André Borges Sampaio

Licenciatura

Resource Sharing and Search in Partially Decentralized Mobile Networks

Relatório intermédio para obtenção do Grau de Mestre em Engenharia Informática

Orientadores : João Carlos Antunes Leitão,
Professor Auxiliar Convidado,
NOVA University of Lisbon
Nuno Manuel Ribeiro Preguiça,
Professor Associado,
NOVA University of Lisbon

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

FEVEREIRO, 2016

Abstract

Media sharing between smart-devices, such as smartphones and tablets, is a common habit between users. This happens typically through third party services such as social networks. In order to achieve this, users usually rely on an Internet connection, either through infrastructure or by relying on 3G/4G technology.

Under the mobile paradigm, direct connections between any pair of mobile devices are not always possible to establish and infrastructured networks solutions are not always available, especially on highly populated events such as concerts or medium to large social gatherings.

The goal of this work is to explore alternatives to address media sharing and search in the context of medium sized to large sized social gathering in mobile networks. We envision a case study application whose goal is to build and maintain a distributed image gallery using an hybrid (partially decentralized) edge computing network architecture. In which mobile devices communicate in a peer-to-peer fashion, except when such is not possible, or when resorting to an available infrastructure is a better option. The gallery can then be searched using sophisticated querying techniques such as facial recognition, that enables querying the network for photos that contain specific facial characteristics.

Keywords: media sharing; smart-devices; mobile; distributed gallery; partially decentralized; peer-to-peer; facial recognition

Resumo

A partilha de multimédia entre dispositivos móveis, como *smartphones* e *tablets*, é um hábito comum dos utilizadores. Esta partilha é feita tipicamente através de serviços como as redes sociais. De modo a aceder a estes serviços, os utilizadores dependem de uma ligação à *Internet*, seja através de uma infraestrutura ou mediante a tecnologia 3G/4G.

No paradigma móvel, nem sempre é possível estabelecer ligações diretas entre qualquer par de dispositivos. Para além disso, nem sempre é possível aceder a serviços que utilizam redes centralizadas, pois estes por vezes encontram-se indisponíveis. Este último impedimento verifica-se especialmente em eventos altamente populados, por exemplo em concertos ou em grandes reuniões.

Este trabalho pretende explorar soluções alternativas para a partilha e pesquisa de dados multimédia, no contexto das redes móveis. Focando-se em particular em situações que envolvem, grupos de pessoas de média a grande dimensão. Pretende-se utilizar como caso de estudo, uma aplicação cujo propósito é de montar e manter uma galeria distribuída que opera sobre uma rede de arquitetura híbrida (parcialmente descentralizada) que faz uso de computação no extremo. Na qual os dispositivos móveis comunicam diretamente entre si (*peer-to-peer*), exceto quando não é possível estabelecer ligação ou nos casos em que recorrer a uma das infraestruturas disponíveis seja mais vantajoso. A galeria será pesquisável, servindo-se de técnicas sofisticadas de procura, como reconhecimento facial que permite pesquisar a rede por características faciais especificas.

Palavras-chave: partilha; multimédia; dispositivos móveis; galeria distribuída; arquitetura híbrida; parcialmente descentralizada; computação no extremo; peer-to-peer; reconhecimento facial

Contents

List of Figures

List of Tables

1

Introduction

Smart-devices nowadays provide computation, storage, and networking resources on the move. These capabilities can be combined if several independent units communicate with each other forming a distributed infrastructure. This has recently been denominated as edge-cloud-computing.

Edge-computing on smartphones takes advantage of the aggregation of available resources from each individual device creating a mobile-edge-cloud that creates an opportunity not only to support file-sharing among users but also offers the possibility to perform distributed computations, minimizing the dependency of other resources or infrastructures therefore, boosting availability in situations where there is network congestion in data services or when those services are not available. For example, when there is a large number of users in the same location trying to upload photos or videos to share those contents with other users (potentially close by) over centralized social networks at the same time. Media sharing between mobile devices, whether it is a smartphone or a tablet, is a recurrent habit between smart-device users, especially in social gatherings[Ahe+07].

Recent approaches have explored this type of strategy and have studied the potential of edge clouds in media sharing as a novel way of publishing and retrieving multimedia content[Mar09]. This type of approach is aimed at poor or no cellular connectivity scenarios such as battlefield settings or highly populated events while also providing a way of sharing crowd-sourced data and computational power, for instance to perform extraction of features and execute distributed search algorithms over the multimedia content being shared.

In order to substantiate the research, previous approaches have modified an existing cloud computing system, Apache Hadoop, to run on Android mobile devices. The results obtained in their investigation allowed to conclude that the usage of this architecture results in more consistent latencies than uploading and serving files over a remote server[Mar09].

Motivation

Infrastructure-based architectures present several challenges like the required contact to a remote host through an Internet connection, or dealing with service unavailability due to infrastructure related problems. In contrast, infrastructure-less network architectures such as the one mentioned earlier, address some of these challenges while also leading to new ones such as the inherent complexity of performing distributed searching on a constantly changing network topology[Lun00]. An hybrid partially-decentralized network topology might have the capacity of mitigating issues present in both infrastructure-based and infrastructure-less architectures by maximizing availability, reducing latency and being more robust to data services overload or unavailability of centralized components.

Adopting this hybrid approach, relying both on smart-devices and remote infrastructure elements, can lead to the extraction of benefits of a fully decentralized (i.e, peer-to-peer) network while still enabling one to fallback to the infrastructure in scenarios when peer-to-peer is not available or not convenient when compared to a solution that resorts to a centralized remote host. Furthermore, the infrastructure can also be leveraged to improve the operation of decentralized mechanisms, such as routing among the peripheral devices, or user authentication, which are hard to tackle in fully decentralized architectures[FLR13].

These characteristics can not only provide mobile applications a with the fundamental mechanism for collecting and retrieving crowd-sourced data but also offers the potential to reduce user frustration through an improved availability that translates into a more fluid, and thus more satisfying experience.

Goals

The file-sharing capabilities of a semi-decentralized network design can empower applications with access to distributed data resources such as photos and videos, taking that into consideration, we envisioned the concrete case-study of a crowd-sourced distributed photo gallery that would feature image searching by facial recognition. Such an application would allow users to query the complete set of pictures taken by other users that are attending the same event or gathering. For example, searching for all pictures where you appear while attending a 100 person birthday party or getting photos from a specific perspective in a football match.

The work to be developed in the context of the thesis intends at exploring the capabilities of a semi-decentralized mobile architecture that enables the operation of such an application, providing access to distributed crowd-sourced data resources through the design and combination of a set of specific components. In particular:

Semi-Decentralized Overlay Network: an hybrid logical network topology designed to logically connect a set of mobile devices that takes into account hardware challenges, such as battery drainage, and the physical location of participants.

Distributed Photo Gallery: a crowd-sourced photo gallery populated by media retrieved from nearby devices, which extracts relevant features from this media.

Facial Recognition Querying: a feature that enables searching over a distributed photo gallery using facial recognition techniques that can leverage computation on the edge-mobile-cloud. The techniques used for facial recognition have already been developed in the FCT/CMU project[Hyr15] scope, in which this work is also included.

2

Related Work

The dissertation will address challenges related with the design of decentralized and partially decentralized distributed architectures with emphasis on the support for media sharing, and distributed querying. Hence, several aspects related with these topics have to be addressed. In the following we present a survey of relevant previous works that address challenges related and complementary to the goals of this work.

In Section 2.1 mobile computing both infrastructure-based and infrastructure-less architectures, client-server and ad-hoc respectively, are described and compared.

In Section 2.2 relevant peer-to-peer systems aspects are discussed and compared, focusing mainly on differences on these architectures accordingly with their degree of decentralization also surveying some alternative designs for overlay networks.

In Section 2.3 edge computing, especially the mobile-edge paradigm, is described and discussed.

In Section 2.4 security on mobile computing is discussed, specifically on wireless ad-hoc and edge computing networks.

2.1 Mobile Computing

Mobile computing distinguishes from the static variant mainly due to the mobility of nodes and the constraints associated to mobile resources such as limited battery life and wireless bandwidth and connectivity.

Even tough mobile devices present some resource restrictions, clients still expect the same level of service from applications as with their stationary counterparts. In order to achieve satisfying service performance and availability, infrastructure-based classical models like the client-server model need to be adapted and extended to fit the specific requirements of this environment[DD08].

2.1.1 Mobile Client-Server

Accordingly to the definition, presented in [JHE99] for client-server information system, a server is defined as any machine that holds a complete copy of one or more databases. A client the entity that communicates with the servers in order to interact or operate over the existing data.

The mobile environment, in certain tasks, blends these two roles to compensate for the resource limitations of mobile devices. By performing some client operations on resource-rich servers or by mimicking the functions of a server in a client to cope with unstable connectivity.

The amount of functions that are relocated among these entities to the other role classifies client-server architectures as:

Thin Client: This architecture moves a great part of the application logic and functionality from clients to servers. This type of architecture is suitable for scenarios where the client devices hardware properties do not meet the application requirements.

Full Client: These architectures emulate server functions on client devices, therefore allowing offline usage and minimizing the implications of connectivity uncertainty over the behavior of applications.

The ability to operate in a disconnected environment is used as fallback in intermittent, low bandwith, high latency, or high expense network cases in which network characteristics have degraded beyond the acceptable usability standards.

Flexible Client-Server: These architectures dynamically redirect and perform application logic on clients and servers. In order to boost performance and availability, it temporarily dims the distinction between mobile devices and stationary hosts[JHE99].

The architectures described above are used in accordance with the requirements of device hardware and expected connectivity scenarios. For instance, mobile thin client computing is an enabler for the execution of hardware demanding applications. Only requiring clients to display graphics and outputting results through virtual network computing (VNC) technology such as:

THINC is a virtual display architecture for thin-client computing that provides high fidelity display and interactive performance in both LAN and WAN environments. Instead of providing a real driver for a particular display hardware, THINC introduces a simple virtual display driver that intercepts drawing commands at the device layer, and sends them over to a client device to display[BKN05].

Another common alternative is to rely on web based applications. Whose logic is fully executed on the server side, and over which the client interacts through a web interface. A good example is:

Office Online is a collection of office productivity tools, previously available offline, such as a word processor (Word), spreadsheet (Excel), slide show presentation (Powerpoint). That can be used through the Internet via web browser[Mic16].

Mobile full client applications such as, photo editing tool, Photoshop[**Adobe2016**] rely on the capabilities of the client, CPU and local storage. By avoiding to resort to a central server and operating locally, they provide a larger set of features and a more fluid user experience (albeit being essentially local applications with few to no distributed aspects).

Mobile flexible client applications use the best of both worlds, by processing locally and using server resources as required. Video games like Diablo 3[Bli16] implement this type of architecture as they need both thin and full client characteristics to support graphics, and store and provide game content. There are also approaches that use this model over the classic client-server on other fields such as e-commerce [Mah12], and object recognition and tracking with augmented reality [Gam10].

2.1.2 Mobile Ad-Hoc Networks

A Mobile Ad-hoc Network (MANET) is a set of wireless mobile hosts dynamically forming a network without the use of an existing network or centralized coordination infrastructure[Tse+02; Su+09].

2.1.3 Routing Protocols

Distributed coordination requires that nodes follow some type of protocol to communicate efficiently. Several protocols have been proposed such as DSDV (Destination Sequenced Distance Vector) and DSR(Dynamic Source Routing)[SWS12]. Routing protocols can be classified as follows:

Proactive protocols are table driven, in other words, each node contains a routing table. This table contains the address of the nodes he may connect to and respective number of hops required to arrive to the destination. Each entry is tagged with a sequence number created by the destination node. In order to maintain stability from time to time, each node broadcasts and updates its routing table.

Reactive protocols focus on reducing overhead by determining routes on demand. The route discovery process, floods the network with RREQ (Route Request) packets, to map the path between source and destination whenever a node needs to communicate.

Some examples of proactive and reactive protocols:

OLSR is a proactive protocol, that can be seen as an optimization of the pure link-state algorithm[1], in a way that satisfies the requirements of the mobile paradigm. The key feature in this protocol, is that only some nodes forward the broadcast messages during a flooding. Furthermore these nodes, also known as multipoint relays (MRPs), are also the only ones that contain the link state information. This technique substantially reduces message overhead and number of control messages, when compared to the traditional flooding mechanism[TP03].

BATMAN is an optimization of OLSR, on its weak performance on large networks. In BATMAN, nodes do not maintain the full route to the destination, instead each node along the route only maintains the information about the next link, through which the best route can be found[JNA08].

DSR is a reactive protocol, that caches the complete hop-by-hop route between sender and receiver, when it floods the network. The packets carry the source route in the packet header, in order to dynamically discover unknown paths between nodes[SWS12].

Some hybrid protocols, such as SHARP[RHS03], use features from both types of protocols, in an adaptive way. Applying what type of protocol fits best on each situation.

2.1.4 Overlay Protocols

The logical topology of Peer-to-peer (P2P) networks is usually abstracted with some sort of overlay network. Overlays allow nodes to communicate to each other at the application level using logical overlay links that hide the physical network structure. This type of virtual network is used to support multiple distributed operations such as indexing and route discovery[Aky07].

The lack of a coordination point, requires that the nodes that make up the MANET communicate among the group rather than to a single centralized entity. So, the communication between nodes usually follows a multicast pattern, where one node sends a piece of information to one or more nodes. However, sometimes overlay protocols provide unicast (one sender, one receiver) support or resort to both communication types.

Several overlay multicast routing protocols have been proposed, each one optimized for efficiency and robustness. These can be categorized, by their dissemination approach, as:

Tree-based overlays adapt tree structures according to specific needs. The most basic scheme, arranges all the routes to form a tree infrastructure with the source node as root, thus creating one and only one path between every pair of sender and receiver. This scheme is very efficient due to resource optimization but introduces overhead induced by tree reconfiguration on node relocation. In a one-to-many on-demand media system such as oSTREAM [CLN04], the approach is to establish a

[1] In a pure link-state protocol, all the links with neighbor nodes are declared and are flooded in the entire network[Jac+01]

minimum spanning tree and use media buffering at the host to aid in the distribution of asynchronous service requests for the same streaming media. However, this strategy is not useful for many-to-many situations. In a many-to-many systems such as Yoid[Fra00], there is a single shared tree from all members from the designated group. Trees are managed by a concept of parent/child relationship between members. Each member divides the set of all other members into two groups called parent-side and child-side members. The groups are defined such that parent side members are all members reachable via the parent and all others are child-side members.

Mesh-based uses multiple redundant routes, that translates into a more robust routing system but wastes resources by forwarding unnecessary duplicate data across the multiple existing paths.

Stateless schemes avoid the overhead of maintaining a tree or a mesh infrastructure, such as the Differential Destination Multicast (DDM) protocol[Cor01]. In the DDM protocol, every data packet includes a header with all the addresses of the receivers.

Overlay multicasting is not made on the network layer. Instead, a virtual overlay network is built on top of the physical network. Links in the overlay are unicast tunnels over the physical layer. These characteristics simplify the multicast operation and supply a static network topology overview even though the underlying physical topology is changing[Su+09].

Routing protocols are not the only concern surrounding node communication, as they also depend on the connectivity of the network.

Some MANETs are fully connected[2], so the connection between two nodes is direct. However, in some cases it may require more than a single-hop of communication. When this occurs, it common to categorize the network as being multi-hop, where source host packets are relayed by several intermediate hosts being able to reach their destination.

In order to communicate or discover routes between two endpoints, nodes usually rely on some form of broadcasting[3].

The most simple broadcast approach is flooding, which is a dissemination mechanism that, every incoming packet is sent out on every outgoing line except the one it arrived on[Tan96]. Even though it generates vast numbers of duplicate packets, in many cases flooding is more efficient than maintaining a structure with the status of the data dissemination due to the high mobility of the nodes.

Broadcasting by flooding also presents some drawbacks, such as:

Redundant message overhead is caused by a large number of redundant message forwarding, particularly in a system with a high-connectivity topology. When multiple messages with the same message ID are sent to a peer by its multiple neighbors,

[2] Each of the nodes is connected to each other. In graph theory it is known as a complete graph

[3] Transferring a message to all recipients simultaneously

all, except for the first message, are considered as redundant messages. These redundant messages are pure overhead: they increase the network transfer and peer processing burden without enlarging the propagation scope[JGZ03].

Contention can be caused by the retransmission of messages by nodes that are in close proximity. These transmissions are all from nearby hosts, and therefore may contend with each other. This can lead to a full disruption of the system due to an emergent phenomena known as broadcast storm[Tse+02].

Collision of two packets that forces the hosts to resend them, and thus reduces the network efficiency, making it extremely complex to have nodes execute a distributed protocol among them[Tse+02].

The results presented in [Tse+02] provide an analysis on the above flooding deficiencies, and motivates the study of further improvements to overlay protocols[Su+09] and broadcasting algorithms[MLR06].

2.2 Peer-to-Peer Networks

In P2P systems, nodes (i.e individual processes that are part of the system) are typically equipotent participants. Each peer acts as both a server and client, providing and consuming resources in a mostly decentralized fashion.

P2P architectures can be categorized by the degree of dependency to a centralized infrastructure:

Fully decentralized networks are completely distributed and do not rely on a central component to offer their services. On this type of network, the exit of any participant should not have any meaningful impact on the provided services as the overall operation of the system.

Partially-decentralized networks have peers communicate to each other directly but use a pivotal infrastructure to provide or simplify the design of some of the network services[Sch01].

2.2.1 Overlay Networks

Overlay networks are classified accordingly to the method used to select and manage the links that are established between nodes (and consequently used to index available resources). These logical topologies can be of two types:

Structured overlay networks are organized according some pre-defined specific topology, in order to leverage the known information about the topology. Typically to improve the performance and cost of search and retrieve of resources.

The most common example of structured overlay are Distributed Hashtables (DHT), in which the ID of each node and object may consist of several dimensions (i.e. several IDs for each object). The hash keys are generated from descriptions of the content such as metadata or keywords.

In this type of topology, nodes are connected based on their identifier. Being that, they are linked to the nodes whose identifier are closest to their own, effectively forming an ordered ring (the identifier space is circular by definition).

On DHT networks, problems arise when there is high churn[4] rates that lead to high volume of traffic due to DHT maintenance. This factor leads to severe scalability and fault tolerance issues[Lei12; LR14] .

Unstructured overlay networks do not present any type of particular topology as a result of peers connecting to each other randomly.

The fact that these networks are not imposed with a pre-defined structure, makes them easy to build and optimize locally. Being unstructured makes all nodes equally privileged, producing an highly robust network to churn.

Unstructured overlay networks limitations are related to their support to perform query routing over the network, because there is no correlation between the local position of a peer in the network and the data maintained locally at that node. Peers usually resort to mechanisms based on message flooding over the network in order to find the resources (e.g particular data pieces). Unfortunately, the lack of an association between the peers and the shared content does not assure that required data is found, except if every node in the network processes the query.[Lv+02; LR14]

P2P networks gained popularity from Napster[Pat01], that inspired unstructured and structured systems like:

Gnutella is a decentralized protocol for performing distributed search, that uses an unstructured overlay model[Kir03]. In Gnutella, nodes called *servants* perform tasks usually associated both clients and servers. They provide client-side interfaces through which users can issue queries, view search results, accept queries from other servants, check for matches in their local data set, and respond with corresponding results. These nodes are also responsible for spreading the information that maintains the network integrity. In order to join the system, nodes initially connect to one of the known hosts, that are almost always available. Then it opens connections to one or more nodes already in the network, to be able to communicate with and through them[Rip01]. A revamped Gnutella protocol as been proposed [Sin01], in which nodes are categorized hierarchically. Nodes can be leaf-nodes or ultrapeers. Leaf-nodes only maintain one connection open, and that is to a Ultrapeer. Ultrapeers act as proxies to leaf-nodes connected to them. Reducing the

[4] Number of nodes exiting the network in a period of time[Sch01]

number of nodes involved in message handling and routing, improves scabality and diminishes traffic among nodes.

Freenet is a decentralized protocol for distributed data store, that also uses an unstructured model, to maintain and distribute information anonymously among network users. Freenet main focus is security and privacy[Fre15].

Pastry is a decentralized distributed object location and routing substrate for wide-area peer-to-peer applications, that resorts to a structured P2P model[RD01].

Chord is a decentralized distributed lookup protocol, that in order to solve the problem associated with data location on P2P networks, also uses a structured overlay protocol[Sto+01].

2.2.2 Fully decentralized VS Partially-decentralized

Choosing between fully decentralized as well as between structured and unstructured and semi-decentralized architectures is a matter of comparing the trade-offs of each one of these design choices and select those that fit the particular requirements of the system being designed.

Fully decentralized and unstructured overlay networks are easier to build and maintain, while also being robust to faults but provide no particular benefit when querying. Unlike structured networks, where search is efficient at the cost of sacrificing some robustness, and scalability.

Partially-decentralized architectures have the potential to have performance than fully decentralized architectures, due to the fact that certain operations, such as creating and maintaining an index to support search operations, can take advantage of the centralized infrastructures. Not only this approach can be efficient but it also benefits from the decentralized unstructured aggregation of nodes. However, using a central infrastructure introduces a dependency that goes against the purpose of achieving high scalability by avoiding potential bottlenecks in the architecture[YCM12].

In this work we plan to resort to a a partially-decentralized architecture, where we only resort to the centralized component when necessary, as to minimize the dependency of a potential contention point. This should offer the possibility to address some complex queries in a efficient way while still benefiting from the perks of decentralized networks.

2.2.3 Ad-hoc Querying

The infrastructureless nature of ad-hoc protocols, translates into a lack of resource indexation. So, when a host needs to find information, he does not know who are the other hosts that hold it. In order to obtain maximum network coverage, a protocol may resort to message broadcasting.

Many ad-hoc querying protocols have been proposed. The ones that do not assume nodes to be equipped with some sort of location hardware, make use of flooding, typically with a few optimizations. Despite many improvements that have been proposed to flooding protocols such as DSR[Joh+03] and ARA[GSB02], many messages are propagated unnecessarily (leading to resource consumption).

Gossip, or epidemic, protocols introduce a probabilistic variable that decides if a message should be forward or not, thus reducing the amount of transmitted messages.

Gossip protocols establish that when a node wants to broadcast a message, it selects f nodes at random and sends the message to them (f is a typical parameter named fanout). Upon receiving the message for the first time, each node proceeds to repeat this procedure[LPR07]. This is feasible as a result of the assumption that any node in the network can send a message to any other node, either because there is a direct link to that node or a route to that node is known. However, in the context of ad-hoc networks, this assumption is not realistic, due to the unique aspects of ad-hoc networks such as the mobility of nodes and the lack of a routing backbone. Nevertheless, this can be circumvented by taking advantage of a radio communications characteristic, the one hop broadcast message delivery. In wireless communications, messages are usually received by all the nodes that are in the close vinicity of the sender (one hop away). By managing this physical-layer broadcast feature, for instance by controlling the probability with which this broadcast is sent, gossip protocols can overcome this issue[HHL06].

This dissemination protocol (which can be used to build a somewhat inefficient routing mechanism), presents a bimodal behavior based on one branch of mathematics named *Percolation Theory*. Being bimodal means that in sufficiently large networks: in some executions the gossip process fails quickly and almost none of the nodes gets the message. However, in the remaining executions, a substantial fraction (close to 100%) of the network gets the message with high probability. The number of executions in which most of the nodes get the message, depends on the gossip probability and the network topology[HHL06].

Gossip protocols present themselves as highly scalable and resilient when implementing reliable broadcast. Studies [HHL06] have obtained results of up to 35% less messages in large networks, when compared with optimized flooding protocols. Even though it exhibits excessive message overhead in order to ensure reliability with high probability.

Some solutions for wired environments have explored gossip behavior to ensure robustness without consistency paying the price associated with the natural redundancy of gossip protocols.

Plumtree, or push-lazy-push multicast tree[LDT07], blends the tree-based broadcast primitives with gossip. By joining the reliability of a epidemic protocol and the small message complexity of tree-based protocols, and therefore covering the major faults of each other. Unfortunately, this protocol was not designed to wireless environments, and adopting it to cable ad-hoc networks is still an open challenge, which we will partially address as denoted in chapter 3.

2.3 Edge Computing

Edge computing is an extension to the Cloud Computing paradigm. Cloud Computing grants clients quick deployments, cost efficiency in terms of maintenance and upgrade, and easy access to data. However, latency-sensitive applications short delay requirements may not be satisfied as a result of servers being at considerable distance.

This novel architecture draws away computing and storage services from centralized infrastructures, to the logical extremes of a network. The edges provide location awareness and wide-spread geographical distribution. Consequently, by covering broad areas, nodes can be in the vicinity of others which translates into low latency to communicate and interact with such nodes.

Edge and Cloud can coexist and be used in conjunction, consider for an example the case where, edge nodes may collect data from sensors, and process it before sending it to the Cloud[Bon+12].

2.3.1 Mobile Edge Computing

Mobile Edge Computing (MEC), or fifth generation (5G) cellular wireless networks, results from the enriching of mobile network carrier radio base stations provided services to include cloud operations [Bec+14]. The 5G networks objective is to provide higher data rates, enhanced end-user quality-of-experience (QoE), reduced end-to-end latency, and lower energy consumption in response to the increasingly demanding mobile Internet requirements. Figure 2.1 portraits a simplified representation of the MEC network topology.

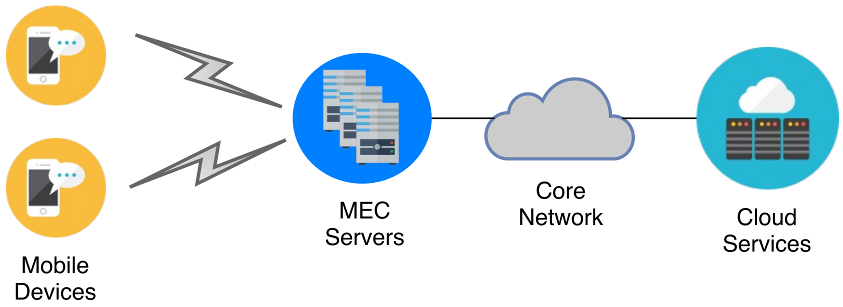

Figure 2.1: Mobile Edge Computing Architecture

In figure 2.1 there are three distinguishable entities:

Mobile devices such as smartphones and tablets, operated by regular mobile device users, in which network carrier subscribers can access cloud services within the range of the *Radio Access Network(RAN)*[5].

[5] collection of base stations, each making independent control plane decisions on the radio layer with some loose distributed coordination via mechanisms such as SON (self organizing networks)[Gud+13]

MEC servers owned and managed by network carriers, have the responsibility of managing traditional network traffic plus hosting and maintaining mobile edge applications such as *Cloud-RAN*. Cloud-RAN is a cloud service composed by radio access networks that resorts to the proximity of end users to provide better latency and throughput[Gud+13].

Cloud Services such as providing access to particular software applications over the network (Application Service Providing) or act as a proxy server to deliver content to users with high availability and performance (Content Distribution Network).

In MEC the goal of resource migration is reducing latency and bandwidth consumption. Being that in the mobile paradigm, by performing certain tasks closer to the client in latency sensitive applications such as real-time, communication or media streaming, network congestion is reduced and applications get better quality of service (QoS)[Bec+14].

Latency reduction is achieved by the proximity of the cloud services, as resources are made available by other clients, and thus being more likely to be geographically closer.

Bandwidth reduction is related to the local processing of large data before sending to the cloud[AA16].

MEC compares to Mobile Cloud Computing (MCC) due to the fact that both provide cloud services. However, by resorting to opposing architecture models[YLL15].

2.4 Security

Security is an important element of any distributed system, especially when these systems communicate through wireless networks. A secure distributed system must ensure a set of essential properties:

Availability: ensures that resources are accessible to authorized parties at expected times. This should happen despite malicious attacks to the network.

Authentication: ensures that the communication between two nodes is legitimate, in other words, each participant is genuine and not an impersonator.

Confidentiality: ensures that data is protected, in way that only the desired recipients can access the information.

Integrity: ensures that messages between two nodes are not modified by a third, not authorized, party.

Non-Repudiation: each peer must provide convincing evidence of the other's participation in a protocol session. If one peer falsely denies participating in a session, then the other peer can present his evidence to a judge, who can safely conclude that the other peer did participate. Crucially, the judge does not have to monitor the network traffic, but can make his judgement on the basis of the evidence alone[ZG96]

2.4.1 Security in MANETS

In the context of wireless networks, and in particular for MANETs, there are specific characteristics that significantly increase the challenges associated with enforcing these security properties.

Decentralization: Being infrastructureless requires nodes to have contributory collaborative roles in the network rather than ones of dependence. For instance, any security solution should rely on cooperative scheme instead of centralized one.

Wireless links: Non physical connections are more vulnerable than wired. Considering that attackers can come from every direction and target any node, wireless connections are more susceptible and harder to protect, in particular to personification attacks.

Multi-hop: In ad-hoc networks hosts also act as application level routers, and packets must pass through several, probably untrustworthy, nodes in their path between source and destination. As a result of this, malicious nodes may attempt to eavesdrop messages or modify their contents without being detected.

Resource limitation: Mobile nodes have limited battery supply and that may be used as a way to disconnect participants from the network. Not only battery, but also computing and storage limitations, restrict the use of complex security solutions such as cryptography.

Node mobility: The fact that several nodes exhibit, potentially fast paced, mobility patterns, complicates node tracking on a large network, and thus makes it more difficult to trace a malicious node[DB04].

2.4.2 Attacks

Attacks are composed by actions that intentionally aim at harming the network. These can be categorized by:

Origin

External attacks include attacks perpetrated by a node that does not belong to the logical network or is not allowed to access it.

Internal attacks include attacks launched by a malicious node that is actively part of the system[DB04].

Types

Passive attacks involve continuously collect of information, in order to use it for mounting of a future attack. The attacker does not disrupt the routing operation, it only eavesdrops packets, for picking up sensitive information or to deduce the network topology for routing information.

Active attacks refer to all remaining attacks that require active interaction with the system. However, these require the attacker to inject arbitrary packets into the network, and therefore increases the chances of detection[Lun00].

Attacks are not the only threats to ad-hoc networks, as internal nodes can (accidentally namely, due to hardware faults) deviate from the intended behavior and damage the the overall connection or performance of the system. For instance, when a node decides to not forward packets on behalf of other participants, in order to spare battery.

The physical network is not the only concern, being that overlay networks also have their vulnerabilities and may be subject to attacks themselves.

In particular, attacks to routing protocols can be classified as:

Modification attacks are those, in which malicious nodes cause redirection of the network traffic and DoS attacks by altering control message fields or by forwarding routing messages with modified values.

Spoofing attacks also known as impersonation attacks, occur when a node uses an identity that is not its own in the network, for instance by altering its MAC or IP address in outgoing packets.

Fabrication attacks correspond to the generation of false routing messages[DB04].

Sybil attacks happen when a peer participates in the network as multiple identities. By doing so, the attacker can gain large influence in the system, that can be used to disrupt communication in systems that rely on quorum based protocols or stealing information [Mon09].

2.4.3 Mobile Edge Network Security

The 5G network challenges arise from introducing IT applications into the telecom world. This integration imposes that these also obey the security requirements of the radio network.

Regarding physical security, there are also some challenges, due to the poor security conditions of radio base stations when compared to large data centers. Furthermore, issues can emerge when using third-party software applications, so these must come from trusted source, authenticated and authorized[Pat+14].

3

Proposed Work

This chapter describes and explains, the proposed solution and respective work plan to address the challenges being tackled on the context of the thesis. The premise of this solution, is that it is possible to use little infrastructure access to create a distributed gallery. By using an hybrid ad-hoc protocol, that focuses on peer-to-peer communication while still using an infrastructure as fallback, and to move some aspects of the system more efficient or effective.

3.1 Proposed Solution

The solution will be explained using a bottom-up approach, in other words, we depart from a solution close to the classical client-server architecture with no peer-to-peer interaction and move towards a solution that focuses on a peer-to-peer architecture while only resorting to the infrastructure as fallback. The intermediary solutions will be used as comparative baselines in the evaluation of our proposal.

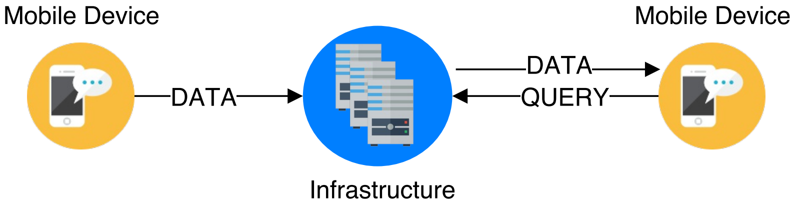

Figure 3.1: Solution 1 - Centralized

The diagram of the starter solution (figure 3.1) represents a client-server architecture using mobile devices. In this solution mobile devices do not communicate directly, and consequently rely on the infrastructure that stores and serves data between clients. This fully centralized solution has a bottleneck and single point of failure in the centralized component, which is the main motivation for the incorporation of peer-to-peer communication.

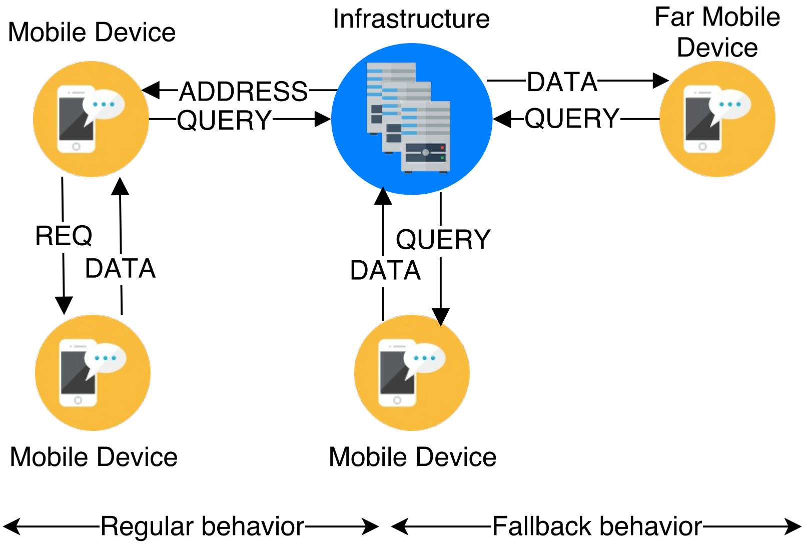

Figure 3.2: Solution 2 - Peer-to-peer with coordination infrastructure

In the second solution, depicted in figure 3.2, mobile devices communicate between themselves whenever possible, in a peer-to-peer fashion resorting to the infrastructure for coordination and resource indexing/querying. Infrastructures maintain an association between device and resources it possesses. So that, when a peer searches for a resource, it only has to contact the infrastructure, in order to obtain the address of the the device(s) that has it. After obtaining the address(es), it requests and transfers the data directly to those devices. However, the owner of the resource may not be directly reachable, in which case the infrastructure serves as a middle-man in the communication between peers.

This protocol, reduces latency and mitigates the bottleneck effect but it does not directly address, the single point of failure challenge.

The third and final solution (figure 3.3) organizes mobile devices into tree-based groups, based on their proximity. In this solution, devices try, whenever possible, to only query the peers in their group to minimize infrastructure dependency and consumption of resources in the centralized component. When this is not possible, for instance, when the searched resource is not on that group, peers select a leader that works as a gateway, through which queries to other groups are made. The leader receives the query from a group member, and retransmits that query to the infrastructure, so that it can query every leader of other groups for the desired resource. As seen on figure 3.3, the leftmost device

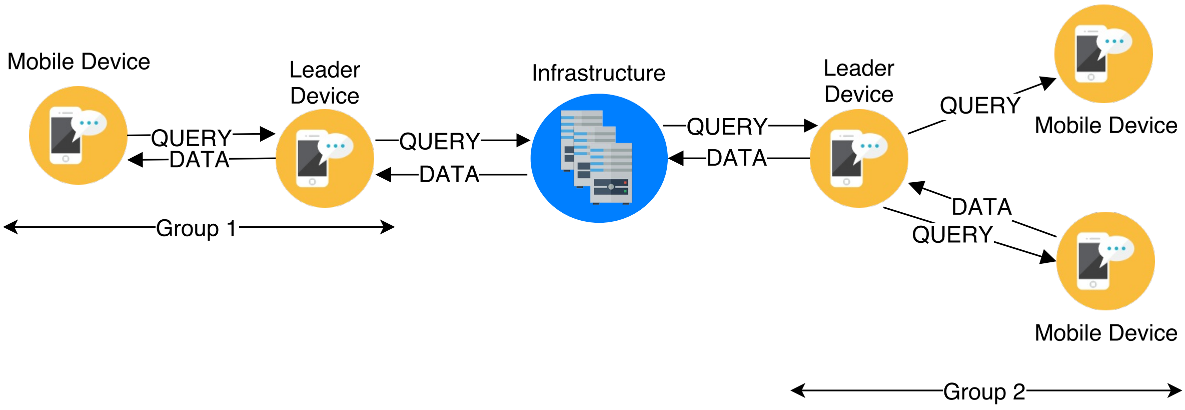

Figure 3.3: Solution 3 - Peer-to-peer with fallback infrastructure

in group 1 queries the leader of its own group, so that he can also query group 2 through the infrastructure. In turn the group 2 leader, floods the group network with the query.

The resource transfer follows the same protocol, when the desired data is found, the owner sends the resource to its group leader, that consecutively transmits the data to the infrastructure, that in turn sends to the leader of the other group, so that he delivers to the group member that executed the query (as depicted in figure 3.3, where the rightmost peer has to send data to the leftmost peer by resorting to both elements of both groups and the infrastructure).

3.1.1 Solution Evaluation

The solutions presented in section **??** will be evaluated in terms of latency, battery consumption, query precision and, message volume (between devices (peer-to-peer) and through infrastructure access). The latency and battery consumption results obtained in the experiments, may serve to provide enough evidence to ground a efficiency comparison between centralized and hybrid architectures, in this context. The analysis of message volume (in bytes) and the query precision, allows to evaluate the feasibility and performance of each solution implementation.

3.2 Work Plan

In this section the work plan for the elaboration of the dissertation is described. The plan is to design the solutions and take early feedback from their implementation to further improve their design. This way, there can be an adaptation of the design to the unexpected challenges that may appear. There will be an implementation and prototyping phase, divided in three steps (one step for each solution). This will be followed by solution evaluation and dissertation writing. Table 3.1 depicts the expected work dates, including duration.

Table 3.1: Work plan calendar

Task	Start Date	End Date	Days
Design	26 February	15 June	105
Implementation and Prototyping	2 March	15 June	105
Solution 1	2 March	6 April	35
Solution 2	6 April	11 May	35
Solution 3	11 May	15 June	35
Evaluation	15 June	20 July	35
Writing	10 July	23 September	75

Design of the system architecture. This task corresponds to the planning of the organization and interaction of the several components that compose each solution.

Implementation and Prototyping of each solution, following a bottom-up approach. This phase includes application development and testing.

Solution 1: classical mobile client-server architecture with no peer-to-peer interaction.

Solution 2: hybrid client-server architecture that resorts to the infrastructure for coordination while using peer-to-peer, when possible, for node interaction (data transfer).

Solution 3: hybrid client-server architecture that uses primarily peer-to-peer communication while still using infrastructure access as fallback mechanism.

Evaluation of the developed solutions in terms of latency, battery consumption, query precision and, message volume (in bytes) between devices (peer-to-peer) and through infrastructure access.

Writing of the dissertation, with the obtained results and drawn conclusions, and a paper that depicts the main contributions.

The figure 3.4 is a Gantt chart, that presents another perspective of how the dates in table 3.1 are organized and how they overlap.

3.3 Summary

In this document, alternatives that tackle media sharing and searching in mobile environments are presented and explained. More specifically, solutions that try to use minimal infrastructure access while resorting to ad-hoc communication between devices. As a case study, a distributed image gallery using an hybrid edge computing network as been proposed. This gallery can be searched using facial recognition techniques, that already have been developed in the project[Hyr15] scope that encompasses this work.

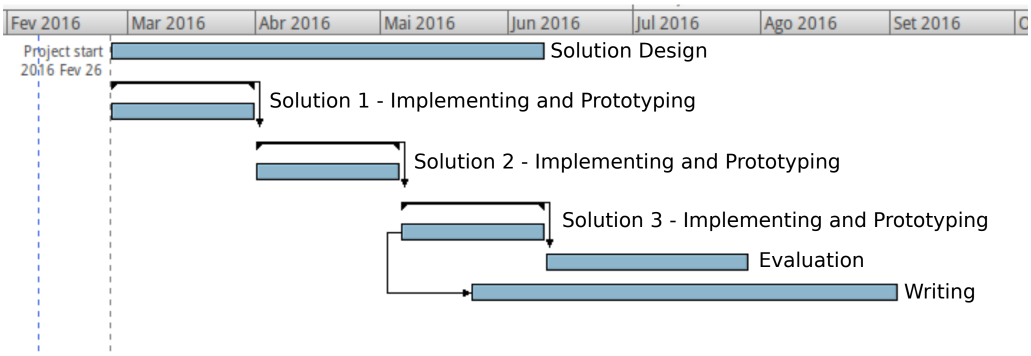

Figure 3.4: Proposed work schedule

Bibliography

[Ahe+07] S. Ahern, D. Eckles, N. S. Good, S. King, M. Naaman, and R. Nair. “Over-exposed?” In: *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07*. New York, New York, USA: ACM Press, Apr. 2007, p. 357. ISBN: 9781595935939. DOI: 10.1145/1240624.1240683. URL: http://dl.acm.org/citation.cfm?id=1240624.1240683.

[AA16] A. Ahmed and E. Ahmed. “A Survey on Mobile Edge Computing”. In: JANUARY (2016). DOI: 10.13140/RG.2.1.3254.7925.

[Aky07] I. F. Akyildiz. *NETWORKING 2007. Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet: 6th International IFIP-TC6 Networking Conference, Atlanta, GA, USA, May 14-18, 2007, Proceedings*. Springer Science & Business Media, 2007, p. 1252. ISBN: 3540726055. URL: https://books.google.com/books?id=r4V2G7yPLIAC%7B%5C&%7Dpgis=1.

[BKN05] R. A. Baratto, L. N. Kim, and J. Nieh. “THINC: a virtual display architecture for thin-client computing”. In: 39 (2005), pp. 277–290. URL: http://dl.acm.org/citation.cfm?id=1095837%20http://www.cs.unm.edu/%7B~%7Ddarnold/classes/papers/Baratto05THINC.pdf.

[Bec+14] M. T. Beck, M. Werner, S. Feld, and T. Schimper. “Mobile Edge Computing : A Taxonomy”. In: *Afin* c (2014), pp. 48–54.

[Bli16] Blizzard. 2016. URL: http://us.battle.net/d3/en/.

[Bon+12] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. “Fog Computing and Its Role in the Internet of Things”. In: *Proceedings of the first edition of the MCC workshop on Mobile cloud computing* (2012), pp. 13–16. ISSN: 978-1-4503-1519-7. DOI: 10.1145/2342509.2342513. URL: http://doi.acm.org/10.1145/2342509.2342513$%5Cbackslash$npapers2://publication/doi/10.1145/2342509.2342513.

[Cor01] M. Corson. "Differential destination multicast-a MANET multicast routing protocol for small groups". English. In: *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*. Vol. 2. IEEE, 2001, pp. 1192–1201. ISBN: 0-7803-7016-3. DOI: 10.1109/INFCOM.2001.916314. URL: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=916314.

[CLN04] Y. Cui, B. Li, and K. Nahrstedt. "oStream: asynchronous streaming multicast in application-layer overlay networks". In: *Selected Areas in Communications, IEEE Journal on* 22.1 (2004), pp. 91–106.

[DD08] O. Das and A. Das. "Performability evaluation of mobile client-server systems". In: *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08*. New York, New York, USA: ACM Press, Mar. 2008, p. 2197. ISBN: 9781595937537. DOI: 10.1145/1363686.1364210. URL: http://dl.acm.org/citation.cfm?id=1363686.1364210.

[DB04] D. DJENOURI and N. BADACHE. "A Survey on Security Issues in Mobile Ad hoc Networks Djamel". In: February (2004).

[FLR13] N. Fernando, S. W. Loke, and W. Rahayu. "Mobile cloud computing: A survey". In: *Future Generation Computer Systems* 29.1 (Jan. 2013), pp. 84–106. ISSN: 0167739X. DOI: 10.1016/j.future.2012.05.023. URL: http://www.sciencedirect.com/science/article/pii/S0167739X12001318.

[Fra00] P. Francis. *Yoid: Extending the internet multicast architecture*. 2000.

[Fre15] Freenet. *Freenet is free software which lets you anonymously share files, browse and publish "freesites" (web sites accessible only through Freenet) and chat on forums*. 2015. URL: https://freenetproject.org/.

[Gam10] S. Gammeter. "Server-side object recognition and client-side object tracking for mobile augmented reality". In: *Cvprw* C (2010), pp. 1–8. DOI: 10.1109/CVPRW.2010.5543248.

[Gud+13] A. Gudipati, D. Perry, L. E. Li, and S. Katti. "SoftRAN: Software Defined Radio Access Network". In: *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*. HotSDN '13. Hong Kong, China: ACM, 2013, pp. 25–30. ISBN: 978-1-4503-2178-5. DOI: 10.1145/2491185.2491207. URL: http://doi.acm.org/10.1145/2491185.2491207.

[GSB02] M. Günes, U. Sorges, and I. Bouazizi. "ARA-the ant-colony based routing algorithm for MANETs". In: *Parallel Processing Workshops, 2002. Proceedings. International Conference on*. IEEE. 2002, pp. 79–85.

[HHL06] Z. Haas, J. Halpern, and L. Li. "Gossip-based ad hoc routing". In: *IEEE/ACM Transactions on Networking (ToN . . .* (2006), pp. 1–12. ISSN: 1063-6692. DOI: 10.1109/TNET.2006.876186. arXiv: 0209011v1 [arXiv:cs]. URL: http://dl.acm.org/citation.cfm?id=1143399.

[Hyr15] Hyrax. 2015. URL: http://hyrax.dcc.fc.up.pt/.

[Jac+01] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. "Optimized link state routing protocol for ad hoc networks". In: *Ieee Inmic 2001: Ieee International Multi Topic Conference 2001, Proceedings: Technology for the 21St Century* (2001), pp. 62–68. DOI: 10.1109/INMIC.2001.995315. URL: http://apps.webofknowledge.com/full%7B%5C_%7Drecord.do?product=UA%7B%5C&%7Dsearch%7B%5C_%7Dmode=GeneralSearch%7B%5C&%7Dqid=2%7B%5C&%7DSID=1A5x8KrqG7PGKaNLOLm%7B%5C&%7Dpage=1%7B%5C&%7Ddoc=2.

[JGZ03] S. Jiang, L. Guo, and X. Zhang. "LightFlood: An efficient flooding scheme for file search in unstructured peer-to-peer systems". In: *Proceedings of the International Conference on Parallel Processing* 2003-January.5 (2003), pp. 627–635. ISSN: 01903918. DOI: 10.1109/ICPP.2003.1240631.

[JHE99] J. Jing, A. S. Helal, and A. Elmagarmid. "Client-server computing in mobile environments". In: *ACM Computing Surveys* 31.2 (1999), pp. 117–157. ISSN: 03600300. DOI: 10.1145/319806.319814.

[JNA08] D. Johnson, N. Ntlatlapa, and C. Aichele. *Simple pragmatic approach to mesh routing using BATMAN*. en. Oct. 2008. URL: http://researchspace.csir.co.za/dspace/handle/10204/3035.

[Joh+03] D. B. Johnson, D. A. Maltz, Y.-C. Hu, and J. Jetcheva. "The dynamic source routing (DSR) protocol for mobile ad hoc networks". In: *IETF Draft, draft-ietf-manet-dsr-009. txt* (2003).

[Kir03] P. Kirk. *Gnutella - A protocol for a Revolution*. 2003. URL: http://rfc-gnutella.sourceforge.net/developer/stable/index.html.

[Lei12] J. Leitão. "Topology Management for Unstructured Overlay Networks". PhD thesis. 2012.

[LR14] J. C. A. Leitão and L. E. T. Rodrigues. "Overnesia: a resilient overlay network for virtual super-peers". In: *Reliable Distributed Systems (SRDS), 2014 IEEE 33rd International Symposium on*. IEEE. 2014, pp. 281–290.

[LPR07] J. Leitão, J. Pereira, and L. Rodrigues. "HyParView: A membership protocol for reliable gossip-based broadcast". In: *Proceedings of the International Conference on Dependable Systems and Networks* (2007), pp. 419–428. DOI: 10.1109/DSN.2007.56.

[LDT07] P. Lu, R. Di, and F. Tr. "Epidemic Broadcast Trees". In: (2007).

[Lun00] J. Lundberg. “Routing security in ad hoc networks”. In: *Helsinki University of Technology* (2000), pp. 1–12. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.144.7589.

[Lv+02] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. “Search and replication in unstructured peer-to-peer networks”. In: *ACM SIGMETRICS Performance Evaluation Review* 30.1 (2002), p. 258. ISSN: 01635999. DOI: 10.1145/511399.511369.

[Mah12] A. Maheshwari. “Hybrid Approach of Client-Server Model and Mobile Agent Technology to Drive an E-Commerce Application”. In: 2.4 (2012), pp. 733–738.

[Mar09] E. E. Marinelli. “Hyrax : Cloud Computing on Mobile Devices using MapReduce”. In: *Science* 0389.September (2009), pp. 1–123.

[Mic16] Microsoft. 2016. URL: https://www.office.com/.

[MLR06] H. Miranda, S. Leggio, and K. Raatikainen. “A power-aware broadcasting algorithm”. In: *The 17th Annual IEEE International Symposium on Personal, Indoor andMobile Radio Communications (PIMRC’06)* (2006).

[Mon09] D. Monica. “Thwarting the sybil attack in wireless ad hoc networks”. In: *Master’s Thesis at the Universidade Tecninca de Lisboa* (2009).

[Pat+14] H. M. Patel, Y. Hu, P. Hédé, I. B. M. J. Joubert, C. Thornton, B. Naughton, I. Julian, R. Ramos, C. Chan, V. Young, S. J. Tan, and D. Lynch. “Mobile-Edge Computing”. In: 1 (2014), pp. 1–36.

[Pat01] L. Patrick. 2001. URL: http://www.napster.com/.

[RHS03] V. Ramasubramanian, Z. J. Haas, and E. G. Sirer. “Sharp”. In: *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing - MobiHoc ’03* (2003), p. 303. DOI: 10.1145/778415.778450. URL: http://dl.acm.org/citation.cfm?id=778415.778450.

[Rip01] M. Ripeanu. “Peer-to-peer architecture case study: Gnutella network”. In: *Peer-to-Peer Computing, 2001. Proceedings. First International Conference on*. IEEE. 2001, pp. 99–100.

[RD01] A. Rowstron and P. Druschel. “Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems”. In: *Middleware 2001* 2218.November 2001 (2001), pp. 329–350. ISSN: 03029743. DOI: 10.1007/3-540-45518-3. URL: http://www.springerlink.com/index/10.1007/3-540-45518-3.

[Sch01] R. Schollmeier. “A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications”. In: *Proceedings First International Conference on Peer-to-Peer Computing* (2001), pp. 2–3. ISSN: 1479-5876. DOI: 10.1109/P2P.2001.990434.

[SWS12] B. D. Shivahare, C. Wahi, and S. Shivhare. “Comparison Of Proactive And Reactive Routing Protocols In Mobile Adhoc Network Using Routing Protocol Property :” in: *International Journal of Emerging Technology and Advanced Engineering* 2.3 (2012), pp. 356–359.

[Sin01] A. Singla. 2001. URL: http://rfc-gnutella.sourceforge.net/Proposals/Ultrapeer/Ultrapeers.htm.

[Sto+01] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications”. In: *Conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '01)* (2001), pp. 149–160. ISSN: 01464833. DOI: 10.1145/383059.383071. URL: http://portal.acm.org/citation.cfm?doid=383059.383071.

[Su+09] B. Su, H. Yu, Z. Ma, C. Yang, and Y. Zhu. “Research on Overlay Multicast Routing Protocols for Mobile Ad Hoc Networks”. In: *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing* (2009), pp. 1–4. DOI: 10.1109/WICOM.2009.5302328. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5302328.

[TP03] E. T. Clausen and E. P. Jacquet. “Optimized Link State Routing Protocol (OLSR)”. In: (2003). ISSN: 2070-1721. URL: http://www.rfc-editor.org/info/rfc3626.

[Tan96] A. S. Tanenbaum. *Computer Networks*. Vol. 52. 169. 1996, pp. 349–351. ISBN: 0130661023. DOI: 10.1016/j.comnet.2008.04.002. URL: http://www.ietf.org/rfc/rfc169.txt.

[Tse+02] Y.-c. Tseng, S.-y. Ni, Y.-s. Chen, and J.-p. Sheu. “The Broadcast Storm Problem in a Mobile Ad Hoc Network”. In: (2002), pp. 153–167.

[YLL15] S. Yi, C. Li, and Q. Li. “A Survey of Fog Computing”. In: *Proceedings of the 2015 Workshop on Mobile Big Data - Mobidata '15*. New York, New York, USA: ACM Press, June 2015, pp. 37–42. ISBN: 9781450335249. DOI: 10.1145/2757384.2757397. URL: http://dl.acm.org/citation.cfm?id=2757384.2757397.

[YCM12] X. Yong, D. Chi, and G. Min. “The Topology of P2P Network”. In: *Journal of Emerging Trends in Computing and Information Sciences* 3.8 (2012), pp. 1213–1218.

[ZG96] J. Zhou and D. Gollman. *A fair non-repudiation protocol*. IEEE, 1996.