

Collaborative Cognitive Content Dissemination and Query in Heterogeneous Mobile Opportunistic Networks

Milena Radenkovic
University of Nottingham
Nottingham NG8 1BB
United Kingdom
milena.radenkovic@nottingham.ac.uk

Vu San Ha Huynh
University of Nottingham
Nottingham NG8 1BB
United Kingdom
psxvsh@nottingham.ac.uk

ABSTRACT

This paper investigates complex challenges of opportunistic discovery of content stored in remote mobile devices and delivery to the requesting nodes in heterogeneous mobile disconnection prone environments. We propose new latency aware collaborative cognitive caching approach suitable for content dissemination and query in heterogeneous opportunistic mobile networks and dynamic workloads. Utilising fully localised and ego networks multi-layer predictive heuristics about dynamically changing topology, dynamic resources and varying popularity content, our cognitive caching achieves high success ratio, low delays and high caching efficiency for very different real world dynamically changing mobile topologies.

KEYWORDS

Opportunistic networks, caching, latency awareness, congestion awareness.

1 INTRODUCTION

We live in the world where smart, ubiquitous devices are embedded in our day-to-day activities and allow us to form diverse communities in which we are able to share rich and complex data. Increasing number of applications and services are being hosted in the mobile edges at a relatively low cost and are becoming significant content providers [4]. Traffic demands for these edge-hosted providers are increasingly challenging as the published data is massive (i.e. the majority of Internet traffic is now dominated by streamed video content [7]) and can lead to surges of traffic if the content becomes popular. While large commercial providers use data centers to hold their data and user population across a very large geo area, and address traffic surges by delegating content distribution to servers located as close as possible to the users [7], we address scenarios which have significantly higher topology dynamics (including potential disconnections) and more dynamic workload patterns (unknown

publishers and subscribers distributions). Our aim is to maximise matching of cached content and queries while minimizing latency and avoiding network congestion. In this paper we propose design for collaborative cognitive caching approach that can predict and adapt to local dynamically changing topologies, dynamic workloads imposed by varying publishers and subscribers, dynamic content popularity while being aware of the storage and network resources as well as delay. At the heart of our approach is distributed *edge based collaborative caching* which consists of several multidimensional predictive analytics that build multi-attribute complementary predictive heuristics and utilities. We address two complex open questions about decisions making in distributed opportunistic caching: *where* to cache and *what* to cache. We use principles of dynamic predictive relative utilities and propose a collaborative algorithm which allows individual nodes to achieve greater utility than if they do not collaborate. Previous research has shown that collaborative caching usually outperforms locally optimized algorithms [13]. Note that our focus is not to build a protocol that forces nodes to collaborate, or provides protection against malicious behaviour but, rather, to design underlying algorithms that can *adaptively* share distributed cache space across trusted collaborators.

We extend the idea of behavioural locality to exploit *similarities* between the *content interests* and *users connectivity*. We expand the idea of *content popularity* with *popularity stability* in order to minimise negative impact of flash crowds (e.g. caused by “fake news”). We tackle the challenge of maximizing the number of data chunks accessible with as low delay as possible even in sparse fragmented topologies. We propose that edge nodes use CafRepCache algorithm to form *dynamic transient interest-based* and *data dissemination topologies* based on predictive analysis and commonalities between their interest, cache and retrieval histories as well as connectivity histories. This provides each node with a set of overlay neighbours whose browsing history most

closely resemble their own. CafRepCache emerges from this topology as the federation of the local caches of a node's ego network and the closest available nodes. We argue that careful management of both replication and caching is necessary to address dynamic fragmented and sparse topologies.

The paper begins by providing an overview of the related work in section II, section III introduces CafRepCache model and describes its heuristics and pseudo-code, section IV evaluates performance of CafRepCache against two competing approaches (SocialCache and SmartCache) across a range of metrics over three heterogeneous realistic mobile social and vehicular traces for dynamically varying workloads and content popularity. Section V gives conclusion.

2 RELATED WORK

Authors in [6] combine betweenness, similarity and tie strength for social routing metric which directs the traffic to more central nodes, and thus increases the probability of finding optimal relay for delivering packets. However, increasing traffic demands will inevitably aggregate at and congest the points that have higher social centrality. [2] propose Café and CafeREP, a congestion aware mobile social framework for data forwarding over heterogeneous opportunistic networks. [9] propose Behave peer-to-peer cache-oriented approach for Web applications that relies on the principles of Behavioural Locality inspired by collaborative filtering but it does not support dynamic mobile user topologies which we do in this paper.

[7] proposes a regional caching approach of video content that takes into account content global popularity as well as regional tastes as well. The authors propose a model that captures the overlap between inter-regional and intra-regional preferences. In [8], authors propose fully distributed reputation mechanism for next generation mobile offloading in resource constrained mobile networks. [3,15] describes integration of energy efficient location aware dynamic DHTs over dynamic and mobile topologies that provide service to in-situ and remote queries. In our paper we build on this but consider more complex topologies and query patterns to combine it with complementary decentralized multi-dimensional cognitive caching to reduce retrieval times drastically.

3 COLLABORATIVE COGNITIVE CACHING

3.1. Model

In game theory, each node attempts to optimise its personal "utility". We model our collaborative cognitive

caching as a bargaining game inspired by heuristic FairCache algorithm [1] and extend it to address real world challenges about the lack of support for dynamic demand matrix, dynamic node availability and congestion identified in [1,7,8,9]. We do this by enabling *responsiveness to dynamically changing network topology, congestion avoidance and varying patterns of content publishers/subscribers* while allowing low latency content retrieval, high cache efficiency and efficient use of resources. Our cognitive caching utility aims to serve subscribers with the lowest possible delay and without saturating available resources, thus either using its local cache, or by redirecting a request to a nearby collaborative cache, rather than forwarding to the original source. We model the network G that consists of a set N of nodes and a set E of edges, $G = (N, E)$. As the connectivity of the network and the state of the nodes change over time, we model each of these as time series, thus $N = \{N_t : t \in T\}$ and $E = \{E_t : t \in T\}$. The objective function for our cognitive cache utility can be modelled as follow:

$$\max \prod_{n \in N} (U_n - \frac{1}{U'_n})$$

where U_n is n 's utility value and U'_n is n 's social and resource utility for forwarding decision.

Each node $n \in N$ measures its Betweenness centrality, Similarity and Tie Strength in order to route content (request) to nodes with higher centrality. We model the congesting rate and node's availability as follow: the demand for node n at time t is given by $D_n^t = \sum_{i \neq n \in N} \sum_{j \neq n \in N} F_{ij}^t(n)$ in which $F_{ij}^t(n)$ denotes the number of paths between i and j that contains n at time t . Each node $n \in N$ in the network can have a different stress level at any given time t : $ST_n^t = \frac{D_n^t}{C_n}$ in which C_n is the node's capacity. $ST_n^t > 1$ implies that the packet loss occurs. We measure delay at node n relatively as the level of congestion experienced by n at a given time t : $delay^t(n) = \frac{D_n^t + \sum_{b \in B} b}{d^t(n)}$, implies that level of congestion experienced is measured by demand D and buffered demand B over the number of available outlets d . We denote Retentiveness $Ret(n)^t = B_c(n) - \sum_{i=1}^M m_{size}^i(n)$ as the node's available storage at time t , measured by sum of all message occupancy subtracted from the node's buffer capacity. We also depict Receptiveness $Rec(n)^t = \sum_{i=1}^M (T_{now} - m_{received}^i(n))$ as the delay node n adds to messages in order to forward them, measured by the sum of differences between the current time and the time each message was received.

We assume that each node in the network $n \in N$ has a cache of size C_n . We denote with O a set of content objects that can be requested by the network. Each content $k \in O$ has the object size s_k . At each node $n \in N$, $q_{n,k}$ is the normalised request rate of the content k (i.e. content popularity) observed locally from n , $\sum_k q_{n,k} = 1$; $w_{n,k}$ is the normalised aggregated request rate of the content k observed from all the neighbours of n , $\sum_k w_{n,k} = 1$. We define the Ego Network of each node n : $EN_n = \{i \mid l_{n,i}^t \leq r_n^t, \forall t \in T, \forall i \in N, i \neq n\}$ in which r_n defines the radius measured by hop (e.g. $r_n = 1$ means direct encounter) and $l_{n,i}$ defines the distance between n and its contact i . Ego network (EN) is defined here as a network consisting of a single node together with the nodes they have encountered and gives each node their own perspective of the network.

To model CafRepCache strategy, we denote $x_{n,k} \in \{0,1\}$ as whether to cache content k at node n , $y_{n,i,k} \in \{0,1\}$ as whether to forward k to neighbour i of n , $z_{n,i,k} \in \{0,1\}$ as whether to replicate k to i and $dr_{n,k} \in \{0,1\}$ as whether to drop k at node n . Thus, the utility value of node n can be described as:

$$U_n = \sum_{n \in N} \sum_{k \in O} \sum_{i \in EN} U(q_{n,k} w_{n,k}) Q(x_{n,k}, y_{n,i,k}, z_{n,i,k}, dr_{n,k}) S_k \frac{1}{T_k}$$

where $U(q_{n,k} w_{n,k})$ weights the value of collaborative observation over local observations, S_k is the success ratio of delivering content k and T_k is the latency to retrieve k .

3.2 Distributed Analytics and Heuristics

Our work improves reliability and latency awareness of the current state of data dissemination and query over mobile edge networks to enable the following: i) local latency awareness; ii) real-time predictive adaptive response to changing local conditions; iii) support varying workloads, vi) support for dynamically changing content popularity and v) support both dense and sparse topologies as more realistic.

We propose multiple mobile edge predictive analytics heuristics which leverage information on the local available resources, connectivity patterns and mobility of publishers/subscribers, and dynamic content popularity (Figure 1). Our multi-layer approach is key to enabling dynamic trade-off between minimising the end-to-end latency and maximising content delivery while enabling congestion avoidance. Note that we assume that any node can start to publish content at any time and from any location while being mobile (e.g. high quality video). We first describe our analytics, heuristics and utilities and

then give overview of how the content gets published and how the subscribers can register their interests and retrieve the content in mobile dynamic complex sparse topologies.

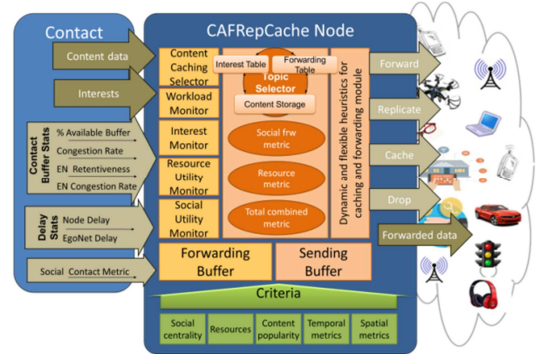


Figure 1. CafRepCache node architecture

Our *social driven analytics* enables forwarding packets to higher centrality nodes (compared to themselves) by applying SimBetTS[6] utility metrics defined below.

- Betweenness:

$$Bet = \sum_{i=1}^N \sum_{j=1}^{i-1} \frac{g_{ij}(n)}{g_{ij}} \quad (1), g_{ij}(n) \text{ is the number of paths linking } i \text{ and } j \text{ that includes } n.$$

- Similarity:

$$Sim(n, m) = |N_n \cap N_m| \text{ is the similarity of contact between two nodes}$$

- Tie strength:

$$TS(n, m) = \frac{f(m)}{F(n) - f(m)} + \frac{rec(m)}{T(n) - rec(m)} + \frac{d(m)}{D(n) - d(m)}$$

combined both frequency and regency of contacts between the nodes.

Social utility of node n compared to node m for delivering a packet to destination d is:

$$SU = \frac{\alpha * Bet(n)}{Bet(n) + Bet(m)} + \frac{\beta * Sim(n, d)}{Sim(n, d) + Sim(m, d)} + \frac{\gamma * TS(n, d)}{TS(n, d) + TS(m, d)}$$

$$\alpha + \beta + \gamma = 1$$

Our *node-resource and delay driven analytics* includes retentiveness - defined as percentage of remaining storage capacity in a node, receptiveness - defined as predictive delay observed by the node, congesting rate - defined as indication of how fast the buffer is to get filled up, as below.

-Retentiveness:

$$Ret(n) = \mu Ret_{old}(n) + (1 - \mu) Ret_{current}(n)$$

-Receptiveness:

$$Rec(n) = \mu Rec_{old}(n) + (1 - \mu) Rec_{current}(n)$$

- Congesting Rate of a node n gives local prediction of how fast the node is likely to congest:

$$CR(n) = \frac{100 * \frac{T_{fullbuffer}(l)}{T_{totaltime}(l)}}{\frac{1}{N} \sum_i T_{end}(i, n) - T_{start}(i, n)}$$

Congesting rate is calculated as the percentage of time a node or region has been congested divided by the average time between congestion periods for the node or region.

Ego-network driven predictive analytics allows collaboration with asynchronous neighbours. Specifically, ego-network metric for heuristic h of node n is calculated as an average of the heuristic values of all the encountered nodes for node n .

$$EN_h(n) = \frac{1}{N} \sum_{i \neq n \in N} h(i), h \in \{Social, Ret, Rec, CR\}$$

For each of the heuristics h , we define their respective utilities $Util_h(n, C_i(n))$ as measurements of their relative gain, loss or equality, calculated as pair-wise comparison between the node's own heuristics and that of the encountered contacts in the following way:

$$Util_h(n, C_i(n)) = \frac{h(n)}{h(n) + h(C_i(n))}$$

We combine the above relative utility driven heuristics in order to allow highly adaptive forwarding by directing traffic to more central nodes while detecting and reacting to network congestion caused by increased topic popularity as below.

$$TotalUtil(n, C_i(n)) = \frac{1}{|H|} * \sum_{h \in H} Util_h(n, C_i(n))$$

$$h \in H = \{Social, Ret, Rec, CR, EN_{SoC}, EN_{Ret}, EN_{Rec}, EN_{CR}\}$$

Our content popularity analytics is defined as:

$$P(Ti) = \frac{ObservedTimePeriod(Ti) *}{Total - time * Betweeness(Ti) * Recency(Ti)}$$

$P(Ti)$ measures probability caching decision over a certain period (i.e. temporal locality) in which P is the weight that identify the content popularity. $Betweeness(Ti)$ is the temporal function that measures the time gap between continuous requests and $Recency(Ti)$ denotes the most recent interest request. $P(Ti)$ aims to provide tradeoff between current observed content popularity versus long terms interest in it in order to balance between potentially fake news and long term useful content. When a caching node detects it is likely to start congesting, it ranks the content in terms of its popularity and delegates the least popular content to a suitable node. Nodes suitability is ranked in terms of the same multi-criteria metric we described (social, resources and workload).

Our total combined utility does not achieve the global optimum. [14] have shown that attaining a global optimum often disadvantages some parties e.g. nodes may be unfairly exploited by other caches redirects (at the cost of their own performance): Contrary to this we enable high performance efficiency of individual caches while avoiding draining the resources of other nodes and decreasing their performances.

Each node has a unique ID and every routed message has an associated key and state information which may contain content topics and content data, publishers IDs, subscribers IDs, timestamps, location, IDs of other encountered nodes, times stamps of these meetings etc. Contents are tagged with a set of attributes which are hashed and stored in DHT-like overlay that effectively matches the hash value of interest with attributes representing the content. When interest packet reaches the nearest cached content or the publisher, the node forwards the actual content data back to the subscriber using CafRepCache forwarding scheme. During content retrieval process, using interest forwarding table, relay node matches the content topic and summary vector of the subscriber with the information it has about the published content, and forwards it to the subscriber. Along with forwarding the content or queries to next hops that have high social centrality and resources, intermediate nodes decide whether to cache the content, forward it or delegate it in case of resources limitations. We provide CafRepCache pseudo code below:

```

List CafRepCacheUtils = {}
For each Contact in scan do:
    Contact.NodeHeuristics = exchNodeHeurInfo(Contact)
    Contact.CalculateNodeUtilities(Contact.NodeHeuristics)
    Contact.CafRepCacheHeuristics = exchCafRepCacheHeurInfo(Contact)
    Contact.CalculateCafRepCacheUtility(Contact.CafRepCacheHeuristics)
    CafRepCacheUtils.Insert(Contact)
    Contact.TopicPopularity = exchTopicPopularityInfo(Contact)
    Contact.CalculateTopicPopularity(Contact.TopicPopularity)
End For
List CacheBuffer = {}
For each Contact in reverse sort CafRepCacheUtils do:
    If recv = Contact.recvPublisherTopic then:
        CacheBuffer.append(recv)
    End If
    Contact.SendingRate = CafRepCache.ComputeSendingRate(Contact)
    Contact.ReplRate = CafRepCache.ComputeReplRate(Contact)
    Vector ContactSTL = Contact.obtainSubscribersTopicList(Contact)
    Buffer SendBuffer = {}
    For each Topic in reverse sort CacheBuffer do:
        While Node.isCongested() do:
            Node.delegate(Topic, Contact)
        While End
        If exists ContactSTL(Topic) then:
            Vector ContactTopicSV =
                Contact.obtainSummaryVector(Contact, Topic)
            For Chunk in Node.Topic.SV complement ContactTopicSV do:
                Chunk.ReplCount = Repl
                ReplicaBuffer.append(Chunk)
            End For
        End If
    End For
    ChunksSent = Contact.Replicate(ReplicaBuffer)
End For

```

4 EVALUATION

We provide systematic analysis of three dynamic collaborative in network caching strategies depending on

how they choose caching locations: 1) preferring nodes with high *social utility* (i.e. betweenness and similarity centrality) to enable congruency with underlying mobile topology and caching closer to the mobile subscribers, 2) preferring nodes with high *social and resource utility* to allow caching close to the dynamic mobile subscribers while being congestion-aware and avoiding it and 3) *social and congestion aware caching with replication* to enable optimal caching in the face of dynamic mobile topologies which are prone to fragmentation. For each strategy, we assume that nodes with highest respective utility metrics were preferred when choosing caching locations and that the cached content was then retrieved by the increasing number of random subscribers using the same forwarding algorithm. As mobility and connectivity of the nodes have a major impact on the performance of any opportunistic communication protocols, it is fundamental to evaluate CafRepCache over multiple heterogeneous real world mobile data sets. Thus, we use three traces in ONE [5] which vary highly in terms of periods of disconnections, periods of connectivity and islands of connectivity [3]: Infocom 2006 [10], RollerNet [11] and San Francisco [12] described below. Infocom trace [10] is a 4 day trace that consists of 78 volunteers equipped with Bluetooth devices and additional 20 static long-range devices placed at various semi-static and static locations of the conference venue. RollerNet [11] trace spans three hours during which 62 roller-bladers travel about 20 miles in Paris and

utilize Bluetooth on their cell phones for communication. San Francisco Cab Trace [12] are GPS traces of 550 cabs over a period of 30 days in the San Francisco Bay Area. Figure 2a shows that SocialCache success ratio in Infocom starts off with 60% for low to medium percentage of subscribers but then decreases to about 40% in the face of increasing congestion and workload. SmartCache keeps success ratio around 80% while CafRepCache increases it from 80% to above 90%. This is due to CafRepCache profiting from both adaptive caching and smart *partial* replication. For RollerNet trace (Figure 2b) success ratios are high for both SmartCache (80%) and CafRepCache (90%), while SocialCache manages to keep around 60%. For San Francisco trace (Figure 2c), CafRepCache shows significant improvements (ranging from 70 to 80%) compared to SocialCache (48% to 38% and SmartCache (48% to 58%) as it can predict and cope with network fragmentations more efficiently. In Figure 3, we observe that SocialCache shows highest delays of answered queries across all three traces and they are increasing with the increasing load and congestion. SmartCache shows delay improvements for each trace with a slight increase for increasing workloads. Only CafRepCache keeps significantly lower delays for all traces and also decreases delays in the presence of increasing number of subscribers and content popularity.

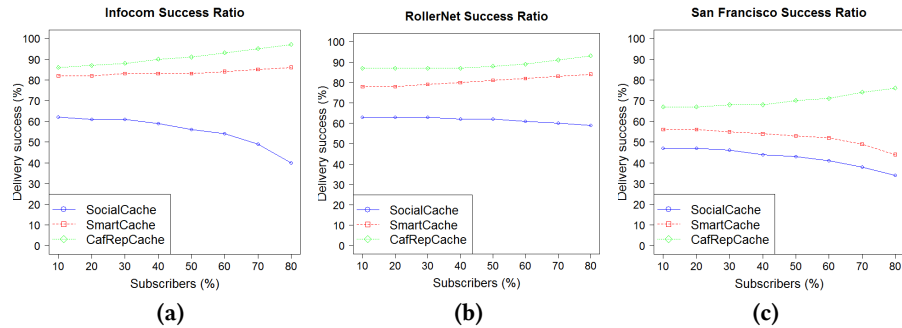


Figure 2. Success ratio for Infocom, RollerNet and San Francisco traces for increasing content popularity

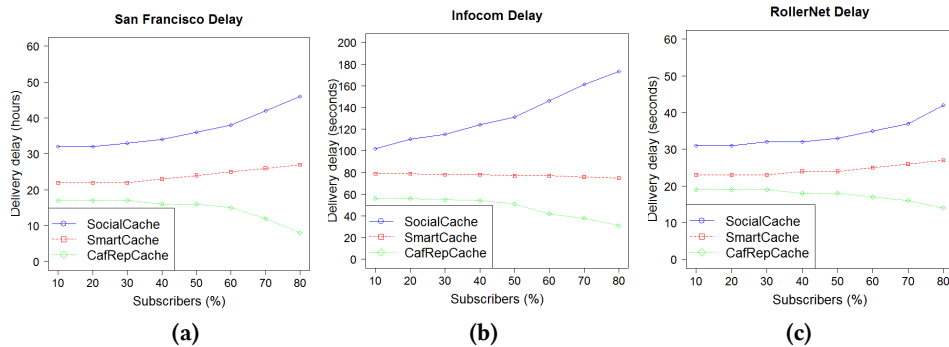


Figure 3. Retrieval delays for Infocom, RollerNet and San Francisco traces for increasing content popularity

SocialCache heuristics perform well because they allow congruency with distributed mobile data queries and dynamic interactions while depicting dynamics of the underlying topology (all three topologies have social character [3] with social metrics being applicable). SmartCache is more successful compared to SocialCache as it performs in network predictive resource analytics and rebalances the caching nodes locations so that it avoids congestion and delays, while keeping high social metrics to drive caching closer to the subscribers. CafRepCache is most successful as it includes both social and resource metrics, but when the caching node predicts that it is likely to get congested, but it delegates caching of the *least popular* content from its local cache to another node that has most appropriate contact it meets.

We evaluate efficiency of CafeRepCache individual caches in terms of how much of the cached content they store is delivered to the subscribers and show in Table 1 that it is 90% (Infocom), 70% (San Francisco) and 84% (RollerNet). Individual partial replication efficiency is very high 93% (Infocom), 68% (San Francisco) and 67% (RollerNet) (Table 1). This shows that our collaborative and adaptive cognitive caching manages to select highly suitable locations for caching and replication as well as suitable content chunks to cache and replicate when needed.

Table 1: Location Efficiency

Location Efficiency	Infocom 2006	San Francisco	RollerNet
Replication	90%	70%	84%
Caching	93%	68%	87%

5 CONCLUSIONS

We showed that multi-path content and interest forwarding with adaptive collaborative cognitive caching and replication can make drastic performance improvements for data sharing in complex temporal fragmented mobile topologies. Our results show that our cognitive collaborative caching manages to maintain high success ratio of answered queries, high efficiency of caches and short download times in the face of heterogeneous topologies, dynamic resources and increasing topic popularity. In case of hostile nodes and fake news, we plan to investigate fully distributed reputation schemes and integrate them in the current data dissemination and sharing approaches. Similarly, further exploration of energy efficient data sharing approaches

are necessary making current smart data dissemination and query approaches usable in opportunistic networks.

REFERENCES

- [1] L. Wang, G. Tyson, J. Kangasharju, J. Crowcroft. *Milking the Cache Cow with Fairness in Mind*. ACM/IEEE Transactions on Networking 2017.
- [2] M. Radenkovic, A. Grundy. *Efficient and Adaptive Congestion Control for Heterogeneous Delay Tolerant Networks*. Ad Hoc Networks, v. 10, n. 7, 2012
- [3] M. Radenkovic, B. Wietrzyk. *Wireless mobile ad-hoc sensor networks for very large scale cattle monitoring*. IEEE ASWN 2006
- [4] M Radenkovic. *Cognitive Privacy for Personal Clouds*. Mobile Information Systems 2016: 7107103:17 (2016)
- [5] A. Keränen, J. Ott and T. Kärkkäinen. *The ONE Simulator for DTN Protocol Evaluation*. SIMUTools'09: 2nd International Conference on Simulation Tools and Techniques. Rome. 2009.
- [6] E. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant manets. Proceedings of ACM MobiHoc, 2007.
- [7] S. Dernbach, N. Taft, J. Kurose, U. Weinsberg, C. Diot, A. Ashkan. *Cache content-selection policies for streaming video services*. INFOCOM p.1-9. 2016.
- [8] H. Flores, R. Sharma, D. Ferreira, V. Kostakos, J. Manner, S. Tarkoma, P. Hui, Yong L. *Social-aware hybrid mobile offloading*. Pervasive and Mobile Computing 36, p.25-43, 2017.
- [9] D. Frey, M. Goessens, A. Kermarrec. *Behave: Behavioral Cache for Web Content*, DAIS 2014.
- [10] Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, A. Chaintreau. *CRAWDAD data set Cambridge/haggle* (v. 2009-05-29), May 2009.
- [11] J Leguay, F Benbadis. *CRAWDAD data set upmc/rollernet* (v. 2009-02-02), Feb, 2009.
- [12] San Francisco cabs data traces, downloaded from the Cab Spotting project, <<http://cabspotting.org/api>> for the period 20 Sep 2011– 20 Oct 2011
- [13] S. Saha, A. Lukyanenko and A. Yla-Jaaski. *Cooperative caching through routing control in information-centric networks*. Proc. IEEE INFOCOM 2013, p. 100–104, 2013.
- [14] D. Bertsimas, V. F. Farias and N. Trichakis. *The price of fairness*. Operations Research, v. 59, n. 1, 2011.
- [15] Bartosz Wietrzyk, Milena Radenkovic, Ivaylo Kostadinov: *Practical MANETs for Pervasive Cattle Monitoring*. ICN 2008: 14-23