Semantic Web in the Fog of Browsers

Pascal Molli and Hala Skaf-Molli

University of Nantes, LS2N, France
{pascal.molli,hala.skaf}@univ-nantes.fr

Abstract. Imagine connecting thousands of web browsers with browser-to-browser connections, sharing storage, bandwidth, and CPU. This builds a fog of browsers where end-user devices are ready to collaborate. Imagine semantic fog applications running in fogs of browsers, querying the linked data servers hosted in the cloud and data hosted in the fog. Fogs of browsers running semantic fog applications create a new massively decentralized infrastructure where RDF data and SPARQL query processing are available both on web servers and on browsers. In this paper, we explore new opportunities and research challenges opened by a fog of browsers for the semantic web.

1 Introduction

Fog computing relies on the collaboration of a multitude of devices located near end-users to provide new services or improve cloud services [12]. By this way, devices in the fog can carry out a substantial amount of storage, communication, and control. Browsers naturally meet most of the criteria for fog computing: they are located near end users, they have storage, CPU, communication and most of all, they are de facto the most widely deployed execution environments in the world. However, browsers do not collaborate. Fortunately, the recent introduction of WebRTC 1 has further extended the capabilities of browsers by introducing support for browser-to-browser communications. This turns browsers into a decentralized execution environment where interactions between humans and web services are enabled without a third party. By just clicking on a web link, a user can join a network of thousands of browsers ready to collaborate. Therefore, browsers are first citizens for deploying fog applications.

Some fog applications are already there. WebRTC peer-to-peer Content Delivery Network such as Peer5 2 can be seen as a fog computing in browsers to distribute live streaming and reduce the load on clouds. CRATE [8] is the first decentralized real-time collaborative editor relying just on browsers.

However, most applications require executing queries where data are hosted in cloud datastores or within the fogs of browsers. Data comes from many unpredictable sources, with different formats, with different schema, making query processing extremely difficult. The semantic web already demonstrated how syntactic and semantic heterogeneity can be significantly reduced relying on linked data principles [3]. Consequently, compact federated query engines are now able to process SPARQL queries within the browsers. Therefore, at a given time, there is virtually billions of browsers running applications able to store RDF locally RDF data and execute SPARQL queries.

By connecting browsers, the fogs of browsers running semantic fog applications create a new massively distributed infrastructure where RDF data and SPARQL query processing are available both on servers and on browsers. In this paper, we explore new opportunities and research challenges opened by a fog of browsers for the semantic web.

1 https://webrtc.org/
2 https://www.peer5.com/
The paper is organized as follows. Section 2 defines semantic fog applications in the fog of browsers. Section 3 presents new opportunities opened by semantic fog applications. Section 4 highlights new research challenges for semantic fog applications. Finally, conclusions are outlined in section 5.

2 Semantic Fog Applications in the Fog of Browsers

A fog of browsers is a set of interconnected browsers with browser-to-browser connections. Such connections are now supported thanks to the WebRTC standard in Firefox, Chrome, Microsoft Edge and IOS. A browser can participate to one or several fogs.

A fog of browsers is accessible through one or several URIs hosted on regular web server. The web server dereferences this address to a JavaScript application bootstrapped with a sample of already connected browsers. This JavaScript application represents a semantic fog application with its own logic. The application is able to manage RDF data and runs SPARQL queries over linked data and/or over data hosted in the fog. We assume that all RDF data are managed following the linked data principles [3].

Once downloaded in the browser, the semantic fog application joins the network of browsers by connecting the browser to at least one of the already connected browsers. Following this approach, at a given time, there is potentially a high number of browsers, running the same application, and all these browsers are connected together. We do not make any assumption about the topology of the network, i.e. structured, unstructured, hybrid or multi-layer. Topology depends on the objective of the semantic fog application. In figure 1, the browsers are connected in an unstructured network, they execute SPARQL queries over data in the fog and 2 datasources hosted in the cloud. The browser $b_0$ contacts the web server hosting the semantic fog application in order to join the network. The web server is returning the semantic fog application and references to two browsers: $b_1$ and $b_2$. $b_0$ contacts one of them to join this fog.

To be usable, a semantic fog application must meet the following requirements adapted from P2P data management [10]:

\footnote{As has been already done in [8] and [6].}
autonomy Each browser participating in a fog of browsers is free to join and leave at any
time. It owns its data and have a full control on it.

query expressiveness A semantic fog application runs SPARQL queries or a subset of
SPARQL. The scope of the query can refer traditional linked data providers and/or fog
participants. Queries can update local data of the browser.

efficiency A fog of browsers is composed by the resources of fog participants and the re-
resources of cloud providers involved in the semantic fog application. The efficient uses of
all resources should result in higher throughput of queries.

quality of service The fog has to improve the user-perceived efficiency of the system.

fault tolerance Quality of service can be maintained for a period of time even in presence
of failures of browsers or failures of linked data providers.

security As an open system, a fog of browsers can be used to steal personal data, attack
other browsers in the fog or attack servers. Access control and resistance to malicious are
crucial for semantic fog applications.

3 Semantic Fog Applications

Deploying Semantic fog applications over a fog of browsers raises several opportunities. In
this section, we present several semantic fog applications illustrating different usages.

3.1 Queries in the fog

```
1 Import swfog.lib;
2 void main()
3 /
4 ∗ configure a virtual shared memory to participants
5 located in Nantes ∗/
6 Overlay.configure(Geolocation='Nantes');
7 String query= 'select ?place where
8 ?place nearby ('+myposition+' 100m)';
9 ResultSet loc = queryExecution.executeSelect(query,model)
10 answer=askUser('Do you like your location?: '+loc)
11 UpdateAction.execute('INSERT DATA'+ 'me'+op:likes+ loc);
12 /
13 ∗ display best most liked places in this fog ∗/
14 every(10s).display(queryExecution.executeSelect(
15 'SELECT ?place COUNT(?likes) {
16 ?place liked ?o} groupby ?place'))
```

Fig. 2. The "tourism in Nantes" semantic fog application

Figure 2 illustrates a potential semantic fog application that helps tourists visiting the
city of Nantes. A visitor can take advantage of the opinions of the others nearby visitors of
Nantes.

1. At line 5, the semantic fog application connects the browser of the visitor to a fog of
browsers where browsers are now located in the city of Nantes.
2. At line 8, the application determines a nearby point of interest thanks to a SPARQL query
over the linked data.
3. The application asks the visitor if she likes her current point of interest and stores her answer locally in the browser as a triple at line 10.
4. At line 12, the application displays every 10s the best liked places in Nantes according to connected browsers.

Of course, this application can be written as a traditional web application. However, a semantic fog application provides valuable advantages:

- As a fog application runs in browsers, the cost of running this application is mostly handled by end-users.
- As the application queries the linked data and retrieves the point of interest as unique URIs, this ensures that the last query of the application is meaningful.
- As the query at line 8 could be executed by many browsers in parallel and generates potentially a huge load on linked data server. A collaborative caching in the fog, as detailed in the next section, can reduce the load on server and makes this application sustainable.
- As the application updates the local database of the browser, this local database can be ephemeral, i.e., just exists as long as the visitor runs this application. "me" can just be a unique identifier that does not reveal my real identity. Such settings can better protect the privacy of users.
- The query at line 12 enhances point of interests with the feedback of connected users. Running this query is challenging if the number of browsers in the fog is high. Security issues are also crucial as browsers have to access data on remote browsers.

3.2 Semantic Collaborative caching

Cyclades [5] is a collaborative caching system that can be used by a semantic fog application as the one presented in figure 2. Cyclades connect similar browsers by assuming that users
with similar queries in the past will certainly perform similar queries in the future. Therefore, data cached at similar nodes could be used to answer queries without using resources of linked data servers.

Cyclades is based on a double overlay networks; the first one builds a random network providing connectivity while the second one incorporates a similarity metric. The similarity metric is able to detect users performing similar queries based on the analysis on their local caches. The two-level network topology of Cyclades is described in figure 3.

In this scenario, the fog is able to reduce the number of calls to data providers. Consequently, this improves data availability and reduces the cost of providing data.

3.3 Queries with the fog

Ladda [6] is a semantic fog application that allows participants to delegate their SPARQL queries over their neighbors in the fog.

For example, one can want to execute:

```python
for each $country in countries
query.execute("SELECT ?software ?company WHERE {
?company dbpedia-owl:locationCountry
[ rdfs:label "$country"@en ].")
```

By parallelizing the execution of queries over different browsers, the execution time of this workloads can be significantly reduced. Figure 4 illustrates a Ladda’s query execution.
In this execution, a browser executes 1509 queries with the help of 6 neighbors in a network composed of 50 participants. Each square represents the execution time of a query on the swim lane of a browser. On this run, the execution time of the workload is 2m37s instead of 3m32s if the workload was executed by one browser.

In this scenario, the semantic fog application allows to share the CPU and bandwidth of browsers for SPARQL query processing.

4 Research challenges

Managing data in the fog of browsers has many scientific problems in common with P2P data management systems [10]. Many works demonstrated how data can be efficiently stored and accessed on structured, unstructured, hybrid P2P networks such as Edutella [9], RDFPeers [4], PierDB [7], GridVine [1] etc. However, the context and objectives of fog of browsers are slightly different:

– The fog approach includes cloud services as basic components of the infrastructure. The fog can just improve the efficiency and the quality of services of data providers as demonstrated in [5] and [6].

– Most of work on P2P data management have been done on TCP/IP networks. However, WebRTC networks used by browsers have several major differences with traditional TCP/IP networks:

1. A WebRTC network is not addressable and basically has no routing. Consequently, contacting a particular browser can be costly.

2. Establishing a WebRTC connection between 2 browsers requires a third party to exchange tokens. Once tokens exchanged, a complex negotiation protocol starts to allow NAT traversal. So, establishing a WebRTC connection can be more costly than a TCP/IP connection.

The constraints of WebRTC change the cost of communications and potentially impact all existing algorithms.

**Customized overlay networks for a fog of browsers.** A fog of browsers connects thousands of browsers over WebRTC. The nature of WebRTC networks and the objective of the semantic fog application can lead to different design choices. As routing is costly in WebRTC, keeping useful neighbors around us in one hop, can be a good strategy for efficiency and quality of service. Indeed, direct neighbors can be contacted at low cost. ‘useful neighbors’ can have different meanings according the application. Many similarity metrics can be defined and many overlay can be combined in the same fog as proposed in [5]. Finding the best similarity metrics, topologies and combinations of topologies for query efficiency and quality of services is clearly an important research direction.

**Dynamic replication and consistency in a fog of browsers.** Data replication is a fundamental concept for improving data availability and performances of query processing. In the context of a fog of browsers, replication contributes to query efficiency, quality of service and fault-tolerance requirements. A replication strategy has to decide what data to replicate, where to replicate and when to replicate. Such decisions are complex in a fog of browsers: the participants are autonomous, the data storage is limited, the communication costs constrained by network topology. Adaptivity of replication to queries seems a good strategy. Materializing data fragments that are frequently retrieved from data providers and spreading them within
the fog can have a significant impact on performances. Defining these fragments, deciding when to replicate them and where to locate them is clearly challenging. Another challenge strongly related to data replication is consistency management. Data needs to be up-to-date. Maintaining consistent data fragments at low-cost in a fog of browsers is clearly challenging.

**Crowdsourcing with a fog of browsers** A browser is not just an execution environment for JavaScript programs. It could also involve humans with their Web of Things devices. By connecting browsers, a fog of browsers also connect people able to collect or curate data. Consequently, a fog of browsers can be seen as a distributed crowdsourcing platform where data are collected, semantified and verified within the fog, before saved on a cloud data providers. How the functionalities of a crowdsourcing platform can be distributed among the fog and cloud providers is an interesting challenge.

**Federated query engines for a fog of browsers.** Federated SPARQL query engines [11, 2] allow to query several data sources in a transparent way. In the context of a fog of browsers, the fog itself could be considered as a new data source that cloud be combined with traditional data providers. However, each fog participant has a fragment of data and has to be contacted to answer queries. Such problems have been partially addressed by P2P data management systems. The challenge is to build a distributed federated query engine running in the fog, able to query data in the cloud and in the fog.

**Security for semantic fog application** If a fog of browsers opens many opportunities, it also brings new threats: A fog of browsers can be used to perform DDOS attacks, to steal personal information from browsers, and to watch people. A semantic fog application has to protect participants and data providers of malicious. Semantic fog applications require appropriate security models.

5 Conclusions

In this paper, we presented how semantic fog applications running in the fogs of browsers creates a massively decentralized infrastructure that extends the semantic web to the browsers of end-users. By this way, the semantic web can take advantage of resources of browsers, including end-users and IoT devices. Semantic fog applications can improve the efficiency and quality of service of linked data providers. It can also enhance the linked data with data provided by end-users.

If some semantic fog applications are already there, more research efforts are needed to fully exploit all the potential of semantic fog applications: pertinent network topologies, dynamic replication, efficient query processing, data quality and security.

Another interesting research questions have not been discussed in this paper: the dynamic-icy of fogs of browsers and how fog of browsers can be combined with distributed ledgers for commercial query processing on the fog of browsers.

References


