. Furthermore, we outline open issues and provide future research guidelines for each class of P2P systems.

Categories and Subject Descriptors: **C.2.4 [Computer-Communication Networks]:** Distributed Systems

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: peer-to-peer, P2P, energy efficiency, file distribution, content streaming, epidemics.

1. INTRODUCTION

Peer-to-Peer (P2P) systems have gained an increasing importance in recent years. *P2P file-sharing* applications are among the most popular Internet services accounting for a large fraction of the overall Internet traffic [Carlinet et al. 2010; Schulze and Mochalski 2009]. They are daily used by a large number of users to locate and download media files such as books, music, movies, and games. In addition to P2P file sharing, *P2P content streaming* services such as *Video on Demand* (VoD) and *Live Video* streaming are also quite popular [Kurose and Ross 2013a]. Finally, several distributed services such as reliable multicasting [Birman et al. 1999; Ozkasap et al. 2009c; Eugster et al. 2003], aggregate computation [Jelasity et al. 2005], frequent items discovery [Cem and Özkasap 2013], overlay topology construction and failure detection rely on *P2P epidemic* approaches.

Originally, P2P solutions have been designed without paying so much attention to their energy demands. For instance, the most popular protocol for P2P file sharing/distribution (i.e., BitTorrent [Bharambe et al. 2005]) has no internal mechanism for energy efficiency. The main motivation for this design choice is that, in P2P systems, energy consumption is distributed among a large number of peers, differently from traditional client-server architectures where energy is mainly consumed at data

Authors' addresses: S. Brienza, and G. Anastasi, Department of Information Engineering, University of Pisa, Italy, email: simone.brienza@for.unipi.it, giuseppe.anastasi@unipi.it; S. E. Cebeci, and Ö. Özkasap, Department of Computer Engineering, Koç University, Istanbul, Turkey, email: senacebeci@ku.edu.tr, oozkasap@ku.edu.tr; S.S. Masoumzadeh, and H. Hlavacs, Research Group Entertainment Computing, University of Vienna, Austria, email: seyed.saeid.masoumzadeh@univie.ac.at, helmut.hlavacs@univie.ac.at.

centers. Hence, in P2P systems energy costs are not supported by a single organization; instead, they are shared among a large number of users. As the energy costs of P2P systems keep increasing with their popularity and share of the overall Internet traffic – and awareness about energy and environmental issues increases – developing energy efficient solutions has attracted more and more attention in recent years. In addition, due to the proliferation of mobile devices (such as smart-phones and tablets), many P2P applications and services have been made available for mobile platforms. Mobile devices are typically powered by batteries with limited energy and, hence, energy efficiency is vital in order to extend their lifetime.

Designing energy efficient P2P solutions is not a trivial task. Since energy consumption is distributed among a number of peers, addressing energy efficiency in P2P systems is generally more challenging than in traditional client-server architectures [Kant 2010]. Introducing energy efficiency in P2P audio/video streaming is even more challenging as, in addition to save energy, the *Quality of Service* (QoS) constraints imposed by the application need to be satisfied as well. Typically, energy savings and QoS provision are contrasting goals.

Despite the aforementioned challenges, a large number of solutions for energy efficiency in P2P systems have been developed in recent years. In this paper, we first introduce a general taxonomy to classify the main approaches to energy efficiency in P2P systems, applications and protocols. Then, we survey the main proposals presented in the literature, with special focus on three main areas, namely *file sharing/distribution*, *content streaming*, and *epidemic protocols*. We show that, while the solutions proposed for the three considered areas are significantly different, the basic ideas are very similar and common to all the three areas. For each considered area, we also outline open issues and provide some general guidelines for future research activities.

The rest of the paper is organized as follows. Section 2 introduces our taxonomy of energy efficiency in P2P systems and applications. Section 3 surveys the main solutions for energy efficiency in P2P file sharing/distribution. Section 4 addresses energy efficiency in P2P content streaming, while Section 5 focuses on energy efficiency in P2P epidemic protocols. Section 6 concludes the paper.

2. GENERAL CLASSIFICATION

In this section, we provide our classification of energy-aware P2P solutions based on six main classes, namely sleep/wakeup, proxy-based, peer selection, resource allocation, structure optimization and message reduction, depending on the approach they take for energy efficiency. As shown in Figure 1, our classification concerns three relevant classes of P2P systems and applications (i.e., file sharing/distribution, content streaming and epidemics). For energy efficient file sharing/distribution, there exist several solutions in the literature, based on the six main classes of approaches we identified. As shown in the figure, energy efficient content streaming solutions are based on three main approaches, namely sleep/wakeup, peer selection, and resource allocation. Energy efficient P2P epidemic solutions make use of four main approaches, namely sleep/wakeup, resource allocation, structure optimization and message reduction. Description of each approach is provided as follows.

Sleep/Wakeup. When taking a sleep/wakeup approach, the whole P2P system, or some of its components are switched off when unused, so as to save energy, and are woken up again whenever necessary to exchange data. Sleep/Wakeup solutions usually exploit features provided by the operating system and/or the network interface

Fig. 1. General Classification

(e.g., Wake On LAN [Lieberman 2006]). Shutdown and wakeup operations can be performed either autonomously by a single peer, or in coordination with other peers in the overlay network. Generally, specific schedules and policies allow to avoid degradation in the performance of the overall system while preventing single peers from repeatedly switching on/off.

Proxy-based. In proxy-based solutions, a dedicated entity (called *proxy*) inside the overlay network acts as a delegate of a number of peers and downloads data on their behalf. During the download, peers can enter a low-power mode to save energy. The proxy will wake up peers as soon as the related download is finished. This way, peers can download data from the proxy at the maximum speed allowed by the connection, thus minimizing the time they must stay active and, hence, their energy consumption. Proxies are quite common in mobile environments, where they are used to prevent peers from participating in swarms with low downloading rate. In fact, by downloading directly from the proxy, a mobile client can achieve higher bit rates and faster download times than with the classical P2P approach, thus extending its battery lifetime.

Peer Selection. When a peer requires some data, it has to select which peer to request them from. The choice of the most appropriate supplying peer should consider several aspects. For instance, if a peer has to fulfill many requests, it uses its CPU more intensively and requires more bandwidth, thus consumes more energy. Therefore, a simple strategy to save energy could be to balance requests among peers, so that none of them is overloaded. Similarly, in a mobile context where peers are directly connected (e.g. using Bluetooth [Bluetooth.org 2014] or WiFi direct [Wi-Fi Alliance 2014]), if the selected peer is far away, or the channel between the two peers is noisy, the communication requires a larger number of retransmissions, thus resulting in a higher energy consumption. In this case, selecting a near peer as a supplier could reduce the energy consumed for communication. It clearly emerges that an appropriate selection of the supplier peer may produce significant energy savings. In this perspective, solutions falling in this category rely on different selection criteria to identify the peer that is able to satisfy data/file requests with minimum energy consumption.

Resource Allocation. P2P applications require significant amount of system resources such as CPU, memory, network interfaces. By using such resources properly, it is possible to reduce the energy consumption of peer devices. To this end, several approaches can be followed. For instance, virtualization techniques can be used to efficiently allocate shared resources among the peers of an overlay network. In mobile

environments, some proposed solutions reduce the energy consumption by properly using specific hardware components. For instance, since short-range connections (e.g., WiFi, Bluetooth) are consuming less energy than long-range ones (e.g., 3G, UMTS, LTE), the former should be preferred whenever possible, for downloading a content. Similarly, in some environments it is possible to limit the energy consumption of peer devices, by appropriately tuning the transmission power of their network interfaces.

Structure Optimization. P2P overlay networks maintain some sort of structure or topology that defines which peers are connected to each other, and how they can communicate. Approaches based on structure optimization specify how such struc-

Table I. General Classification

	FILE DISTRIBUTION/SHARING	CONTENT STREAMING	EPIDEMICS
<i>Sleep/Wakeup</i>	[Blackburn and Christensen 2009] [Lee et al. 2010] [Forshaw and Thomas 2013] [Sucevic et al. 2011] [Andrew et al. 2011] [Hlavacs et al. 2012]	[Jourjon et al. 2010] [Pussep et al. 2010]	[Lefebvre and Feeley 2000] [Hou et al. 2006] [Rajeswari and Venkataramani 2010]
<i>Proxy-based</i>	[Nordman and Christensen 2010] [Anastasi et al. 2010] [Giannetti et al. 2012] [Hlavacs et al. 2012] [Jimeno and Christensen 2007] [Agarwal et al. 2009] [Kelényi and Nurminen 2010] [Kelényi et al. 2010a] [Kelényi et al. 2010b] [Kelényi et al. 2011a] [Kelényi et al. 2011b] [Kelényi et al. 2012]		
<i>Peer Selection</i>	[Mawji and Hassanein 2010] [Enokido et al. 2010]	[Li et al. 2009] [Maamar et al. 2011] [Maamar et al. 2013]	
<i>Resource Allocation</i>	[Hlavacs et al. 2012]	[Chen et al. 2011] [Chen et al. 2013] [Albiero et al. 2008] [Yeung and Kwok 2008] [Ramadan et al. 2008] [Barua et al. 2013] [Zhang et al. 2007]	[Garbinato et al. 2010] [Kontos et al. 2014]
<i>Structure Optimization</i>	[Han et al. 2008] [Trunfio 2014] [Chandrasekar et al. 2012] [Raj et al. 2013] [Leung and Kwok 2005] [Leung and Kwok 2008]		[van Renesse 2002] [Ozkasap et al. 2014]
<i>Message Reduction</i>	[Kelényi and Nurminen 2008a] [Kelényi and Nurminen 2008b] [Ding and Bhargava 2004] [da Hora et al. 2007]		[Fauji and Kalpakis 2011] [Lu et al. 2008] [Khouzani et al. 2012] [De Rango and Amelio 2013] [De Rango et al. 2013] [Lee et al. 2014]

tures should be formed. In addition, solutions called *hierarchical* allow to build a new layer on top of the P2P overlay, formed by special peers (namely *super peers*) that perform additional tasks with respect to regular peers. By properly controlling/maintaining the topology or introducing super peers, it is possible to optimize data and information flow, thus achieving significant energy savings.

Message Reduction. Message reduction approaches aim at conserving energy in P2P overlay networks by reducing the number of messages required for their operation and maintenance. Basically, peers have to exchange a smaller number of messages. This way, processing and transmission overheads inside peers are reduced, thus saving energy in the overall P2P overlay network.

Table I shows the main solutions proposed for energy efficiency in P2P systems and applications, classified according to the taxonomy introduced above. For each considered scenario, the main solutions will be discussed in the following sections.

3. FILE DISTRIBUTION/SHARING

The most popular applications of P2P overlays are in the field of file distribution and file sharing. These applications account for a large share of the overall Internet traffic, ranging from 40% up to 73%, depending on the considered geographic area [Carlinet et al. 2010; Schulze and Mochalski 2009]. It is therefore evident why energy efficiency in P2P file distribution/sharing is so important.

From the energy point-of-view, the main motivation to use a P2P approach for file sharing/distribution systems is that in a P2P architecture energy consumption is distributed among a large number of peers, while in the traditional client-server architecture energy is mainly consumed at data centers. Hence, in P2P file sharing/distribution, energy costs are not supported by a single organization, but are shared by a large number of users. However, the question that could arise is: *which architecture (P2P vs. client-server) is more energy efficient without any mechanism for controlling energy consumption?*

To answer this question, different factors contributing to energy consumption in both architectures have been considered and the most energy efficient one was investigated in [Nedevski et al. 2008]. In a P2P file sharing system, peers typically spend a high portion of the day fully powered-up, while their average utilization is typically very low. This factor plays a significant role in energy consumption of P2P file sharing/distribution. Apart from this, a P2P system makes much heavier use of the network than a client-server system. This is due to the overhead traffic generated for peer discovery and searching, as well as for membership maintenance. The average path length between peers can be significantly longer than the average path length between a client and the central server. On the other side, in a client-server architecture, the power consumed by cooling in data centers is the most important factor that contributes to energy consumption, while a P2P architecture is exempt from this cost due to its geographic distribution and the fact that peers are mostly home PCs. Another factor is the high baseline energy consumption of PCs, defined as the amount of energy that a PC consumes just to be powered up (i.e., when it is idle). The baseline energy consumption is typically much higher than the additional energy consumed by the PC when its utilization increases. The difference between P2P and client-server systems mainly lies in the fact that a P2P service is only responsible for the increase in the peer's power consumption but is not responsible for baseline energy consumption, while in client-server architecture, the data center is responsible for the entire power consumption. The authors of [Nedevski et al. 2008] considered all

the aforementioned factors in modeling energy consumption and comparing P2P and client-server architectures. They concluded that, in terms of energy consumption at end systems, P2P is likely the most convenient. However, this conclusion strongly depends on the assumption that the baseline energy consumption of peers comes for free.

In any case, the P2P architecture turns out to be actually a valid option for file sharing/distribution, also in terms of energy consumption, compared to the traditional client-server paradigm. Therefore, starting from this assumption, many solutions have been proposed to further improve the energy efficiency of P2P file sharing/distribution systems. The main solutions appeared in the literature are discussed in the following, according to the taxonomy introduced above.

3.1 Sleep / Wakeup Approaches

Sleep/wakeup solutions are a class of energy efficient techniques, in which peers are turned off and switched on in order to save energy, without any (or with a little) degradation in the system performance. The proposed approaches in this category can be further divided into two sub-classes, namely *self-adaptive* and *scheduling-based* solutions, depending on who makes the decision about the state (sleep/wakeup) of peers in the swarm.

3.1.1. Self-Adaptive Solutions. In self-adaptive solutions, peers themselves decide when to switch between sleep and wakeup mode according to their own local calculations, or in cooperation with other peers. This approach typically requires modifications in P2P file sharing applications and protocols. Therefore it is obvious that the solutions in this subcategory are achieved in the application layer. In this respect, a Green BitTorrent protocol has been proposed in [Blackburn and Christensen 2009]. In this modified protocol, peers can be turned into sleep mode without being dropped from the peer list in the tracker (i.e., the host that contains the list of all the clients in a specific swarm). This is achieved by flagging a disconnected peer as asleep, instead of removing its entry from the peer list. Therefore, a tracker can maintain knowledge of sleeping peers through the swarm. If the resources of a sleeping peer are later needed, it can be woken up to restore the TCP connection. To this end, Green BitTorrent relies on the Wake on LAN (WoL) technology [Lieberman 2006]. A similar approach is also used in [Lee et al. 2010], where an energy efficient set-top-box prototype for BitTorrent networks is proposed. Like Green BitTorrent, this solution is based on the definition of new peer states, including energy saving modes. Similarly, the communication of this state information is realized through specific hibernation and wakeup messages, and sleeping peers are resumed through the WoL technology.

In [Forshaw and Thomas 2013], the authors consider the perspective of a content provider and propose an energy efficient solution for BitTorrent-based content distribution systems. Basically, they consider a *seed pool*, i.e. a group of servers in charge of sharing content to peers, ensuring satisfactory levels of performance. This pool is said to be elastic, in the sense that the number of its elements is adapted dynamically in response to real-time service demand, so as to provide the required performance while minimizing the energy consumption of the pool. In this respect, each seed can be in a *sleep* or in an *active* state. Basically a seed can be placed in the active state, to serve the requests from the swarm, or in the sleep state, to limit its energy consumption. While asleep, the state of the server is stored in memory, allowing the machine to quickly resume into the active state.

3.1.2. Scheduling-based Solutions. These solutions rely on a given schedule to turn peers on and off, in order to decrease the energy consumption of the system and are typically achieved in the application layer. Such schedule is obtained analytically from an energy model of the P2P network. The approach is generally centralized, since the schedule is calculated by a central algorithm and peers are switched on and off accordingly.

Following this approach, a scheduling mechanism for powering down peers inside a P2P file sharing system is proposed in [Sucevic et al. 2011]. The authors present a mathematical model and an optimal strategy for small networks. They prove that, for small networks, the optimal strategy consists of turning peers off as soon as they have finished their downloads. However, they also show that the optimal strategy quickly becomes extremely complex as the number of peers grows, resulting in a substantial limitation in practical situations. Therefore, in order to study the case of large networks, the authors consider a heuristic approach, in which the peers with the highest upload capacities remain active in the swarm longer, to ease the downloads of the other peers. Simulation results show that this scheduling turning peers off according to an increasing order of capacity, reduces the average on-time (and, thus, the energy consumption) per peer, compared to other approaches (i.e., decreasing order of capacities and random order).

Based on the same idea, the authors of [Andrew et al. 2011] present a study about the energy efficiency of P2P file distribution and propose a different scheduling strategy to minimize the total energy consumption. In order to obtain the best results, they show that peers should be active only when they can download at the highest possible rate, and should be turned off as soon as they have downloaded the file. Just a small set of energy efficient peers is required to be always on in order to help others.

In [Hlavacs et al. 2012] the authors present an energy consumption model for the BitTorrent file sharing protocol, inspired by [Qiu and Srikant 2004]. Starting from this model, they derive the analytical expression of the optimal seeding time, i.e., the optimal time seeders should stay online in order to minimize the globally consumed energy of the swarm. Simulation results show that, if seeders leave the swarm and are powered off according to the derived optimal seeding time, the system energy consumption is minimized and the average downloading rate is maximized. However, the authors do not discuss how the optimal seeding time can be obtained in a real scenario so as to define a usable schedule.

3.2 Proxy-based Solutions

In proxy-based solutions an entity called *proxy* acts as a delegate of one or more peers, either to maintain the network presence of devices while they are in sleep mode [Nordman and Christensen 2010], or to prevent them from participating in swarms with low downloading rate (typically in mobile environments). These solutions are mainly achieved in the application layer, because they need to negotiate with file sharing applications at the protocol level.

In [Anastasi et al. 2010], the authors proposed a proxy-based solution, namely *EE-BitTorrent*, for the BitTorrent protocol [Bharambe et al. 2005; Kurose and Ross 2013b]. It achieves energy efficiency by allowing a number of sleeping peers to be served by a BitTorrent proxy. The latter can run either on a dedicated computer or on a computer that already provides other network services (e.g., DHCP, Web proxy, etc.). In EE-BitTorrent peers delegate the download process to the associated proxy,

which performs the task on behalf of them. Most of the time, the requested file is already available on the proxy and can, thus, be immediately transferred to the user's device. If this is not the case, the proxy participates in the BitTorrent overlay network as a regular peer, and takes care of the overall download process. Therefore, the user's device can be switched off during the download phase. The file will be downloaded from the proxy when the user's device reconnects. EE-BitTorrent does not introduce any change in the BitTorrent protocol, and is very effective when the user's uplink rate is low. In such a scenario, the legacy BitTorrent protocol performs poorly due to the Tit-for-Tat mechanism, and EE-BitTorrent outperforms it in terms of average download time and energy consumption. However, the opposite occurs when the uplink rate is good. Therefore, in this respect, the same authors have proposed an adaptive algorithm, namely *AdaBT* [Giannetti et al. 2012], being able to dynamically select the most efficient option between legacy and EE-BitTorrent, according to the operating conditions available at download time.

Download sharing [Hlavacs et al. 2012] is another proxy-based approach for BitTorrent, mainly targeted for unpopular files, i.e., files for which very few copies are available in the overlay. In this case, the number of *seeders* (i.e., peers with a *complete copy* of the file being downloaded by other peers) is very small. Hence, the P2P download tends to degenerate to a client-server file transfer that results in a high download time and energy consumption at the peer device. The basic idea of *Download sharing* is employing a subset of *leechers* (i.e., peers with a *partial copy* of the file) as the external proxies or *active leechers*, to manage the download task on behalf of other leechers (*passive leechers*), which can thus be hibernated.

Leechers with higher upstream network bandwidth are preferably selected to act as active leechers. The idea then is to combine the upload capacity of seeders and avoid semantically redundant download flows. Figure 2 shows the state cycle of a leecher. In state *P* (*Passive*), a leecher is hibernated and consumes only the power for holding the connection to the network (*P* is the starting state for all leechers). When a leecher in state *P* creates a new download task, it switches to state *AB* (*Active – Blocked*). In state *AB*, the leecher searches for other leechers being in state *A* (*Active*) with the aim to send out the task to them. If no active leecher can be found, the leecher itself switches to state *A* to execute the task locally. Also, from this time on, it accepts remote tasks generated by other leechers. Ideally, a leecher in state *A* is executor of many tasks created by different owners which are in state *P*. After the executor has finished a task, it sends back the results to the owners. If an owner is in state *P*, it must switch to state *ABC* (*Active-Blocked-Content*) for the time the upload from the executor to the owner takes. After the successful upload, the task is finally

Fig. 2. State Cycle of Leecher.

completed and the owner can switch back to state *P*.

Based on similar observations, the authors of [Jimeno and Christensen 2007; Agarwal et al. 2009] have proposed other proxy-based solutions for P2P file sharing/distribution. However, unlike the approaches described above, these solutions rely on a *private* proxy rather than an external shared entity. The proxy proposed in [Jimeno and Christensen 2007] for the Gnutella file sharing/distribution protocol [Klingberg and Manfredi 2002] operates in a low power micro-controller, co-located on the Ethernet Network Interface Card (NIC) of the user's device. The proxy supports a subset of the Gnutella capabilities, including initiating and accepting connections to and from other peers, receiving request messages (*Query* in the Gnutella terminology), replying through an acknowledgement message (*QueryHit*) if the requested file is shared by the host, or forwarding request messages to neighboring peers. An idle peer can go to sleep as its presence in the overlay is guaranteed by the proxy. However, the latter has limited resources and, thus, it cannot store the files shared by the host. Hence, as soon as the proxy receives a request for downloading a file, it immediately wakes up the sleeping host and forwards the received request to it. In general, TCP connections with neighbors are maintained by the host when it is awake, and by the proxy when the host is sleeping. Specifically, when the control is transferred from the host to the proxy, TCP connections are terminated by the host and re-established by the proxy. Conversely, when control passes from the proxy to the host, the opposite occurs. To appropriately manage the interaction between the proxy and the host, some state information is essential, such as power state of the host (i.e., powered-on or sleeping), shared file names, and IP addresses of the neighboring peers. In particular, file names are shared between the host and the proxy in the form of a Bloom filter [Jimeno and Christensen 2007]. Experimental results show that the proposed proxy-based solution outperforms the legacy Gnutella protocol, in terms of energy consumption. Clearly, deploying this solution in the real world requires a specialized NIC, which is a major limitation of the proposal. To avoid this limitation, the authors of [Agarwal et al. 2009] have proposed a solution called *Somniloquy*, where the proxy is implemented through an external low-power embedded device equipped with an Ethernet NIC and connected to the user's PC through a USB port. Somniloquy is able to guarantee the continuous operation of various P2P applications (e.g., BitTorrent, instant messaging) even when the PC is sleeping. To run BitTorrent on the low-power embedded processor, the authors have customized a console-based client called *ctorrent*. Since the energy consumed by the embedded device is much lower than the energy consumed by the PC (in the order of 5%), the energy savings provided by Somniloquy may be very significant.

Proxy-based approaches are also used for P2P file sharing/distribution in wireless mobile systems. In this context, proxy entities are not used to turn peers into sleep mode, but mainly to prevent them from participating in swarms with low downloading rates. In fact, experimental results reported in [Nurminen 2010] show that low bit rates cause a high energy consumption in file transfers. Therefore, proxies are used to mediate between the mobile client and the other peers, so as to transfer data in high-speed bursts. Following this approach, the authors of [Kelényi and Nurminen 2010] have proposed *CloudTorrent*, a proxy-based solution where the proxy resides on a cloud server. In CloudTorrent the proxy behaves as a regular BitTorrent peer and acts as an intermediary for mobile peers during the download process. Essentially, files are downloaded by the proxy, and then transferred to the corresponding mobile client through a fast and dedicated connection. This approach results in a signifi-

cant reduction of energy consumption at the mobile device with respect to the legacy solution (i.e., without proxy). However, although running the proxy on a cloud is an attractive idea, it is unclear if any service provider would have enough incentive to do it.

To avoid the problem, in [Kelényi et al. 2010b], the same authors have proposed the *router proxy* solution, where the proxy is hosted on the home router used by the user for residential Internet access. The authors highlight that home routers represent a good choice to act as proxies, due to their pervasiveness and stable energy consumption (i.e., independent from the operation load). They also investigate the energy efficiency of the proposed solution. Simulation results show that the performance of the router proxy is strongly influenced by the limited memory space available at the router. Specifically, the amount of memory allocated to serve requests from other peers influences the file download speed, and, consequently, the energy consumption of peers. This is because, unlike regular BitTorrent peers, a router proxy deletes chunks already transferred to the mobile device, in order to reuse the limited memory space. Therefore, downloaded chunks are no longer available for other peers. However, the authors claim that the router proxy approach can reduce the energy consumption of mobile devices up to 80%, in comparison with the legacy approach.

In [Kelényi et al. 2012], the memory issue of router proxies is further investigated, and its impact on energy consumption is analyzed. The authors define an analytical model to analyze the memory allocation strategies of router proxies and predict how they influence the performance of the P2P system. Besides, they study how the size of chunks (sent by router proxies to the mobile peers) affects the energy consumption of peers. They conclude that using large data chunks – although requiring more memory – reduces the number of transmissions and the related overhead, thus improving energy efficiency.

To overcome the memory problem of router proxies, the same authors, in [Kelényi et al. 2011b], propose to divide the available memory on the router proxy into two buffers, namely *upload* and *download* buffer. Specifically, the download buffer holds transient data directed to mobile peers. This means that chunks are downloaded, sent to the mobile device, and discarded, so that memory can be reused to download other chunks. Therefore, the content of the download buffer is constantly changing as the download of the file progresses. Conversely, the upload buffer stores those chunks that are served to the swarm. In practice, after a chunk in the upload buffer has been downloaded and sent to the mobile device, it remains in the memory and is made available for other peers according to the BitTorrent protocol. The authors introduce an adaptive algorithm to determine the optimal upload/download buffer allocation and show that it can lead to a significant improvement in the file download speed and, thus, to a reduction of the energy consumption at the mobile peer.

In [Kelényi et al. 2010a], the router proxy idea is further extended. The authors propose a *distributed proxy* solution where multiple proxies are involved in the download process. The aim is to increase the limited uplink rate of the home router by using multiple router proxies, in order to meet the downlink capacity requirements of the served mobile devices. The effect on energy consumption of increasing the number of proxies in the distributed proxy approach is investigated in [Kelényi et al. 2011a]. It is shown that, after a certain threshold, adding more proxies to the system is not convenient since the aggregate proxy upload rate meets the download rate of the mobile devices. In addition, energy consumption for various upload/download ratios in single and multiple proxy settings are explored. The experiments show that

the energy consumption of multiple proxies slowly decreases when the upload/download ratio is low, and becomes stable once the aggregate upload rate meets the download requirements of mobile devices.

3.3 Peer Selection Approaches

Peer Selection is one of the most effective techniques to reduce energy consumption of P2P file distribution/sharing systems. The idea is to decide which peer (i.e., a *server peer*) must be selected by another peer (i.e., a *client peer*) to download from, towards optimizing energy consumption. As the decision maker needs information at the level of application, these solutions typically are achieved in the application layer.

In the standard BitTorrent algorithm, peers are selected according to a three-step strategy [Cohen 2003]. At the first step, a central server (i.e., tracker in BitTorrent terminology) randomly selects a list of peers to be sent to a new peer. At the second step, the new peer selects a random neighbor set from the received list. Finally, at the third step, the peer selects peers in its own neighborhood based on a choking algorithm guaranteeing a reasonable level of upload and download reciprocation. Although standard BitTorrent employs random peer selection along with a choking algorithm to improve both the robustness and quality of service, there is no strategy to select peers in an energy efficient manner.

In this respect, the authors of [Mawji and Hassanein 2010] propose a peer selection approach for a P2P file sharing overlay on top of mobile ad hoc networks (MANETs). The objective is to formulate and solve the problem of optimally selecting the server peers – and their related paths – to download from, considering energy and download time constraints. To this end, the authors employ a mixed integer linear optimization program that returns the optimal set of paths for clients to choose. However, the proposed algorithm requires prior knowledge of all client download requests, server locations, cost, and download time, thus limiting its practical value.

In [Enokido et al. 2010] the authors propose a peer selection approach for data transfer applications (like FTP) using a laxity-based algorithm. Starting from experimental results, they show that the power consumption of a server peer strictly depends on the transmission bandwidth, in the sense that the higher the transmission rate, the more power is consumed. Therefore, the authors present a power consumption model for a server peer that is applied to derive the laxity-based algorithm. The proposed algorithm allows to select a server peer among all the available peers, so that the deadline constraints are satisfied, and the power consumption of the server peer is minimized. Evaluations show that the laxity-based peer selection is more efficient in terms of total power consumption than the traditional round-robin selection method.

3.4 Resource Allocation Approaches

Hardware virtualization technique can be effective in P2P systems and applications towards reducing energy consumption. In fact, by employing virtualization technique in a P2P system, resources can be managed by virtual machines to perform tasks in an energy efficient manner. Some studies propose virtualization approach to conserve energy in P2P systems [Garcia et al. 2010; Hlavacs et al. 2008; Hlavacs et al. 2010]. They are based on the idea of energy efficient resource sharing among home networks interconnected via a P2P overlay, through virtualization of tasks. Virtualization of tasks means moving the virtual machine responsible for a particular task

from one Virtual Home Environment (VHE) to another VHE. Shifting load to a small number of computers provides that the relieved computers can be hibernated to save energy. A task virtualization prototype for unpopular files in a P2P file sharing system has been developed in [Hlavacs et al. 2012], where a virtual machine (VM) that remotely takes care of managing the resources for file download is used. By using the VM, download tasks are migrated to other machines. However, this approach has drawbacks in terms of privacy and security. The authors have developed scenarios to investigate the resource utilizations of CPU, memory and NIC, and the results show that resource utilizations grow linearly while the number of VMs involved in the system increases. Furthermore, the effects of load, upload bandwidth and fairness on the energy consumption were demonstrated. As the proposed solutions in this category need application level information to guarantee quality of service, they are achieved at the application layer.

3.5 Structure Optimization Approaches

During their life span, P2P networks maintain some sort of structure or topology, defining how peers maintain connections and communicate with other peers. In this section, we review approaches where this structure is constantly adapted and optimized with the new performance goal (i.e., energy efficiency) in mind. The existing approaches have been categorized into two sub-classes, namely *hierarchical* and *topology control* solutions.

3.5.1. Hierarchical Solutions. Hierarchical solutions build a new layer on top of the P2P file sharing overlay. In this layer, some peers carry out specific tasks in comparison to other peers in the overlay. The aim of this approach is to reduce the number of active peers in the system and, hence, the network overhead and resource usage. The special peers – usually called *super peers* – are either selected basing on an energy efficiency criterion across all peers [Han et al. 2008] or are fixed entities inside the overlay without any selection procedure [Trunfio 2014]. Super peer solutions can be achieved in both the network and application layer, depending on what kind of information the super peers need to maintain the system topology. For instance, if only routing is optimized, super peers act at the network level, whereas if content is indexed, super peers act at the application level.

In [Trunfio 2014], a hierarchical P2P file sharing model with two logical layers has been proposed. The *lower layer* is composed of a set of peers offering files to be shared, while the *upper layer* being made of super peers is responsible to form a distributed hash table (DHT), aiming to index the peers (including their availability status) and the files at the lower layer. The idea behind the proposed model is to save energy by allowing peers to switch between wake and sleep modes cyclically. To this end, the author proposes an energy saving algorithm, allowing peers to make decisions autonomously about their own availability status (sleep-wake cycle) based on a local criterion only (the number of files owned by them). Besides, the author proposes an extended DHT by introducing an additional data structure, which is handled by super peers to keep decentralized information about the availability status of all peers at lower layer. The reported results are promising since they show up to 50% decrease in energy consumption. In [Han et al. 2008], the authors propose an energy efficient routing approach for double-layered mobile P2P systems aimed to improve the system lifetime. In a double-layered P2P system there are two types of peers, namely super peers and sub-peers. The super peers are selected according to the

highest residual energy and the search process is carried out mainly through them. Each super peer in the proposed system maintains a routing table and a file table to keep the information regarding the routes and file lists, respectively.

In the literature, there exists another type of hierarchical solution, which is not based on super peers, and is referred as *active/passive peers*. In this category we can include *Smart Mobile Cloud* (SMC) [Chandrasekar et al. 2012], that is an energy efficient cloud-based P2P content sharing solution for mobile devices. SMC uses a strategy for grouping mobile nodes (i.e., *helper nodes* acting as active entities that manage the file download process, and *seeker nodes* requesting file downloads as passive entities) close to each other based on their location. Serving file download requests by using resources of the nodes in the same group would then facilitate energy efficiency.

Another energy adaptive technique for P2P file sharing in mobile devices is proposed in [Raj et al. 2013] where the idea is to gather mobile users according to their energy budget and to restrict their transmission capacity in order to achieve energy savings. More specifically, peers are grouped into *Energy Sufficient (ES)* and *Energy Constrained (EC)* peers. Peers in the ES group are either not battery constrained, or they are battery constrained but are able to download the file successfully within their given energy budget (based on the average upload and download rate of the peers inside the group). In the ES group there are no restrictions on bandwidth usage, and peers download the file only from neighboring peers from the same group. Peers in the EC group do not have enough energy budget to download the file successfully from the ES group. They are divided further into smaller sub-groups characterized by their respective energy budgets. In each such sub-group, peers have a similar energy budget and download the file using the same bandwidth. Energy sub-groups also have a unique energy consumption profile depending on the maximum upload and download rates at which peers in that group must adhere to while downloading the file. Therefore, peers usually join an energy sub-group with a lower energy consumption profile than its factual energy budget. In addition, peers in energy sub-groups are able to receive additional download rates from neighboring peers from the ES group by a credit exchanging strategy to prevent free riding. The authors of the paper evaluated and implemented the proposed mechanism in the context of the BitTorrent protocol with some changes in the tracker function.

3.5.2. Topology Control Solutions. A decentralized and dynamic topology control protocol, namely TCP2P, is proposed in [Leung and Kwok 2005]. The protocol is able to support energy efficiency, fairness and incentives in wireless P2P file sharing applications by the dynamic construction of a specific adjacency set for each node that can be used in any typical ad hoc routing algorithm. In this work, the authors define a preference function enabling each node to quantify other nodes' energy level as well as its intention to be a relaying node. This function is a combination of four metrics, including contribution, aggressiveness, file popularity, and energy. The contribution metric evaluates the level of contribution of each node to upload files. In fact it shows how selfish a node is in terms of downloading files without any uploading. The aggressiveness metric evaluates the level of contribution of each node to maintain the network infrastructure. The file popularity metric evaluates how popular the files owned by each node in the network are, and the energy metric evaluates the remaining energy level of each node. The preference function, along with a threshold value, is employed to construct the adjacency set inside each node dynamically.

In [Leung and Kwok 2008] the authors have extended their idea by adding a new component to their topology control protocol. In general, the protocol presented in the

new work is composed of two components. The first one is a selective neighbors set construction mechanism (Adjacency Set Construction – ASC), exactly the same already mentioned, and the second new one is a community-based scheme for asynchronous wakeup (Community-based Asynchronous Wakeup – CAW), being responsible for grouping users into communities, each of which applies a different sleep-and-wake schedule. The rationale behind this idea, for example in music file-sharing networks, is that data communication between users with similar interests is likely to be higher than between other users with different interests. Therefore, using community formation along with a different sleep-and-wake scheduling not only increases the chance of successfully downloading a file, but also allows a group of nodes to turn into sleep mode and conserve energy when other groups are active. These solutions are achieved at the application layer, as they need information at the level of application e.g., file popularity, contribution of the node in uploading and downloading.

3.6 Message Reduction Approaches

Reducing the number of messages being responsible for P2P file sharing system operation and maintenance (i.e., ping/query message in Gnutella) can result in reducing the processing overhead inside peer nodes and, consequently, conserve energy. In [Kelényi and Nurminen 2008a] the authors investigate the behavior of a mobile DHT client inside the Kademlia-based mainline BitTorrent DHT. They show that the energy consumption used by a mobile client is a function of the number of incoming messages processed. In this work, two operational modes are introduced, namely, *full-peer* mode and *client-only* mode. In the former, the node participates to the DHT by replying the queries sent by other nodes. In the latter, the node does not answer queries and drops all incoming query messages in the application layer. In fact, the client-only mode allows a node to use the service of the DHT, however, it does not have to participate in the operations initiated by other clients. Further, the authors show that the client-only mode for a mobile client could achieve a significant energy conservation. A protocol extension of the Kademlia-based DHTs is presented in [Kelényi and Nurminen 2008b]. It allows mobile clients to drop messages selectively according to a probability, which can dynamically depend on the battery status of the node. They show that dropping incoming requests hardly influences the overall performance of the DHT if the number of nodes using this method is relatively low. Message dropping solutions can be achieved in different layers (e.g., network or application layer), depending on how much information they need in order to act intelligently. For instance, random message dropping has the lowest level of intelligence, and thus it can be achieved in the network layer.

Reducing the number of messages can be considered in designing P2P routing protocols. The authors of [Ding and Bhargava 2004] claim that broadcast-based routing protocols are inefficient in terms of energy consumption in P2P file sharing systems over MANETs. Therefore, they propose five different routing algorithms by integrating broadcast-based and DHT-based algorithms, and compare these approaches in terms of energy efficiency, complexity and scalability. Gossiping protocols, as routing algorithms, are able to reduce the number of messages through their inherent probabilistic forwarding behaviors. Based on this, the authors of [da Hora et al. 2007] propose an adaptive routing protocol based on gossiping on top of MANETs which is able to adjust its forwarding probability according to the network load. They indicated energy reductions by at most 32% using simulation analysis of Gnutella.

3.7 Open issues & Research guidelines

In this section, we identify some open issues existing in the approaches presented in the previous sections, and try to propose possible solutions for them.

Some of the most significant challenges in proxy-based approaches [Anastasi et al. 2010; Hlavacs et al. 2012] are privacy and security. In general, the proxy, as an external and dedicated entity, can become a weak point of the system. If the proxy is hacked (for example, when it is polluted or infected by viruses, worms or Trojan horses), the peers connected to it probably would be infected. On the other hand, for privacy reasons peers might not be willing that others be informed about what are sought by them. When considering privacy, privately owned proxies are to be preferred to publicly shared ones.

Privacy and security issues can be also considered in task virtualization solutions [Hlavacs et al. 2012], where downloaded tasks are migrated to untrusted peers as virtual machines. Problems arise since such VMs can contain private data, which might compromise the VM owner. Thus, the VM should be protected from outside analysis, which is very difficult to be guaranteed once the VM in its entirety is stored at foreign storages. Relying on hardware enforced encryption and decryption as, for instance, provided by *Trusted Platform Modules* (TPMs) might be a solution, but is hindered by the low acceptance and availability of such devices.

Within the sleep/wakeup class, the proposed scheduling-based solutions [Sucevic et al. 2011; Andrew et al. 2011] need to have a global knowledge from all peers to achieve balance between energy consumption and performance degradation in terms of downloading speed inside the system. This often leads to centralized management and, consequently, negates the distributed nature of P2P systems. A solution might be to design a fully decentralized scheduling mechanism in which peers would be able to make decisions about their own mode (sleep/wakeup) based on their local state and information received from neighboring peers.

Peer selection approaches [Mawji and Hassanein 2010; Enokido et al. 2010] are based on the assumption that peers have global knowledge, such as client download requests, server locations, cost and download time, etc., effectively limiting their practical value. A solution for these approaches can be to apply a coordination strategy between peers whereby each peer would be able to share its own information with others. In this case, further issues including privacy and communication overhead must be considered.

As we stated in the previous subsection, reducing the number of messages received by the peers inside a P2P file sharing system, although resulting in decreasing the processing overhead of the peers and, consequently, in decreasing the total energy consumption, can influence the overall performance of the system [Kelényi and Nurminen 2008a; Kelényi and Nurminen 2008b]. The challenge posed here is to find a trade-off between energy consumption and system performance. A solution can be obtained by using a cooperative setting, in which each peer is able to decide locally whether to drop incoming messages (i.e., not to forward them), and how often to do that. This effectively reduces the number of messages in the system and, therefore, the global energy consumption. The drop-probability might then be determined by some global parameter guiding the desired overall system performance, computed with the help of other peers. Since determining the desired drop-probabilities requires communication and, thus, energy itself, care must be taken to prevent a situation where computing drop-probabilities costs more energy than which is effectively saved by message dropping itself.

Combination of two or more approaches to find a hybrid solution can be considered as an open issue. In [Leung and Kwok 2005] and [Trunfio 2014] the authors combined a structure optimization approach with a sleep/wakeup approach, and proposed a hybrid solution to achieve more energy efficiency in a P2P file sharing application. A question raised here is how to combine the approaches without any conflict. Consider a solution, which is a combination of sleep/wakeup and peer selection. In this situation, a peer might go to sleep by a command from a sleep/wakeup scheduler while it was selected previously by a peer selection algorithm to download from. Therefore, an orchestration strategy might be necessary to avoid such conflict in a hybrid solution.

4. CONTENT STREAMING

Multimedia content streaming usually presents several QoS requirements and differs from traditional file sharing for a number of key distinguishing features. For instance, the client typically begins video playout within a few seconds after it has started receiving the video packets (chunks) from the server. This means that the video playback starts while later parts of the video are being received. Thus, during the playback, it is desirable to minimize the probability of “freeze-ups” caused by buffer underflow. This clearly affects the download process. In file sharing protocols (i.e. BitTorrent), files are downloaded according to the available chunks. Therefore, after the user has started the download, the actual downloading of chunks may be delayed, interrupted or out of order. Conversely, in audio and video streaming, downloads are time-constrained and should be sequential. With reference to live streaming, download time and speed are upper bounded by the source. In this case, a download speed close to the bit rate of the video source is preferable. Anyhow, the network must provide each multimedia flow with an average throughput that is larger than the video consumption rate.

In the following, we consider two main categories of multimedia applications [Kurose and Ross 2013a]: *stored content streaming* and *live content streaming*. In live video/audio streaming (aka live streaming), a source distributes content gradually as it is created, such as a live telecast, similarly to broadcast television. In stored video/audio streaming (aka Video on Demand or VOD), the user requests to watch a content (e.g. a movie) that has been pre-recorded and made entirely available by a source. Many Internet companies provide video streaming, such as Google (YouTube) and Netflix. While watching a stored video, the user may pause and reposition forward/backward through the video content. The system must be responsive and execute the user’s actions with minimum delay.

In the context of live content streaming, energy efficiency techniques are generally designed for mobile devices. In fact, appropriate download strategies can significantly limit the use of network interfaces, thus reducing the energy consumption and extending the battery lifetime. Ideally, applications should be able to satisfy both energy and resource constraints imposed by devices, while, at the same time, fulfilling the QoS requirements needed for an enjoyable video playback. In this respect, given the increasing relevance of mobile video streaming, a number of tools have been developed to evaluate the efficiency of streaming protocols and applications. For instance, the tools described in [Mohiuddin et al. 2013] and [Wichtlhuber et al. 2012] (an OM-Net++ based simulator and an Android-based tool, respectively) allow to quickly develop and compare different wireless P2P video streaming techniques and monitor the energy consumed by peers.

In the wired context, instead, optimizing network communications does not lead to a significant energy benefit. Essentially, in order to obtain more significant savings, the entire device should go into standby or sleep mode. Obviously, this is not feasible in case of live streaming, since the client device must be on for the whole duration of the video. On the other hand, an approach based on scheduled shutdowns/wakeups can be applied in the case of stored content, as we will see for set-top boxes below.

In the following sections, for each considered approach, we will survey the main solutions proposed in the literature.

4.1 Sleep/Wakeup

The sleep/wakeup approach has been mainly applied to the wired context, particularly for IPTV applications. *IPTV (Internet Protocol Television)* is a system that allows to deliver audiovisual contents on a packet-switched network based on *TCP/IP* protocols. Usually, contents are offered in the form of stored video streaming.

Typically, the network architecture for content streaming is based on *data centers*. In practice, Internet Service Providers (ISPs) or Internet companies manage large server farms that provide for the distribution of multimedia contents. Therefore, users download contents through the client-server paradigm. Downloaded files can be placed into a cache memory, so that the user will be able to view them in the future. There are many devices that can be used to access the available contents: consoles, digital receivers, video recorders, home gateways (generically referred as *set-top boxes*, *STBs*). An evolution of this fully centralized approach is represented by *Content Delivery Networks* or *Content Distribution Networks (CDNs)*. A CDN [Peng 2004] is a large-scale distributed system made up of servers deployed in multiple data centers at strategically chosen locations throughout the Internet. In a CDN, contents are replicated from the source server to replica servers scattered over the Internet and requests are served by the replica server closest to the requesting user. CDNs (e.g. *Akamai*, *CloudFront*, *Coral*, etc.) are typically used to deliver large media contents, such as IPTV, and offer audio and video streaming services. Indeed, CDNs result in several benefits both for users and content providers. For instance, since requests are handled by servers close to the destination, download times are typically shorter. In addition, splitting the load among a number of distributed data centers allows providers to optimize bandwidth usage and reduce management and electricity costs, as shown in [Qureshi et al. 2009; Feldmann et al. 2010; Gao et al. 2012].

Fig. 3. Collaborative scheme for IPTV STBs.

A different emerging approach for content streaming consists in creating a P2P network among the users' set-top boxes [Huang et al. 2008; Laoutaris et al. 2008]. This way, STBes can share the previously downloaded contents in their cache, relieving the load of servers (as shown in Figure 3). Therefore, users' STBes form *Nano Data Centers (NaDa)*, as opposed to traditional data centers.

In the literature, there are many studies and comparisons about the architectures used for content streaming and, in particular, about Nano Data Centers (a synthesis can be found in [Ramos 2013]). In [Valancius et al. 2009], the authors propose to use ISP-controlled home gateways as Nano Data Centers, providing them with proper hardware in order to perform computing and storage of contents. In this way, STBes form a P2P network managed by the ISP. In the same work, the authors present the results of simulations carried out from real content access traces obtained from three popular sources (*Netflix*, *Youtube*, and a large ISP). The results show that the NaDa approach achieves significant energy savings compared to the solution based on traditional data centers. Specifically, energy savings are mainly due to two factors. First, it is possible to save costs for server *cooling*, since the distributed architecture does not need it. In addition, the considered approach makes efficient use of devices that are always on (i.e., home gateways), allowing to exploit the already committed *baseline power* of underutilized/idle gateways.

However, as highlighted by the analysis in [Liu and Wang 2013], the energy efficiency of the NaDa approach mainly depends on the energy consumption of STBes, especially when considering devices different from home gateways. Obviously, the consumption of STBes is determined by a number of factors. Firstly, the kind of set-top box must be considered (e.g., a gaming console consumes much more than a home gateway). In addition, the amount of time that STBes are active must be taken into consideration. For instance, if energy-consuming STBes are used and are left on even when no uploads/downloads are in progress, the P2P approach is not convenient (as shown in [Baliga et al. 2009] and [Feldmann et al. 2010]). Conversely, significant savings – with respect to the traditional data-center approach – can be achieved by minimizing the uptime and the energy consumed by STBes. Besides, keeping end devices active causes a shift of consumption (and, thus, costs) from data centers to users, resulting in a major drawback. In fact, users may not be interested in spending more in order to achieve an overall saving or a saving for the ISP. For all these reasons, some solutions have been proposed to limit the STBes' uptime, thus allowing to benefit from the P2P architecture (i.e., decreasing load and power consumption in the data center) while minimizing the energy costs for users. These solutions are typically implemented at the application layer, as the streaming application determines when the STB must be switched off and on, basing on the likelihood (known a priori or estimated by the device) that a cached content is requested for download. In [Jourjon et al. 2010] the authors perform a detailed analytical study about the energy consumption of an STB network. The presented model shows that the fixed part of energy consumption (simply due to the fact that the device is on) is a major part of the total consumption, even though the STB is involved in downloads or uploads. For this reason, turning off the STB when it is not being used can lead to significant energy savings. In the paper, a simple algorithm is proposed to determine, how many, and which STBes should be switched off so as to leave the minimum number of devices needed to keep the IPTV service up and running. The proposed solution can drastically cut the total consumption of the P2P network. However, the algorithm is

purely theoretical and requires to know some data that are not available a priori (e.g. how many times each video is requested for download during the day).

To overcome the previous limitations, the authors of [Pussep et al. 2010] have designed and implemented several adaptive algorithms. In particular, the authors analyze the following Sleep/Wakeup policies.

- Selfish*. The STB is powered on only for the time the user utilizes it to watch videos. Obviously, this solution limits the effectiveness of the NaDa paradigm by producing a considerable amount of traffic to the servers.
- Wakeup*. The STB is switched off automatically at the end of a download. If a file that has been cached by the STB is requested, and the other online peers are not able to provide the necessary bandwidth, the STB is switched on through Wake-On-LAN. In theory, this constitutes an optimal algorithm. However, since Wake-on-LAN presents many shortcomings (e.g. security issues and starting delay), this policy has little practical applicability.
- Popularity- and Supply-aware Policy (PSP)*. For each video in the cache, the STB calculates (and periodically updates) a level of “*desirability*”, based on the number of requests received in the past for this video, and the bandwidth that other peers in the network can provide for its upload. If the desirability of the cached videos is low, the STB is turned off, otherwise it remains active. Information about files owned by each peer and their bandwidth can come from a dedicated server (called *indexing server*) or can be exchanged by peers via *gossiping*.

The simulation results show that the PSP algorithm exhibits the best behavior, with a hit rate (i.e., probability that a video is found in the P2P network without accessing the server) of 89% and an uptime equal to 8% of the total, thus allowing an energy saving in the P2P network of 86% compared to the always-on policy.

4.2 Peer selection

Peer selection approaches consist in picking proper peers of the P2P network to request chunks, so as to save energy. In other words, whenever a peer needs a particular content, the streaming application should properly choose the node to download from, in order to minimize the energy consumption. To this end, many criteria have been proposed for choosing the supplier peer. Some examples are presented below.

Maamar et al. in [2011] and [2013] consider a protocol to select the supplier peer in mobile P2P 3D streaming applications. In augmented reality, the real world is mapped to the virtual world and mobile devices are used to navigate the virtual environment. Peers that are near in the real world are considered near in the virtual environment too. Therefore, in P2P applications, a peer can ask for 3D data to every other node which is in its same *Area Of Interest* (AOI). In order to manage the peers’ resources and limit the energy consumption, the authors propose an algorithm that aims to relieve overloaded peers and extend their battery lifetime, providing an equal distribution of the energy consumption among peers. Thus, the authors present a load estimator that takes into account the peer’s residual energy and the number of currently served requests. Basically, each peer periodically broadcasts a beacon message containing information about its position, speed, buffer indices and estimated load. Therefore, a peer that needs some 3D data uses the collected information to select the least loaded source among the peers, in the AOI, that have the required data. In an improved version of the algorithm, each supplier peer estimates its own load and compares it with its neighbors’ loads. In case it is overloaded, in comparison with

its neighbors, it informs the other peers by sending a special message. Both versions of the algorithm have been tested, and simulation results show that they perform better than other mobile P2P 3D streaming protocols, in terms of energy consumption and response time.

In order to select the best peer, the application can also exploit information obtained from the lower layers, following a cross-layer approach. This is the case of the solution presented in [Li et al. 2009]. In this work, the authors propose an algorithm that aims to schedule the video transmission among the peers of a P2P network to avoid freeze-ups in playback due to buffer underflow. They extend the algorithm presented in [Li et al. 2007] and [Li et al. 2008] to wireless networks, taking the energy issue into account. The authors consider a P2P network where one peer is a video source and the other peers have to get the shared video. The algorithm calculates the level of “happiness” of each node, measured by a utility function depending on the content reserve level of the node (i.e., the number of video segments remained in the buffer to be played). Then, for each scheduling interval, a coordinator selects the peer i with the least content reserve level (the least happiness) to be served first. The peer i must choose a peer j to request the needed video segments. In order to optimize the energy consumption, peer i selects the peer with the best channel gain (and, hence, the highest effective transmission rate) among all the peers that have those segments available. Simulation results show that, in both cases, the proposed solution balances the tradeoff among battery life, bandwidth limit and playback quality, achieving better overall system efficiency.

4.3 Network-interface-based resource allocation

The resource allocation approach consists in selecting the appropriate resources to perform a video streaming with minimum energy consumption. It is particularly suitable for mobile devices provided with several wireless network interfaces. In fact, nowadays, modern smartphones, tablets and notebooks are equipped with long-range interfaces (LRI) to connect to cellular networks (e.g. LTE, 3G, UMTS) and short-range interfaces (SRI) to connect to near access points, other PCs or smartphones (e.g. via WiFi or Bluetooth). Since different network interfaces are characterized by different data rates and energy consumption, it is possible to select the most appropriate solution that minimizes the energy consumption. Essentially, long-range interfaces are much more energy consuming than short-range ones, as highlighted in [Huang et al. 2012]. Hence, using SRIs rather than LRIs would lead to considerable energy savings. In addition, several studies (e.g., [Chen et al. 2013]) point out that people in metropolitan areas tend to concentrate in the same areas (e.g., hotspots) where they can enjoy multimedia applications. In such situations – with a high density of mobile devices – it is likely that several users are interested in the same contents. Starting from these considerations, P2P protocols can be designed so that peers can share the data they have already downloaded through the SRI, instead of downloading them from servers via the long-range interface (see Figure 4). Once again, the application decides when to use a LRI or a SRI to download a video chunk, basing on information directly obtainable from the application layer (e.g., residual energy or necessary video segments) and/or data coming from lower levels (e.g., channel gains measured at the physical layer). A proper choice of the wireless medium (and, therefore, of the access protocol), not only can reduce the energy consumption, but can also result in a better utilization of the bandwidth and, thus, in shorter download times.

Fig. 4. NIC-based cooperative P2P streaming architecture.

In this perspective, Chen et al. in [2011] and [2013] propose a *Distributed Bitrate-aware Entrust* algorithm for live video streaming in wireless mobile networks. Its goal is to maximize the system lifetime, while minimizing the total consumed energy and achieving an equal distribution of energy consumption among user devices. Basically, the algorithm aims to make near peers exchange downloaded packets through the SRI in order to reduce the requests to servers and, hence, the usage of the cellular network. Specifically, the algorithm works as follows. Periodically, each peer communicates its residual energy to its neighbors. Therefore, a peer that needs certain data sends an entrust packet to the neighbor with the highest residual energy. The selected peer has to download the data packet by itself, via its LRI, and broadcast it to the requesting peer via the SRI. This way, the algorithm limits the number of downloads on the cellular link and the number of unnecessary transmissions on the SRIs. Simulation and experimental results show that the proposed algorithm can achieve significant energy savings compared to the traditional client-server approach where each node downloads data from servers, improving the battery's lifetime by 27%. In addition, this solution also allows to stream videos with a variable bit rate, or a bit rate higher than the LRI. In the latter case, the algorithm selects more than one peer to download (and then share) data from servers at the same time.

Other studies address the energy problem in mobile video streaming when different coding schemes, such as *Multiple Description Coding (MDC)* and *Scalable Video Coding (SVC)*, are used. Briefly, in MDC, the data stream is split into a set of stripes that can be delivered and decoded independently as separate streams of media packets. The video quality at the receiver depends on the number of correctly received stripes. In SVC, the video signal is split into several layers as well, but in a hierarchical way, with a main *base layer (BL)* and several *enhancement layers (EL)*. Basically, the BL provides coarse video quality and corresponds to the lowest supported video performance, whereas ELs are used to progressively refine the quality achieved by the base layer. Exploiting the characteristics of these coding schemes, it is possible to design P2P energy efficient protocols, such as the ones in [Albiero et al. 2008]. In short, the authors propose a system where each node downloads a specific stripe/layer through the LRI and, then, exchanges data with other peers via SRI, till it can reconstruct the whole stream. They also propose a *Dynamic Cellular Channel Allocation* for SVC coding. In this case, for each frame period, all the layers are downloaded by the node with the best channel gain towards the base station of the cellular network. Then, the video is shared among the other peers in a BitTorrent fashion. Interestingly, the authors perform an analytic evaluation of the energy consumption with the proposed algorithms for different kind of SRIs and LRIs (i.e. WiFi, GPRS, UWB, Bluetooth), showing that significant savings can be achieved compared to the

non-cooperative approach. In addition, experimental results obtained from a real testbed for the MDC algorithm show that a configuration GPRS/Bluetooth can lead to roughly 50% energy savings even with only two cooperating terminals.

A key aspect to take into consideration when dealing with these approaches is the willingness of nodes to cooperate, i.e., to join the P2P media streaming system and consume energy for other nodes. Indeed, solutions that aim to minimize the overall energy consumption of the system could increase the energy expenditure of some specific nodes, compared to the client-server approach. For instance, if a node has to download a whole video from a server and then forward it to the other peers, it will consume more energy than downloading the video just for itself. Obviously, a node will agree to participate in a P2P collaborative system only if it can achieve some benefits (e.g., it can save energy and extend its battery life). Starting from these considerations, [Yeung and Kwok 2008] proposes some protocols for energy efficient media streaming in mobile networks that take into account the amount of energy that each node is available to consume (namely the *available energy*). Particularly interesting is the algorithm proposed for the MDC scheme. Basically, inside the P2P network, a peer is elected as coordinator and asks every other peer in the network for its available energy. Upon receiving responses from the nodes, the coordinator calculates, for each node, the number of stripes that can be downloaded directly from servers (through LRI) and the number of stripes that can be exchanged with other peers (through SRI), according to the obtained values. Therefore, each peer directly downloads only some stripes and receives the remaining ones from the other nodes, without violating its energy consumption constraint. Simulation experiments show that the algorithm allows nodes to obtain more stripes than with the traditional client-server approach and, thus, enjoy a better video quality with the same energy consumption. Since each node achieves a benefit from the system, none of them has interest in exiting the system. Therefore, a Nash Equilibrium has been achieved.

In the literature, there are a lot of other solutions addressing video streaming for mobile devices and proposing energy efficient cooperation systems based on a smart usage of LRIs and SRIs (e.g. [Ramadan et al. 2008; Barua et al. 2013]). Some solutions apply the same concept to specific contexts. For instance, the authors of [Zhang et al. 2007] focus on IP-services over DVB-H (Digital Video Broadcasting on Handheld) networks.

4.4 Open issues & Research guidelines

The solutions surveyed in Section 4 present some limitations. In this section, we propose some possible enhancements to overcome the aforementioned drawbacks. In addition, we present a new and promising approach for P2P multimedia streaming that deserves to be further investigated to achieve energy efficiency.

For example, the NaDa approach – while ensuring overall energy savings – moves part of the energy consumption to the users. Obviously, users may decide to accept a slightly higher energy expenditure, only in return for some benefits in the streaming service, e.g., a greater number of available contents and/or faster download times. Furthermore, in the case of IPTV, the described solutions are not suited to all STBs, e.g., to those which are not equipped with the proper hardware (for instance, a hard disk for the storage of contents). Similarly, Sleep/Wakeup solutions cannot be applied to home gateways, since they must be always active. More generally, the energy consumption, and therefore the effectiveness of the proposed solutions, strictly depends

on the STBes used. Hence, in order to evaluate all these factors, it would be appropriate to carry out an experimentation with real hardware, conducted by real users.

Peer-selection solutions, instead, require non-trivial calculations and evaluations to determine the best supplier peers (e.g., estimates of load, or channel gain, for each peer of the network). Moreover, the required exchange of information often leads to additional, non-negligible overhead and traffic. Experimental evaluations in a real environment are necessary to determine the actual efficiency of these solutions.

Finally, the network-interface-based resource allocation approach – in order to be effective – assumes that many nearby users request the same multimedia contents. However, it is usually unlikely that many people require the same content inside the coverage area of a smartphone's SRI. In a real scenario, finding devices with the desired contents can be difficult and even energy consuming, since SRIs would not be used for streaming but mainly for searching the content among the neighbors. In the literature, some solutions address this problem. For instance, in [Miyake and Bandai 2013] the authors propose a context-aware approach through which a mobile device can decide when to activate the WiFi interface for communicating with other peers. Basically, the user's mobile device periodically searches for neighboring devices, using a low-power SRI (e.g., Bluetooth). When another device is found, the WiFi interface is enabled and the two peers exchange information. If the desired content has been found, the P2P streaming takes place. Otherwise, the WiFi interface is powered off. Anyway, although this solution allows to efficiently identify peers that own the required content, new ways to extend the P2P network should be studied, even among peers that do not need the same contents.

It is worth noting that further energy savings can be achieved by properly combining the above approaches. For instance, it could be interesting to consider the use of peer selection strategies in a P2P swarm whose nodes are equipped with both LRIs and SRIs. In this case, in fact, not only the energy consumption can be reduced by limiting the use of LRIs, but it can be further optimized by selecting the peers in the P2P network that can supply the requested chunks with the minimum energy consumption.

In addition to the surveyed approaches, there are other technologies for content streaming in P2P networks. Many of them have been initially designed to maximize the throughput of the network, limit response/download times and avoid freeze-ups. However, these solutions often neglect the energy problem. Therefore, given the popularity of streaming applications – especially in mobile environments – they should be revisited to make them more energy efficient. This is the case of several P2P architectures following a structure optimization approach [Magnetto et al. 2010; Tran et al. 2003; Guo et al. 2008]. Basically, in these solutions, the contents provided by a single media source are distributed to many clients by organizing them into a tree rooted at the server. Peers are grouped into clusters representing sets of collaborating peers, and a hierarchy is formed among peer clusters. In this way, it is possible to realize a scalable P2P streaming architecture offering excellent user Quality of Experience (QoE) and able to accommodate large user populations and support high streaming rates, while still providing low delay and robustness to peer churning. Indeed, this approach actually improves the exploitation of the available upload bandwidth of all participating users, and offers a better user experience compared to the traditional client-server paradigm. However, in the mentioned solutions energy consumption is not considered as a performance criterion. Therefore, it should be further investigated the possibility of applying the structure optimization approach to

achieve energy savings, also in combination with other strategies. For instance, in this perspective, a network-interface-based resource allocation strategy could be used to reduce the energy consumption of mobile nodes, simply by creating clusters of near peers and by making them exchange data through their SRIs.

Another significant streaming technology that is inefficient from the point of view of energy consumption is represented by Network Coding. It is an innovative technique for content distribution, that aims to maximize the throughput in multicast communications. It is particularly suitable and appreciated for P2P applications, since nodes can be arranged in random overlays without requiring coordination. Basically, nodes do not simply relay the packets they receive, but they take several packets and combine them together for transmission. Essentially, each node generates packets that are linear combinations of the previously received ones. Therefore, a peer can obtain the original content, after receiving a limited number of linearly independent packets. Network Coding is used in multimedia streaming [Magli and Frossard 2009], as it allows to achieve a good quality video, but also to maintain low buffering times [Wang and Li 2007b; Wang and Li 2007a] and minimize the number of control messages and acknowledgments exchanged. However, coding/decoding operations are computationally expensive and, thus, energy consuming [Heide et al. 2008]. Currently, several studies analyze the use of Network Coding in P2P content streaming in the mobile context. Many energetic evaluations have been made, such as in [Shojania and Li 2009; Vingelmann et al. 2010]. In addition, some solutions have been proposed to reduce the energy consumption of decoding operations. For instance, [Angelopoulos et al. 2011] shows that Network Coding can be efficiently implemented using ad-hoc hardware. [Heide et al. 2009] and [Fiandrotti et al. 2014] offer an optimization of Network Coding technique that reduces the computational complexity of video packages decoding. At any rate, this technology, although it comes with many benefits, cannot be considered energy efficient yet, when compared with other P2P or client-server schemes. Thus, more work in this direction should be desirable.

5. P2P EPIDEMICS

Designing energy efficient P2P epidemic (gossip-based) protocols and services has become significant due to their wide usage in large scale distributed systems. Several distributed services such as reliable multicasting [Birman et al. 1999; Ozkasap et al. 2006; Ozkasap et al. 2009c; Eugster et al. 2003], aggregate computation [Jelasity et al. 2005], frequent items discovery [Cem and Özkasap 2013], overlay topology construction, failure detection, buffer management [Ozkasap et al. 2009b], P2P streaming and data dissemination [Ozkasap et al. 2009a] use epidemic approaches. Key advantages of epidemic protocols are their simplicity, scalability and high fault-tolerance properties. Moreover, all peers have equal responsibilities, and hence the system is inherently load balanced. Energy awareness of epidemic protocols has attracted attention and has been addressed in both wired and wireless network settings at different layers of the protocol stack [van Renesse 2002; Rajeswari and Venkataramani 2010; Fauji and Kalpakis 2011; Ozkasap et al. 2014; Lefebvre and Feeley 2000; Garbinato et al. 2010; Hou et al. 2006; Lu et al. 2008; Khouzani et al. 2012; Kontos et al. 2014]. These energy efficient mechanisms for P2P epidemics can be grouped into approaches of sleep/wakeup, resource allocation, structure optimization and message reduction. Based on the epidemics terminology, a peer holding data or an update it is ready to share is called *infectious*. If a peer has not yet received data,

it is called *susceptible*. In an epidemic protocol, each non-faulty peer can either be susceptible or infectious at a given time [Ozkasap et al. 2010]. An epidemic algorithm consists of periodical rounds for data dissemination. In each round, every peer randomly picks and contacts a fan-out number of peers to exchange states. The algorithm finishes in multiple rounds, and data is disseminated to the network resembling the dissemination of an epidemic disease in biological networks [Birman et al. 1999]. In epidemic protocols there are three communication styles used to exchange states, namely *push-based*, *pull-based*, and *push-pull*. In the push-based model, each peer chooses random peer(s) to *send* its local state to. In the pull-based model, each peer chooses random peer(s) to *receive* the respective local states from, and in the push-pull model, each peer chooses random peers both to *send and receive* states.

An energy cost model for a peer using gossip-based communication that considers the effects of gossip protocol parameters (e.g. fan-out, maximum gossip message size, number of rounds for convergence) is proposed in [Cem et al. 2012]. As a case study, it considers an epidemic-based distributed protocol for frequent item set discovery in P2P networks (*ProFID*) that uses a novel atomic pairwise averaging operation [Cem and Özkasap 2013]. The developed cost model is then extended to a hierarchical epidemics protocol for energy efficiency [Ozkasap et al. 2014]. ProFID is an application layer protocol that executes in rounds and peers periodically exchange their states with their neighbor(s).

There are two types of messages used in gossiping: push gossip message sent by the initiator of the gossip, and pull gossip message sent by the gossip target to the gossip initiator as a reply. The protocol involves three main components of operations performed by each peer, namely energy consumed while computing new state, sending messages and receiving messages. In the cost model, energy consumption during an atomic pairwise averaging operation between two peers P_i and P_j is depicted in Figure 5. E_{send} and E_{receive} denote the energy required to send and receive the item tuple respectively. $E_{\text{compStarter}}$ represents the energy required to choose tuple to send and update the state, and $E_{\text{compTarget}}$ is the energy required to compute the average and prepare the tuple to send.

Fig. 5. Illustration of an atomic pairwise averaging operation

There exist two main classes of *epidemic algorithms*, namely *flat* and *hierarchical*. Flat algorithms include *basic* and *neighborhood* epidemics. The basic epidemic requires global knowledge of peer population and performs uniform gossiping, therefore it is not practical. On the other hand, neighborhood epidemic uses local knowledge which is more practical and performs gossiping with neighbors. Although neighborhood epidemic is better when compared to basic, it still has the problem of redundant communication. However, hierarchical epidemic makes use of structure among peers and aims to reduce communication overhead. In addition, it provides the possibility of active/passive peers to save energy.

As provided in Table I, energy efficiency studies in the context of P2P epidemic algorithms fall into the categories of sleep/wakeup, resource allocation, structure optimization and message reduction. The state of the art in these categories, as well as the open issues and research guidelines, are described in the following subsections.

5.1 Sleep/Wakeup

Large scale P2P systems consume significant amounts of energy as they require all peers should be active and their resources are assumed to be free. However, this assumption is not true. The study in [Lefebvre and Feeley 2000] is based on the fact that idle peers can switch to sleep mode to save energy and focuses on energy efficiency in P2P storage systems. For this purpose, an upper bound for the duration of time a peer can reside in the sleep mode is derived by considering the durability of data stored in the P2P storage system.

When dynamic peer departures and arrivals are considered, a number of challenges are proposed for the design of an energy efficient P2P storage system. A key challenge is identified as durability which is the ability of the system to preserve data continuously. To ensure durability, transient and permanent departures of peers are aimed to be differentiated. This would help in preventing cost of data copy when replicating data on idle nodes. The idea in this study is to determine a bound for the duration an idle peer can be disconnected from the system. If a peer is disconnected more than the specified bound, then it is assumed to be permanently departed. The derived upper bound is based on the replication factor and permanent departure rate estimations. By using the derived upper bound, the aim is to differentiate peers that are in sleep mode from the ones that are failed.

Achieving inter-peer communication, when a small number of peers are available, is considered as another key challenge in the design of an energy efficient P2P storage system [Lefebvre and Feeley 2000]. The peers should monitor each other to determine peer failures using periodic message exchanges. The main question is how does a peer can find out if another peer is alive when the probability of both sender and receiver peers are awake simultaneously is small. As a solution, a gossip based failure detection protocol that can work with active nodes in the system is proposed. The failure detection protocol functions at the application layer.

A gossip-based energy conservation method based on sleep/wakeup approach for wireless ad hoc and sensor networks is introduced in [Hou et al. 2006]. In the proposed *Gossip-based Sleep Protocol* (GSP), each node randomly switches to sleep mode for a time duration using a gossip probability which depends on the network density. The idea is to save energy by turning some nodes into sleep state and still keeping the network connectivity, since in a well-connected network there exist multiple paths between a source node and a destination node. Two types of network layer GSP, namely synchronous and asynchronous, are proposed suitable for synchronous and asynchronous networks, respectively.

By using probabilistic sleep modes, the aim is to reduce energy consumption of the network and improve network lifetime [Hou et al. 2006]. A drawback of sleep mode could be increasing the failure rate and length of path between source and destination. Through simulations it is demonstrated that for specific gossip sleep probabilities and topology densities, network connectivity is preserved. Network lifetime, throughput and end-to-end delay metrics are investigated to observe the efficiency of GSP. Both static sensor networks (synchronous case) and mobile ad hoc networks

(asynchronous case) are considered. It is shown that GSP is able to achieve significant energy savings with only affecting the network performance slightly.

Adaptive Energy Efficient and Reliable Gossip Routing (AEERG) protocol for mobile ad hoc networks was proposed in [Rajeswari and Venkataramani 2010], which aims to provide both energy efficiency and reliability in gossip based routing. In the network layer solution AEERG, nodes can be in active mode or sleep node based on a gossip probability, and for each node the number of neighbors that are in active mode are adjusted based on the packet delivery ratio. In contrast to traditional ad hoc routing protocols, this epidemic approach does not require all nodes to be active listening incoming packets, and hence it helps reducing energy consumption and improving delivery ratio and reliability. Using sleep/wakeup approach, it also increases the lifetime of the network.

5.2 Resource Allocation

A novel approach for reducing the energy consumption of epidemic data dissemination in wireless sensor networks is proposed in [Garbinato et al. 2010]. This distributed approach adjusts the transmission range of nodes before they transmit messages, thus allocates wireless communication resources in an energy efficient way without degrading reliability. The adjustment of transmission range is done according to power-law distribution, and this ensures that many nodes can access few neighbors and few nodes can access many neighbors. The approach was applied to traditional epidemic algorithms, namely *Probabilistic Gossiping* (PG), *Counter-Based Scheme* (CBS), *Hop-Count Aided Broadcasting* (HCAB) and *Six-Shot Broadcast* (6SB), at the network layer.

The PG algorithm does not cause any delays during message forwarding and uses a specific probability when sensors broadcast a received message. This scheme is classified as context unaware since it does not use any prior information. CBS and HCAB schemes exploit message counting mechanism to set message forwarding delays. Additionally, in HCAB hops are taken into account and extra delay added to the total message forwarding delay, however the reliability is strengthened. CBS and HCAB use network traffic information before determining the message forwarding delays, so they are categorized under traffic aware context type. In the 6SB, on the contrary, delays are set according to the location of the six settled targets and the closest located target selected among them to set shorter delay, therefore this scheme is considered as location aware context type.

The performance of the algorithms were evaluated in terms of delivery and forward ratios. The results confirm that PG outperforms the other algorithms in terms of delivery and forward ratio, but it performs the worst in terms of energy consumption. Overall, CBS, HCAB and 6SB reduce energy consumption by up to 75% in comparison to the PG algorithm. Furthermore, the power-law modulation of wireless communication generates a routing topology with scale-free characteristics (i.e., many sensors select short ranges while few sensors select long ranges). Therefore, it helps efficient data dissemination and reduces the power consumption.

For ad hoc networks, an information dissemination method that combines epidemic principles with adaptive modulation and coding (AMC) is proposed in [Kontos et al. 2014]. This approach aims at achieving high network coverage and minimum energy consumption by adjusting message forwarding probability and coding mode using information from the lower network layers, thus it adaptively allocates network resources by cross-layer approach with channel state awareness. More precisely, the

idea is to identify the most convenient combination of message forwarding probability and coding mode.

To eliminate the redundant transmissions and high error control code (ECC) overhead, epidemic dissemination of the information is used. The nodes in the network are categorized into two groups, namely susceptible and infected. Susceptible nodes do not transmit any information and infected nodes broadcast the information to infect the other nodes, hence the information is spread with a certain message forwarding probability. Then, the message forwarding probability is combined with the coding mode to generate successful reception probabilities referred as evaluation indicators. Four different schemes, namely, time to full coverage (T2FC), time to 90% coverage (T29C), energy to full coverage (E2FC), and energy to 90% coverage (E29C) are compared, and the effect of modifying the indicators is explored. The results reveal that these schemes reduce the energy consumption in comparison to the static AMC scheme. The proposed approach contributes to adaptive epidemic data dissemination by supporting it with the flexibility of AMC.

5.3 Structure Optimization

Some energy efficient mechanisms for epidemic protocols focus on structure optimization [van Renesse 2002; Ozkasap et al. 2014]. In terms of their power usage, efficiency of the three aforementioned models of epidemic protocols (i.e., basic, neighborhood and hierarchical epidemics) was first examined in [van Renesse 2002] to show their suitability in power constrained networks such as wireless sensor and mobile ad hoc networks. Basic epidemic needs all peers to have knowledge about the entire population and it is inefficient in its power usage since it does not consider geographical location of peers to contact. Thus, it is not suitable for energy aware applications. On the other hand, both neighborhood epidemics and hierarchical epidemics (a structure optimization approach) can use power more efficiently by considering locality of peers. However, they need to deal with the trade-off between scalability and latency. In neighborhood epidemics, peers exchange information with neighbors in proximity, and a peer's power consumption amount is independent of population size. However, in large scale settings, neighborhood epidemic is not practical, and latency increases roughly polynomial with the population size. On the other hand, in hierarchical epidemics, peers are structured into a hierarchy of regions, and it has logarithmic latency growth with the population size and the power usage increases with population size. However, this study evaluates different epidemics at the network layer through simulations only and provides results on latency and power proportional to the gossip rate. Moreover, effects of gossip parameters such as fan-out and maximum gossip message size were not investigated.

A recent study addresses power awareness features and develops energy cost model formulations for flat and hierarchical epidemics [Ozkasap et al. 2014]. It also proposes energy cost models for generic peers using epidemic communication, and examines the effect of protocol parameters to characterize energy consumption. A novel *hierarchical epidemic* approach based on structure optimization while constructing the overlay hierarchy is proposed. This application layer epidemic approach utilizes the benefits of both flat epidemic and hierarchical approaches, uses only local knowledge and provides the possibility of active/passive peers to save energy. Robustness against peer failures is improved and the message overhead is significantly reduced thanks to the hierarchy. As a case study protocol, ProFID [Cem and Özkasap 2013], an epidemic protocol for frequent items discovery in P2P systems, is used in

[Ozkasap et al. 2014]. Through large scale simulations on PeerSim, the effect of protocol parameters on energy consumption, and comparisons between flat and hierarchical epidemic approaches are carried out.

In the hierarchical epidemic model in [Ozkasap et al. 2014], the dominating set idea is used to build a high-level overlay, as illustrated in Figure 6. A Dominating Set (DS) can be defined as a subset B of a graph $G=(V,E)$ such that every vertex in G is either in B or adjacent to a vertex in B . In the approach proposed [Ozkasap et al. 2014], a DS corresponds to a subset of peers such that a peer in the system is either in DS or a neighbor of a DS peer. There are two types of peers in the system, namely *dominating set (DS) peer* and *non-dominating set (NDS) peer*. The aim is to save energy by reducing the number of peers performing gossip operation. NDS peers are excluded from gossip operations. DS peers collect local states of NDS neighbors. Consequently, DS peers form a high level overlay topology on which gossiping is performed. There are two main advantages of this approach. First, NDS peers send their local state to one or multiple DS neighbors and, then they switch to a passive mode in which they just wait for the result (such as the set of frequent items discovered) to be announced by a DS neighbor. During this time period, they contribute to the reduction of energy consumption since they are in passive mode. Second, only DS peers participate in gossiping and, hence, the convergence time and message complexity of the epidemics is reduced.

Fig. 6. Construction of high level overlay using dominating set.

5.4 Message Reduction

A *Gossip-based Protocol for Aggregation* (GPIA) in wireless sensor networks which considers energy efficiency and fault tolerance is proposed in [Fauji and Kalpakis 2011]. GPIA disseminates information via epidemic communication at the application layer, and in contrast to pure gossiping, nodes perform selective message reception and early convergence detection of aggregation that achieves reduced energy consumption. By decreasing the number of messages for performing in-network aggregate computations, the network lifetime also increases significantly in comparison to other aggregation approaches namely *Ridesharing*, *Synopsis Diffusion* and the pure gossip.

The *Ridesharing* approach uses a robust directed acyclic graph (DAG) in communication, but leads to a higher energy consumption due to duplicates and node failures result in errors for data aggregation phase. *Synopsis Diffusion* uses co-centric ring technique that provides a multi-hop message communication from the sink node to the target node. Compared with the *Ridesharing* approach, *Synopsis Diffusion* is

superior in avoiding the duplicates by using hash-based synopses in the gossiping phase. The pure gossip approach is not energy efficient in terms of data dissemination and it is robust to prevent node failures. However, it performs well in single or two-hop message communication.

Considering energy constrained wireless sensor networks, the *NBgossip* algorithm, based on neighborhood epidemics and network coding, is proposed in [Lu et al. 2008] as a network layer solution. In general, gossiping protocols can be divided into two main components, namely, communication partner selection and message transmission. For instance, the uniform gossip algorithm randomly selects the nodes it communicates with and sends randomly selected messages to a specific node. In *Spatial gossip* [Kempe et al. 2004], message transmission is the same with uniform gossip. However, the communication partner selection is carried out by choosing among the neighbor nodes.

The *Alge-gossip* [Deb et al. 2006] algorithm also follows the same partner selection as uniform gossip for communication. However, a network coding strategy linearly combining the received data into a single message is used for message transfer. In *NBgossip*, nodes communicate with neighbor nodes only (i.e., the same as in *Spatial gossip*) and instead of forwarding messages received, nodes transmit linear combinations of messages (i.e., the same as in *Alge-gossip*).

Theoretical and simulation findings indicate that *NBgossip* outperforms uniform gossip over 90% and spatial gossip around 86% in terms of energy savings. For both *NBgossip* and *Alge-gossip*, similar numbers of rounds are required to disseminate information. However up to 60% energy savings are achieved by the *NBgossip* due to message transmission between neighbor nodes only. When computational complexities of the approaches are compared, Uniform and Spatial gossip have the same computational complexity due to the random message selection behavior. However, *Alge-gossip* has $O(n)$ and *NBgossip* has $\theta(\log n)$ computational complexity, where n denotes the number of messages linearly combined.

Epidemic routing can also be helpful in *Delay Tolerant Networks* (DTNs) to address energy constraints. Since DTNs are characterized by sparse network topologies and node mobility, message forwarding is done by intermediate nodes using random contacts. Therefore, epidemic algorithms would fit with their fully distributed nature and ease of implementation to transmit messages. A theoretical study for DTNs focuses on energy efficient epidemic routing where each node decides to forward a message by considering its remaining energy and age of the message [Khouzani et al. 2012]. Mathematical modeling of the trade-off between energy saving and forwarding efficiency shows that the optimal message forwarding decisions can be simply based on thresholds that depend on the nodes' remaining energy. Furthermore, analytical characterization of the dependence of the thresholds on the remaining node energy is explored. For instance, considering a node with low remaining energy, it is easy to predict that the node is not eager to transfer the message due to critical energy problems. On the other hand, higher remaining energy on a node causes longer message forwarding periods.

Another epidemic routing approach [De Rango et al. 2013] considering optimization of energy consumption and message delivery probability in DTNs is based on an *n-epidemic* routing mechanism (i.e. a node is able to transmit information when it has at least n neighbors in its transmission range). Choosing the value of n is crucial, since when it is too large the probability of receiving the message by the neighbor node will be low. Conversely, when n is too small, then the probability of transmit-

ting the message from source node to neighbor nodes will be high, and thus the energy consumption increases. Consequently, n is adjusted dynamically with the prevalence strategy, which simply chooses n based on neighbor relations of the nodes, and the nodes' energy levels. Simulation results confirm that the prevalence strategy reduces the energy consumption and increases the message delivery probability, due to its dynamic adjustment of n , in comparison with traditional n -epidemic routing approaches.

As an extension to the aforementioned approach, an improved prevalence strategy that considers number of hops and end-to-end delays is proposed in [De Rango and Amelio 2013]. In this modified prevalence strategy at the network layer, n is determined according to the average number of neighboring nodes and energy levels for a particular node. Basically, the average number of neighboring nodes is computed as follows. First, each node periodically computes the number of neighbors in a distributed manner and locally, then each node generates its own vector v_i by using the computed neighboring node information. Lastly, the average number of neighbors for a particular node is computed by dividing the sum of vector components by the number of vector components, where the number of vectors is equal the number of nodes. After the computation, simulations are conducted with preset threshold values.

The performance of the modified prevalence strategy is compared with the classical prevalence strategy and n -epidemic routing. The results show that the modified prevalence strategy outperforms the classical prevalence strategy and n -epidemic ($n=7$) in terms of energy efficiency, due to its dynamic adjustment of n , leading to significant energy reduction. However, n -epidemic ($n=7$) performs better in terms of message delivery probability in comparison to the modified prevalence strategy, since the threshold value for a dynamic n is not well-chosen for the modified prevalence strategy. Furthermore, it is observed that the modified prevalence strategy reduces the average number of hops in message delivery and the end-to-end delays in data delivery since lower number of messages is transmitted in the network in comparison with n -epidemic approach.

An energy efficient gossip (*E-Gossip*) protocol that allows minimum energy consumption of sensor nodes and maximum lifetime for wireless sensor networks is proposed in [Lee et al. 2014]. In *E-Gossip* network layer protocol, the remaining energy of a sensor node is used to determine the gossip probability. For example, when the remaining energy of a sensor node is less than the average remaining energy of all nodes, then the gossip probability is decreased and the node enters sleep mode. Thus, the low gossip probability reduces redundant message transmissions and helps to prolong the network lifetime. On the other hand, if the remaining energy of a sensor node is higher than the average remaining energy, then it becomes an active node and possibly takes place in gossip operation with a high gossip probability (the preset gossip probability in active mode is 80%). To evaluate the performance of *E-Gossip*, its performance is compared with the classical gossip protocol [Hedetniemi et al. 1988] and *AODV* (Ad hoc On demand Distance Vector) [Perkins and Royer 1999] protocol in terms of average remaining energy and average message delivery. In the classical gossip protocol, nodes communicate with each other based on periodical random selection, without any preset gossip probability. Conversely, in *AODV*, nodes leverage the information about link connections stored by each node inside the routing table. For instance, if any disconnected links occur between hops while sending a packet to the source node, then disconnected links are deleted from the routing table for successful packet delivery. In wireless sensor networks, the sink node collects da-

ta from all the other sensor nodes through multi-hop communication and, hence, its neighbors have higher energy consumption compared to all the other nodes. Therefore, by consuming all their remaining energy, these nodes become incapable to operate and the sink node is not able to receive data from the other nodes. The performance analysis, carried out in terms of average remaining energy and average message delivery, shows that E-Gossip outperforms classical gossip and AODV, since E-Gossip reduces the number of messages transmitted when the remaining energy of a sensor node is low.

5.5 Open issues & Research guidelines

In this section, we discuss some open issues and potential enhancements on energy efficient P2P epidemics. In our discussion we refer to the different categories considered in our taxonomy.

When dealing with sleep/wakeup solutions, an open issue in the context of energy efficient P2P storage systems concerns the ability of the system to handle load variations [Lefebvre and Feeley 2000]. Efficiently estimating future load when putting peers into sleep mode is an open problem that can be investigated. Based on the estimated future load, the sleep time of peers can be adjusted dynamically. Another open issue concerns the possible improvements that caching can provide [Lefebvre and Feeley 2000]. For instance, if the main priority is to increase the ratio of satisfied peer requests, then caching popular files and keeping few replicas for unpopular files could help in reducing the total storage cost.

For approaches based on resource allocation, additional mechanisms for mobility, location information and failure detection could be considered [Garbinato et al. 2010]. For instance, the power-law gossiping approach can be easily adapted to mobile devices due to its local and decentralized nature. By using the location information, a mobile device can become aware of devices closer to itself for communication. Besides, since power-law gossiping allows each node to adjust its transmission range for data dissemination, mobile devices can also use this property before transmitting data to other mobile devices in a decentralized way. Obviously, in wireless sensor networks, locations of the sensors would affect the energy consumption. Therefore, location information could be involved into energy calculations for data dissemination. When evaluating the system's performance, it would be more realistic to take into account the occurrence of failures, types of failures and efficient methods for failure detection. Furthermore, context aware adaptive epidemic schemes can be optimized to reduce energy consumption [Kontos et al. 2014].

When structure optimization techniques are considered, the tradeoff between the accuracy of the data collected through the structured overlay, and the energy consumption of the system, can be further studied. For example, identifying the optimal percentage of peers that should be kept active in the overlay to achieve a certain range of accuracy in the results is an open issue that can be further investigated.

Message redundancy is an open issue in epidemic protocols that has the potential for further improvement. For instance, in order to reduce the number of redundant messages, optimizations can be performed in data aggregation process [Fauji and Kalpakis 2011]. Thus, the energy consumption could be significantly decreased and better results would be achieved in terms of system performance and network lifetime. Furthermore, by extending the GPIA approach [Fauji and Kalpakis 2011], computing complex aggregates and functions in an energy efficient way would become possible. Further research can be conducted to determine the optimal gossip

probability and find out the arguments affecting the energy consumption and data delivery [Lee et al. 2014]. Moreover, improvements can be introduced for adjusting transmission power of sensor nodes to reduce the energy consumption. For example, one solution is keeping the distance of sensor nodes close to each other to decrease the transmission power, therefore the energy efficiency improves. Finally, epidemic routing approach in DTNs leads to longer message forwarding periods when the remaining energy of the node is higher [Khouzani et al. 2012; De Rango and Amelio 2013; De Rango et al. 2013]. To overcome this problem, delays in message forwarding should be studied and optimized.

Overall, when different classes of solutions for energy efficiency are considered, combining multiple approaches might help to achieve better performance. For example, in a structure optimization solution, the overlay structure can be built by determining the optimal number of active peers in the system based on the mechanisms used in sleep/wakeup approaches. Then, the overlay can be set to include the active peers. Another approach could be achieving efficient message reduction and resource allocation using the ideas in structure optimization solutions. Investigating the benefits of such hybrid mechanisms extensively could be an area for further research for P2P epidemic protocols and their energy efficiency.

6. CONCLUSIONS

In recent years, energy efficiency has emerged as one of the most important design requirements for modern large-scale distributed systems, such as P2P systems and applications, which are regularly used by a large number of users, both in desktop and mobile environments. Designing energy efficient P2P system solutions is a non-trivial task as the Quality of Service (QoS) constraints imposed by the applications need to be well fulfilled dynamically. In this paper, we have studied and classified different system designs and methods to achieve energy efficiency in P2P systems and applications. The recent developments have been discussed and categorized into six main classes (sleep/wakeup, proxy-based, peer selection, resource allocation, structure optimization, and message reduction) with special focus on three main application areas, namely P2P file sharing/distribution, content streaming and epidemic protocols.

Our survey indicates that the proxy-based approaches are the most popular amongst others to design energy efficient P2P file sharing/distribution systems. The reason for this popularity is that, on the one hand, these approaches almost guarantee the performance of such applications in terms of downloading speed and on the other hand they can be implemented/developed as an add-on with no (or little) change in the original file sharing applications (e.g., BitTorrent, Gnutella). Another finding of the survey shows that the resource allocation approaches have been the most used ones in the content streaming applications as they are suitable for mobile devices equipped with several wireless network interfaces. Our investigation also indicates that the probabilistic message forwarding behavior, as a key principle of epidemic protocols, causes the message reduction approaches considered more appropriate by researchers in such protocols.

While a number of solutions are already available, we believe there is still room for research activities and improvements in the considered areas. Therefore, we investigated open issues and research guidelines for the future works for each application area at the end of the corresponding sections. It is worth mentioning that, there

is a general “concern” in all the system solutions (independent from application) reflected in this survey, which can be also considered by researchers in the future. In fact, most of the solutions rely on simple energy models in which energy cost only considers the energy consumption of individual peers (baseline energy consumption, energy consumed for processing, communication, etc.) while for an accurate modeling, the energy cost should also consider energy consumption related to network communication (e.g, energy consumed by all intermediate routers, etc.). From our point of view, this survey provides a structural taxonomy in the field of energy efficiency of P2P systems within which the reader can situate both existing and future works based on the application.

ACKNOWLEDGMENTS

This work was partially supported by the COST (European Cooperation in Science and Technology) framework, under Action IC0804, and by TUBITAK (The Scientific and Technical Research Council of Turkey) under Grant 109M761.

REFERENCES

- Y. Agarwal, S. Hodges, R. Chandra, J. Scott, P. Bahl, and R. Gupta. 2009. Somniloquy: Augmenting Network Interfaces to Reduce PC Energy Usage. In *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. 365–380.
- F. Albiero, M. Katz, and F. H.P. Fitzek. 2008. Energy-Efficient Cooperative Techniques for Multimedia Services over Future Wireless Networks. In *Communications, ICC. IEEE International Conference on*. 2006–2011.
- G. Anastasi, I. Giannetti, and A. Passarella. 2010. A BitTorrent proxy for Green Internet file sharing: Design and experimental evaluation. *Computer Communications* 33, 7 (2010), 794 – 802. 0140-3664.
- L. L.H. Andrew, A. Sucevic, and T. T.T. Nguyen. 2011. Balancing peer and server energy consumption in large peer-to-peer file distribution systems. In *Online Conference on Green Communications (GreenCom), IEEE*. 76–81.
- G. Angelopoulos, M. Médard, and A. P. Chandrakasan. 2011. Energy-Aware Hardware Implementation of Network Coding. In *NETWORKING 2011 Workshops*. Lecture Notes in Computer Science, Vol. 6827. Springer Berlin Heidelberg, 137–144.
- J. Baliga, R. Ayre, K. Hinton, and R. S. Tucker. 2009. Architectures for Energy-Efficient IPTV Networks, In *Optical Fiber Communication Conference and National Fiber Optic Engineers Conference*.
- B. Barua, P. Karunakaran, H. Bagheri, and M. Katz. 2013. Energy and delay efficient cooperative media content download. In *Wireless Days (WD), IFIP*. 1–4. 2156-9711.
- A. R. Bharambe, C. Herley, and V. N. Padmanabhan. 2005. Analyzing and improving bittorrent performance. *Microsoft Research*, 2005–03.
- K. P. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky. 1999. Bimodal Multicast. *ACM Transactions on Computer Systems* 17, 2 (May 1999), 41–88. 0734-2071.
- J. Blackburn and K. Christensen. 2009. A Simulation Study of a New Green BitTorrent. In *Communications Workshops, ICC Workshops. IEEE International Conference on*. 1–6.
- Bluetooth.org. 2014. Bluetooth® Specification Version 4.2. (Dec 2014). <https://www.bluetooth.org/en-us/specification/adopted-specifications>
- Y. Carlinet, H. Debar, Y. Gourhant, and L. Mé. 2010. Caching P2P Traffic: What Are the Benefits for an ISP?. In *Networks (ICN), Ninth International Conference on*. 376–383.
- E. Cem, E. Demirkaya, E. Esiner, B. Ozaydin, and O. Ozkasap. 2012. Energy Cost Model for Frequent Item Set Discovery in Unstructured P2P Networks. In *Computer and Information Sciences II*, Springer London, 117–123. 978-1-4471-2154-1
- E. Cem and Ö. Ozkasap. 2013. ProFID: Practical frequent items discovery in peer-to-peer networks. *Future Generation Computer Systems* 29, 6 (2013), 1544 – 1560.
- A. Chandrasekar, K. Chandrasekar, H. Ramasatagopan, and A. R. Rafica. 2012. SMC: An Energy Conserving P2P File Sharing Model for Mobile Devices. In *Proceedings of the Eleventh ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE '12)*. 66–73.
- M.H. Chen, C.F. Chou, K.H. Lee, and C.Y. Chang. 2013. On Cooperative Energy-Efficient P2P Live Streaming System for Mobile Hotspots. In *Green Computing and Communications (GreenCom), IEEE and Internet of Things (iThings/CPSCoM), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*. 195–199.
- M.H. Chen, C.Y. Yang, C.Y. Chang, M.Y. Hsu, K.H. Lee, and C.F. Chou. 2011. Towards Energy-efficient

- Streaming System for Mobile Hotspots. In *Proceedings of the ACM SIGCOMM Conference*. 450–451.
- B. Cohen. 2003. Incentives build robustness in BitTorrent. In *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems*. 67–72.
- D.N. da Hora, D. F. Macedo, J. M. S. Nogueira, and G. Pujolle. 2007. Optimizing Peer-to-Peer content discovery over wireless mobile ad hoc networks. In *Mobile Wireless Communications Networks, 9th IFIP International Conference on*. 6–10.
- F. De Rango and S. Amelio. 2013. Performance evaluation of scalable and energy efficient dynamic n-epidemic routing in Delay Tolerant Networks. In *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), International Symposium on*. 167–173.
- F. De Rango, S. Amelio, and P. Fazio. 2013. Enhancements of Epidemic Routing in Delay Tolerant Networks from an energy perspective. In *Wireless Communications and Mobile Computing Conference (IWCMC), 9th International*. 731–735.
- S. Deb, M. Médard, and C. Choute. 2006. Algebraic gossip: a network coding approach to optimal multiple rumor mongering. *IEEE Transactions on Information Theory* 52, 6 (June 2006), 2486–2507. 0018-9448.
- G. Ding and B. Bhargava. 2004. Peer-to-peer file-sharing over mobile ad hoc networks. In *Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops*. 104–108.
- T. Enokido, K. Suzuki, A. Aikebaier, and M. Takizawa. 2010. Laxity Based Algorithm for Reducing Power Consumption in Distributed Systems. In *International Conference on Complex, Intelligent and Software Intensive Systems (CISIS)*. 321–328.
- P. Th. Eugster, R. Guerraoui, S. B. Handurukande, P. Kouznetsov, and A.M. Kermarrec. 2003. Lightweight Probabilistic Broadcast. *ACM Transactions on Computer Systems* 21, 4 (Nov. 2003), 341–374.
- S. Fauji and K. Kalpakis. 2011. A gossip-based energy efficient protocol for robust in-network aggregation in wireless sensor networks. In *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. 166–171.
- A. Feldmann, A. Gladisch, M. Kind, C. Lange, G. Smaragdakis, and F.J. Westphal. 2010. Energy trade-offs among content delivery architectures. In *Proceeding of the 9th Conference on Telecommunications Internet and Media Techno Economics (CTTE)*. 1–6.
- A. Fiandrotti, V. Bioglio, M. Grangetto, R. Gaeta, and E. Magli. 2014. Band Codes for Energy-Efficient Network Coding With Application to P2P Mobile Streaming. *IEEE Transactions on Multimedia* 16, 2 (Feb 2014), 521–532. 1520-9210
- M. Forshaw and N. Thomas. 2013. A Novel Approach to Energy Efficient Content Distribution with BitTorrent. In *Computer Performance Engineering, Lecture Notes in Computer Science*, Vol. 7587. Springer Berlin Heidelberg, 188–196.
- P. Xiang Gao, A. R. Curtis, B. Wong, and S. Keshav. 2012. It's not easy being green. *SIGCOMM Comput. Commun. Rev.* 42, 4 (August 2012), 211–222.
- B. Garbinato, D. Rochat, M. Tomassini, and F. Vessaz. 2010. Injecting power-awareness into epidemic information dissemination in sensor networks. *Future Generation Computer Systems* 26, 6 (2010), 868–876.
- A. E. Garcia, R. Weidlich, L. R. de Lope, K. D. Hackbarth, H. Hlavacs, and C. San Leandro. 2010. Approximation Towards Energy-efficient Distributed Environments. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools '10)*. Brussels, Belgium.
- I. Giannetti, G. Anastasi, and M. Conti. 2012. Energy-efficient P2P file sharing for residential BitTorrent users. In *IEEE Symposium on Computers and Communications (ISCC)*.
- Y. Guo, C. Liang, and Y. Liu. 2008. dHCPS: decentralized hierarchically clustered p2p video streaming. In *Proceedings of the international conference on Content-based image and video retrieval (CIVR '08)*. ACM, New York, NY, USA, 655–662.
- J.S. Han, J.W. Song, T.H. Kim, and S.B. Yang. 2008. Double-layered Mobile P2P Systems Using Energy-Efficient Routing Schemes. In *Telecommunication Networks and Applications Conference, ATNAC. Australasian*. 122–127.
- S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman. 1988. A survey of gossiping and broadcasting in communication networks. *Networks* 18, 4 (1988), 319–349. 1097-0037.
- J. Heide, M. V. Pedersen, F. H.P. Fitzek, and T. Larsen. 2008. Cautious view on network coding - From theory to practice. *Journal of Communications and Networks* 10, 4 (Dec 2008), 403–411.
- J. Heide, M. V. Pedersen, F. H.P. Fitzek, and T. Larsen. 2009. Network Coding for Mobile Devices - Systematic Binary Random Rateless Codes. In *IEEE International Conference on Communications Workshops, ICC Workshops*. 1–6.
- H. Hlavacs, K. A. Hummel, R. Weidlich, A. M. Houyou, and H. De Meer. 2010. Modelling Energy Efficiency in Distributed Home Environments. *International Journal of Communication Networks and Distributed Systems* 4, 2 (Jan. 2010), 161–182. 1754-3916.
- H. Hlavacs, R. Weidlich, and T. Treutner. 2008. Energy saving in future home environments. In *1st IFIP Wireless Days (WD '08)*. 1–5.
- H. Hlavacs, R. Weidlich, and T. Treutner. 2012. Energy efficient peer-to-peer file sharing. *The Journal of*

- Supercomputing* 62, 3 (2012), 1167–1188.
- X. Hou, D. Tipper, and S. Wu. 2006. A Gossip-Based Energy Conservation Protocol for Wireless Ad Hoc and Sensor Networks. *Journal of Network and Systems Management* 14, 3 (2006), 381–414.
- J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. 2012. A Close Examination of Performance and Power Characteristics of 4G LTE Networks. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys '12)*. ACM, 225–238.
- Y. Huang, T. Z.J. Fu, D.M. Chiu, J. C.S. Lui, and C. Huang. 2008. Challenges, Design and Analysis of a Large-scale P2P-vod System. In *Proceedings of the ACM SIGCOMM Conference on Data Communication (SIGCOMM '08)*. ACM, 375–388.
- M. Jelasity, A. Montresor, and O. Babaoglu. 2005. Gossip-based Aggregation in Large Dynamic Networks. *ACM Transactions on Computer Systems* 23, 3 (Aug. 2005), 219–252. 0734-2071.
- M. Jimeno and K. Christensen. 2007. A Prototype Power Management Proxy for Gnutella Peer-to-Peer File Sharing. In *Local Computer Networks, 2007. LCN. 32nd IEEE Conference on*. 210–212.
- G. Jourjon, T. Rakotoarivelo, and M. Ott. 2010. Models for an Energy-Efficient P2P Delivery Service. In *18th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*. 348–355.
- K. Kant. 2010. Challenges in Distributed Energy Adaptive Computing. *ACM SIGMETRICS Performance Evaluation Review* 37, 3 (Jan. 2010), 3–7. 0163-5999.
- I. Kelényi, Á. Ludanyi, and J. K. Nurminen. 2010a. BitTorrent on mobile phones - energy efficiency of a distributed proxy solution. In *International Green Computing Conference*. 451–458.
- I. Kelényi, Á. Ludanyi, and J. K. Nurminen. 2011a. Distributed BitTorrent proxy for energy efficient mobile content sharing. In *Wireless Personal Multimedia Communications (WPMC), 14th International Symposium on*. 1–5.
- I. Kelényi, Á. Ludanyi, and J. K. Nurminen. 2011b. Energy-efficient BitTorrent downloads to mobile phones through memory-limited proxies. In *Consumer Communications and Networking Conference (CCNC), IEEE*. 715–719.
- I. Kelényi, Á. Ludanyi, J. K. Nurminen, and I. Puustinen. 2010b. Energy-efficient mobile bittorrent with broadband router hosted proxies. In *Wireless and Mobile Networking Conference (WMNC), Third Joint IFIP*. 1–6.
- I. Kelényi and J. K. Nurminen. 2008a. Energy Aspects of Peer Cooperation Measurements with a Mobile DHT System. In *IEEE International Conference on Communications Workshops, ICC Workshops '08*. 164–168.
- I. Kelényi and J. K. Nurminen. 2008b. Optimizing Energy Consumption of Mobile Nodes in Heterogeneous Kademlia-Based Distributed Hash Tables. In *The Second International Conference on Next Generation Mobile Applications, Services and Technologies, NGMAST '08*. 70–75.
- I. Kelényi and J. K. Nurminen. 2010. Cloudtorrent – Energy-efficient BitTorrent Content Sharing for Mobile Devices via Cloud Services. In *Proceedings of the 7th IEEE Conference on Consumer Communications and Networking Conference (CCNC'10)*. 646–647.
- I. Kelényi, J. K. Nurminen, Á. Ludányi, and T. Lukovszki. 2012. Modeling resource constrained BitTorrent proxies for energy efficient mobile content sharing. *Peer-to-Peer Networking and Applications* 5, 2 (2012), 163–177. 1936-6442.
- D. Kempe, J. Kleinberg, and A. Demers. 2004. Spatial Gossip and Resource Location Protocols. *Journal of The ACM* 51, 6 (Nov. 2004), 943–967. 0004-5411.
- MHR. Khouzani, S. Eshghi, S. Sarkar, N. B. Shroff, and S. S. Venkatesh. 2012. Optimal Energy-aware Epidemic Routing in DTNs. In *Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '12)*. 175–182.
- T. Klingberg and R. Manfredi. 2002. Gnutella Protocol Specification Version 0.6. http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html. (2002).
- T. Kontos, C. Anagnostopoulos, and S. Hadjiefthymiades. 2014. Wireless Channel State-Aware and Adaptive Epidemic Dissemination in Ad Hoc Networks. *International Journal of Wireless Information Networks* 21, 1, 58–67. 1068-9605.
- J. F. Kurose and K. W. Ross. 2013a. Multimedia Networking Applications. In *Computer Networking, A Top-Down Approach (sixth edition)*. Pearson International Edition.
- J. F. Kurose and K. W. Ross. 2013b. Peer-to-Peer Applications. In *Computer Networking, A Top-Down Approach (sixth edition)*. Pearson International Edition.
- N. Laoutaris, P. Rodriguez, and L. Massoulie. 2008. ECHOS: Edge Capacity Hosting Overlays of Nano Data Centers. *ACM SIGCOMM Computer Communication Review* 38, 1 (Jan. 2008), 51–54. 0146-4833.
- B. Lee, H. K. Song, Y. Suh, K. H. Oh, and H. Y. Youn. 2014. Energy-Efficient Gossiping Protocol of WSN with Realtime Streaming Data. In *Dependable, Autonomic and Secure Computing (DASC), IEEE 12th International Conference on*. 219–224.
- Y.J. Lee, J.H. Jeong, H.Y. Kim, and C.H. Lee. 2010. Energy-saving set-top box enhancement in BitTorrent networks. In *Network Operations and Management Symposium (NOMS), IEEE*. 809–812.

- G. Lefebvre and M. J. Feeley. 2000. *Energy Efficient Peer-to-Peer Storage*. Technical Report TR-2003-17. Department of Computer Science, University of British Columbia.
- A. KH. Leung and YK Kwok. 2005. On topology control of wireless peer-to-peer file sharing networks: energy efficiency, fairness and incentive. In *World of Wireless Mobile and Multimedia Networks (WoWMoM). Sixth IEEE International Symposium on*. 318–323.
- A. KH. Leung and YK Kwok. 2008. On Localized Application-Driven Topology Control for Energy-Efficient Wireless Peer-to-Peer File Sharing. *IEEE Transactions on Mobile Computing* 7, 1 (Jan 2008), 66–80.
- Y. Li, Z. Li, M. Chiang, and A.R. Calderbank. 2008. Video transmission scheduling for peer-to-peer live streaming systems. In *IEEE International Conference on Multimedia and Expo*. 653–656.
- Y. Li, Z. Li, M. Chiang, and A. R. Calderbank. 2009. Energy-Efficient Video Transmission Scheduling for Wireless Peer-to-Peer Live Streaming. In *Consumer Communications and Networking Conference, CCNC. 6th IEEE*. 1–5.
- Z. Li, J. Huang, and A. K. Katsaggelos. 2007. Content reserve utility based video segment transmission scheduling for peer-to-peer live video streaming system. In *Proceedings of the 45th Allerton Conference on Communications, Control, and Computing*. 563–567.
- P. Lieberman. 2006. White Paper: Wake on LAN Technology. (2006).
- Y. Liu and M. Wang. 2013. An Energy Perspective of Multimedia Streaming Systems. In *Third International Conference on Cloud and Green Computing (CGC)*. 99–104.
- F. Lu, L.T. Chia, K.L. Tay, and W.H. Chong. 2008. NBgossip: An Energy-Efficient Gossip Algorithm for Wireless Sensor Networks. *Journal of Computer Science and Technology* 23, 3 (2008), 426–437.
- H. R. Maamar, G. R. Alonso, A. Boukerche, and E. Petriu. 2011. Energy-aware analysis for supplying partner selection in mobile P2P 3D streaming. In *IEEE Symposium on Computers and Communications (ISCC)*. 74–79.
- H. R. Maamar, G. R. Alonso, A. Boukerche, and E. Petriu. 2013. Energy management control for supplying partner selection protocol in mobile peer-to-peer three-dimensional streaming. *Concurrency and Computation: Practice and Experience* 25, 5 (2013), 752–769. 1532-0634.
- E. Magli and P. Frossard. 2009. An overview of network coding for multimedia streaming. In *Multimedia and Expo, ICME. IEEE International Conference on*. 1488–1491.
- A. Magnosto, R. Gaeta, M. Grangetto, and M. Sereno. 2010. TURINstream: A Totally pUsh, Robust, and efficient P2P Video Streaming Architecture. *IEEE Transactions on Multimedia* 12 (8), 901–914.
- A. Mawji and H. Hassanein. 2010. Optimal path selection for file downloading in P2P overlay networks on MANETs. In *Computers and Communications (ISCC), IEEE Symposium on*. 1133–1138.
- S. Miyake and M. Bandai. 2013. Energy-Efficient Mobile P2P Communications Based on Context Awareness. In *IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*. 918–923.
- M. M. Mohiuddin, P. M. Eittenberger, and U. R. Krieger. 2013. Unwired: A simulation test-bed for mobile P2P live streaming. In *Wireless Communications and Mobile Computing Conference (IWCMC), 9th International*. 1601–1606.
- S. Nedeveschi, S. Ratnasamy, and J. Padhye. 2008. Hot Data Centers vs. Cool Peers. In *Workshop on Power Aware Computing and Systems (HotPower '08)*. San Diego, CA, USA.
- B. Nordman and K. Christensen. 2010. Proxying: The Next Step in Reducing IT Energy Use. *Computer* 43, 1 (Jan 2010), 91–93. 0018-9162.
- J. K. Nurminen. 2010. Parallel Connections and their Effect on the Battery Consumption of a Mobile Phone. In *Consumer Communications and Networking Conference (CCNC), 7th IEEE*. 1–5.
- O. Ozkasap, M. Çaglar, and A. Alagoz. 2009a. Principles and performance analysis of SeCond: A system for epidemic peer-to-peer content distribution, *Journal of Network and Computer Applications*, 32, 3 (2009), 666 – 683.
- O. Ozkasap, M. Çaglar, E. Cem, E. Ahi, and E. Iskender. 2009b. Stepwise fair-share buffering for gossip-based peer-to-peer data dissemination, *Computer Networks*, 53, 13 (2009), 2259–2274.
- O. Ozkasap, M. Çaglar, E. S. Yazici, and S. Küçükçifçi. 2010. An analytical framework for self-organizing peer-to-peer anti-entropy algorithms. *Performance Evaluation* 67, 3 (2010), 141 – 159.
- O. Ozkasap, E. Cem, S. E. Cebeci, and T. Koc. 2014. Flat and hierarchical epidemics in P2P systems: Energy cost models and analysis. *Future Generation Computer Systems* 36, 0 (2014), 257-266. 0167-739X.
- O. Ozkasap, Z. Genc, and E. Atsan. 2006. Epidemic-based approaches for reliable multicast in mobile ad hoc networks, *ACM SIGOPS Operating Systems Review*, 40, 3 (2006), 73–79.
- O. Ozkasap, Z. Genc, and E. Atsan. 2009c. Epidemic-based reliable and adaptive multicast for mobile ad hoc networks. *Computer Networks* 53, 9 (2009), 1409 – 1430. 1389-1286.
- G. Peng. CDN: Content Distribution Network. Arxiv preprint cs/0411069, 2004.
- C. E. Perkins and E. M. Royer. 1999. Ad-hoc on-demand distance vector routing. In *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*. 90–100.
- K. Pussep, S. Kaune, O. Abboud, C. Huff, and R. Steinmetz. 2010. On energy-awareness for peer-assisted streaming with set-top boxes. In *International Conference on Network and Service Management*

- (CNSM). 166–173.
- D. Qiu and R. Srikant. 2004. Modeling and Performance Analysis of BitTorrent-like Peer-to-peer Networks. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '04)*. ACM, 367–378.
- A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. 2009. Cutting the electric bill for internet-scale systems. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication (SIGCOMM '09)*. ACM, New York, NY, USA, 123–134.
- M. Raj, K. Kant, and S. K. Das. 2013. Energy Adaptive Mechanism for P2P File Sharing Protocols. In *Euro-Par 2012: Parallel Processing Workshops*, Lecture Notes in Computer Science, Vol. 7640. Springer Berlin Heidelberg, 89–99.
- S. Rajeswari and Y. Venkataramani. 2010. An adaptive energy efficient and reliable gossip routing protocol for mobile adhoc networks. *International Journal of Computer Theory and Engineering* 2, 5 (2010), 740–745.
- M. Ramadan, L. El Zein, and Z. Dawy. 2008. Implementation and evaluation of cooperative video streaming for mobile devices. In *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*. 1–5.
- F. Ramos. 2013. *GREEN IPTV: a resource and energy efficient network for IPTV*. Ph.D. Dissertation. University of Cambridge.
- H. Schulze and K. Mochalski. 2009. Internet Study 2008/2009. *IPOQUE Report* 37, 351–362.
- H. Shojania and B. Li. 2009. Random Network Coding on the iPhone: Fact or Fiction?. In *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '09)*. ACM, New York, NY, USA, 37–42.
- A. Sucevic, L. L.H. Andrew, and T. T.T. Nguyen. 2011. Powering Down for Energy Efficient Peer-to-peer File Distribution. *ACM SIGMETRICS Performance Evaluation Review* 39, 3 (Dec. 2011), 72–76.
- D. A. Tran, K. A. Hua, and T. Do. 2003. ZIGZAG: an efficient peer-to-peer scheme for media streaming. *INFOCOM 2003. 22nd Annual Joint Conference of the IEEE Computer and Communications*. Vol. 2, 1283–1292.
- P. Trunfio. 2014. A two-layer model for improving the energy efficiency of file sharing peer-to-peer networks. *Concurrency and Computation: Practice and Experience* (2014), 1532–0634.
- V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez. 2009. Greening the Internet with Nano Data Centers. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT '09)*. ACM, 37–48.
- R. van Renesse. 2002. Power-aware epidemics. In *Proceedings of the 21st IEEE Symposium on Reliable Distributed Systems*. 358–361.
- P. Vingelmann, M. V. Pedersen, F. H. P. Fitzek, and J. Heide. 2010. Multimedia Distribution Using Network Coding on the Iphone Platform. In *Proceedings of the ACM Multimedia Workshop on Mobile Cloud Media Computing (MCMC '10)*. ACM, New York, NY, USA, 3–6.
- M. Wang and B. Li. 2007a. Lava: A Reality Check of Network Coding in Peer-to-Peer Live Streaming. In *INFOCOM. 26th IEEE International Conference on Computer Communications*. 1082–1090.
- M. Wang and B. Li. 2007b. Network Coding in Live Peer-to-Peer Streaming. *IEEE Transactions on Multimedia* 9, 8 (Dec 2007), 1554–1567. 1520–9210.
- Wi-Fi Alliance. 2014. Wi-Fi CERTIFIED Wi-Fi Direct®: Personal, portable Wi-Fi® technology. (Sep 2014). <http://www.wi-fi.org/file/wi-fi-certified-wi-fi-direct-personal-portable-wi-fi-technology-2014>
- M. Wichtlhuber, J. Ruckert, D. Stingl, M. Schulz, and D. Hausheer. 2012. Energy-efficient mobile P2P video streaming. In *Peer-to-Peer Computing (P2P), IEEE 12th International Conference on*. 63–64.
- M. K. H. Yeung and Y.K. Kwok. 2008. Energy efficient media streaming in wireless hybrid peer-to-peer systems. *Parallel and Distributed Processing, IPDPS. IEEE International Symposium on*. 1–11.
- Q. Zhang, F. H. P. Fitzek, and M. Katz. 2007. Cooperative Power Saving Strategies for IP-Services Supported over DVB-H Networks. In *Wireless Communications and Networking Conference (WCNC). IEEE*.