A Novel Data Dissemination Model for Organic Data Flows

Anna Foerster[1], Asanga Udugama[1](✉), Carmelita Görg[1],
Koojana Kuladinithi[2], Andreas Timm-Giel[2], and Alejandro Cama-Pinto[3]

[1] University of Bremen, Bremen, Germany
adu@comnets.uni-bremen.de
[2] Hamburg University of Technology, Hamburg, Germany
[3] Universidad de la Costa, Barranquilla, Colombia

Abstract. The number of computing devices of the IoT are expected to grow exponentially. To address the communication needs of the IoT, research is being done to develop new networking architectures and to extend existing architectures. An area that lacks attention in these efforts is the emphasis on utilisation of omnipresent local data. There are a number of issues (e.g., underutilisation of local resources and dependence on cloud based data) that need to be addressed to exploit the benefits of utilising local data. We present a novel data dissemination model, called the Organic Data Dissemination (ODD) model to utilise the omni-present data around us, where devices deployed with the ODD model are able to operate even without the existence of networking infrastructure. The realisation of the ODD model requires innovations in many different area including the areas of opportunistic communications, naming of information, direct peer-to-peer communications and reinforcement learning. This paper focuses on highlighting the usage of the ODD model in real application scenarios and the details of the architectural components.

Keywords: Internet of Things · Opportunistic networks · Organic Data Flows · Reinforcement Algorithms

1 Introduction

The ubiquity of computing devices is growing rapidly. Omnipresent computing devices form the basis of the Internet of Things (IoT). The International Data Corporation (IDC) predicts that by 2020, the total number of connectable devices in the IoT will be above 200 billion [1]. Predicting further, IDC estimates that by 2020, the digital universe (DU) will consist of 44 ZB of data. 10 % of this data will be contributed by IoT based computing devices and 27 % will be created by mobile devices. To address these growth predictions, new networking architectures have to be proposed that focus on areas such as information centricity of networks, resource restricted nature of the computing devices of the IoT and large scale network based services.

R. Agüero et al. (Eds.): MONAMI 2015, LNICST 158, pp. 239–252, 2015.
DOI: 10.1007/978-3-319-26925-2_18

The networking architectures proposed for Information Centric Networking (ICN) address the information centricity of communications [2]. A number of enhancements to current architectures are attempting to address the resource restricted nature of the devices of the IoT (e.g., Constrained Application Protocol [3]). Another set of enhancements to existing computing architectures attempt to address the enablement of large scale network based services (e.g., Virtualisation [4]).

Though these new architectures and architectural changes address a number of issues, none of them place an emphasis on the use of the omnipresent data in a local context. Therefore, we foresee the following as being key issues to be addressed for the deployment of localised solutions of IoT.

- **Non-availability of Information Locally.** While the cloud paradigm allows for more flexibility on the server and infrastructure side, collecting all data into common repositories allowing a range of services to be deployed poses additional issues for the system architecture at the edge of the network. Most information is local by nature, but with cloud based solutions it must be transferred to the cloud, creating requirements dependent on always having the reliable connection to the cloud [5].
- **Pull Style Communications.** Current communication architectures require users to initiate communications to retrieve data (e.g., IP networks, where the communicating parties require the knowledge of host addresses to establish a communication session or in ICNs, where the named data is requested for by a user) or register to receive data when they are created. Further, these architectures are highly dependent on reliable communications and when failures are experienced, complete data are re-requested. This kind of communication architecture hinders probing and sensing of the environment for already available data locally.
- **Vertical Systems.** State of the art IoT applications are typically designed, implemented and deployed based on specific application requirements and for specific use cases. They tend to be largely proprietary and their maintenance cost is often included in the price of the installation itself or carried out based on expensive maintenance contracts that often create difficulties in the long term if the supplier is not the same as the organisation that maintains the systems afterwards. The adoption of this process has resulted in a myriad of isolated systems, all working with different communication standards, and using different applications and data storage solutions. These isolated systems are destined to never interoperate with each other, unless very costly integration efforts are made, sometimes costing more than a new, more advanced system installation.
- **Underutilised Local Resources.** The proliferation of smart mobile devices hosting heterogeneous communication technologies, considerable memory and processing capabilities together with sensory capabilities has brought about an environment in which service provisioning could be done anytime/anywhere. These devices together with the omnipresent resource constrained devices, such as sensors, collectively provide an exploitable environment to deploy

a multitude of services even without the support of infrastructures. These omnipresent resources cannot be utilised in a flexible manner mainly due to the above mentioned issues in the current communication architectures.

Therefore, we are motivated to solve some of the aforementioned issues by proposing a solution that creates an omnipresent distributed local knowledge base among the interested parties. The users will collaboratively sense their environment and opportunistically exchange and merge their data in a localised context, even with the absence of infrastructure networks.

In some scenarios such as environmental monitoring, data can be collected and utilised in an ad-hoc manner where the data has higher importance at its origin and decreasing importance after some time.

The data sources we refer to could range from human carried devices (e.g., smart phones, tablets and wearable sensors) to stationary devices with smart homes and smart cities (e.g., static sensors and actuators) and many more in between (e.g., mobile autonomous robots and autonomous cars).

The objective of this paper is to discuss a novel data dissemination model called the **Organic Data Dissemination (ODD)** model, inspired by human communications, in which the flow of information occurs based on application needs and feedback. Therefore, we propose a radically different communication paradigm, compared to the state of the art of networking approaches. The primary features of this communication paradigm are the non-existence of dedicated endpoints (sources/sinks), leading to a destination-less and thus connection-less communication paradigm, where data is propagated opportunistically allowing a node to self-learn based on the reinforcements received through feedbacks (reinforcement learning [6]).

This paper is structured as follows. The next section provides the details of 2 application scenarios where deployed devices can use the ODD model. The third section details the architectural components of the proposed ODD model. The fourth section builds the story board of one of the application scenarios providing details on how the reinforcement learning based self learning process operates in the ODD model. The last section concludes the paper.

2 Application Scenarios

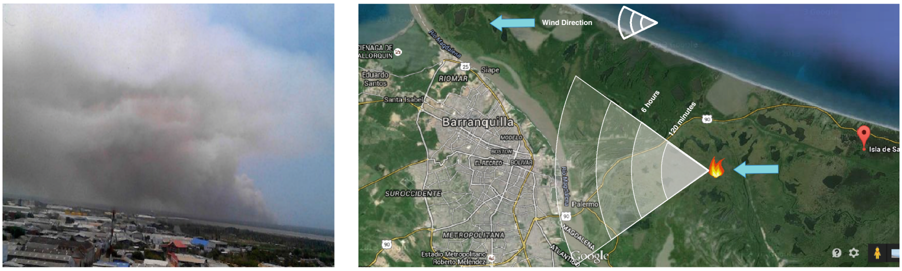
The Organic Data Dissemination (ODD) model has a number of application areas. This section provides the details of 2 real world scenarios that are considered for deployment of ODD based devices.

2.1 Notifications of Natural Fires at VIPS

Barranquilla is a city in the north of Colombia in South America. It is home to over 1 million inhabitants. The city lies to the west of the river Magdalena which flows into the sea at the north of the city. On the west side of the river, facing the city is the nature reserve called the “Via Isla Parque de Salamanca” (VIPS). This nature reserve is home to many wild animal species unique to Colombia.

During the dry season, which falls between December to March, the soil and the vegetation becomes extremely dry in the VIPS nature reserve and becomes prone to natural fires [7]. These fires, when stated and without any interventions may last between 3 to 5 days. Annually, there are around 30 such fires.

These fires result in destruction to the nature reserve and its species. Additionally, the smoke that is generated by these fires is blown away and results in the city of Barranquilla being blanketed by a thick cover of smoke (Fig. 1(a)). The smoke results in breathing problems for children and older people, and furthermore, results in eye problems for many of the residents.



(a) (b)

Fig. 1. The blanket of smoke affecting Barranquilla and a map providing information of the fire

Finding solutions to the above mentioned problems requires answering 2 questions. *How are the fires detected and communicated in a timely manner?* and *What actions can be taken when the fires are detected?*. When identifying a solution, the following aspects must be considered.

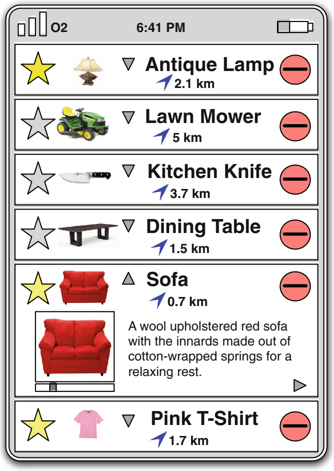
- **Developing country** - Colombia is a developing country. Therefore, the solutions must be cost-effective to deploy and maintain.
- **Climate** - Barranquilla has a tropical savanna climate. It is a hot weather with an all-year-round average temperature of 28 degrees with day temperatures sometimes rising up to 32 degrees. Solutions in such climatic conditions require reliability in terms of operations with minimum downtimes.
- **Deployment effort** - Access to the VIPS nature reserve is very limited due to the harsh terrain. Therefore, a solution must consider the ease of deployment.
- **Interest in community** - Inhabitants in Barranquilla are very keen on having a solution as they are directly affected by the smoke. Therefore, the adopted technologies must include components that allow the inhabitants to get involved in influencing the decisions of the authorities.

There are different sources from which data can be collected for detection in this scenario. This data cannot only help in detecting the origin of the fire, but also assist in the firefighting effort and reduce the impact of the fire (e.g., provide data to families vulnerable to such fires). Directly involved individuals such as forest rangers or government officials, and other parties such as trekkers are able to collect and propagate data about the forest fires due to their proximity to the origin of the fires. For example, air conditions and smoke sensors could be carried by government officials or fire fighters or pictures of air conditions (smoke) could be taken by tourists or residents. This data can be used to build a map of the current fire origin and the wind direction (Fig. 1(b)) to inform the government departments responsible for taking action and also the public.

2.2 Uni Recycler

Another application area of the ODD model is for the "Sharing Economy" [8]. There are many household items that people wish to dispose of, though these items may still be usable. In the context of the University of Bremen, it has been found that there are many freshmen who require such items when they commence their academic lives.

The *Uni Recycler* application is a Smartphone based application for people who wish to make any item available to give away (or for a nominal fee) to anyone who is interested. Figure 2 shows a view of the Uni Recycler application.



A wool upholstered red sofa with the innards made out of cotton-wrapped springs for a relaxing rest.

Fig. 2. Uni Recycler application

The application uses the ODD model to propagate information like in a *Grapevine*. The operations of this application are as follows:

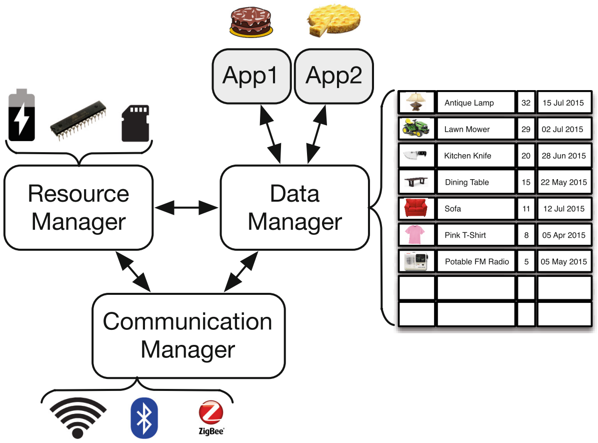
- A user makes information of a recyclable item available in the Uni Recycler app.
- This information is propagated to a selected set of users in the neighbourhood of the original user.
- The neighbourhood users may have differing interest levels for the said recyclable item. This interest may range from liking this type of recycled items to complete dislike. The actions of the user in the app for this particular

item may result in different types of reinforcement leaning messages being generated about the app.
- These reinforcement learning messages are used to compute the relative popularity of the item by all users using the app (popularity is identified numerically by a value called the *Goodness* value in the ODD model).
- The popularity values are used by all users using the app to propagate messages to other users.
- The user who wishes to buy the item may contact the owner through another communication means.

In the remainder of this paper, we will use the UniRecycler application as our main scenario to explain and detail the envisaged ODD model.

3 Overview to ODD Model

In this section, we will introduce the Organic Data Dissemination (ODD) model in detail and define all its main functionality and components. The general model overview is presented in Fig. 3. In the next paragraphs, we will explain the functionality of each individual component. Section 4 will give a detailed example of how ODD works in reality.



	Antique Lamp	32	15 Jul 2015
	Lawn Mower	29	02 Jul 2015
	Kitchen Knife	20	28 Jun 2015
	Dining Table	15	22 May 2015
	Sofa	11	12 Jul 2015
	Pink T-Shirt	8	05 Apr 2015
	Potable FM Radio	5	05 May 2015

Fig. 3. The complete ODD model and its internal interactions.

The ODD model consists of four main components:

- Communication Manager takes care of the communications of the model. This includes scanning the neighbourhood for new connections, sending and receiving messages and evaluating whether the neighbourhood has changed significantly or not.

- Resource Manager takes care so that the ODD model does not drain the resources of the device. The main assumption here is that the device has some primary usage, such as for telephoning or browsing the web and that these primary applications should not be disturbed. The Resource Manager dictates to the ODD model how often it should scan the neighbourhood, how often and how much data should be exchanged, etc.
- Data Manager organises the data cache of the device by the quality of the data items. This quality can also be seen as the popularity or the *Goodness* of the cached data and is simply a number, which reflects how interesting and valuable this particular data item is for all users.
- Applications are sitting in fact on top of the ODD model, but they do implement one key functionality of the ODD model: the evaluation of the *Goodness* value of the data items. Only the applications can say whether a particular data item is interesting and valuable or not, and only they can observe the interactions of the user with the data items.

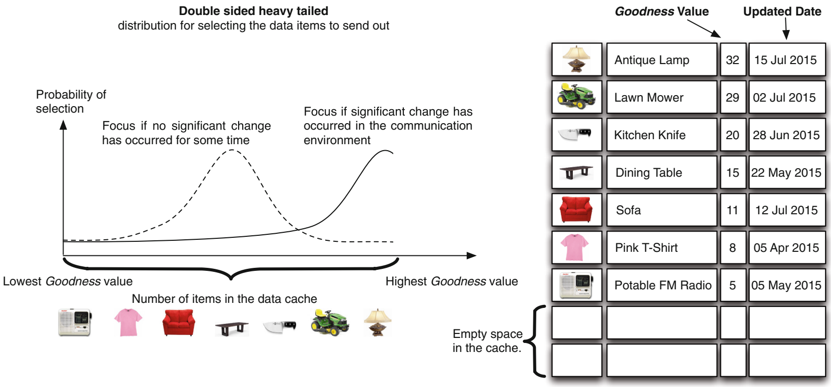
3.1 Data Manager

The *Data Manager* (DataManager) is responsible for all operations related to the cache in an ODD deployed node. It is aware of the size of the cache that it has to manage and uses this cache space to hold the different data items that traverse the node. The operations that the DataManager performs are as follows:

- Ordering data items in an ascending order based on the computed *Goodness* value. The ordering assists in the process of timely access and transmission of data items when required.
- The reinforcement learning model is used to calculate (re-calculate) the *Goodness* values associated with each of the cached data items. The reward function on which the *Goodness* values are computed **reside in the application** that uses that particular type of data (e.g., Uni Recycler). The reward function determines the value of the data based on user interactions or application requirements.
- The *Goodness* values of the data items "age" with time and loses their relative importance. The data manager defines and implements the ageing function.
- The data in the cache are purged using the *Goodness* values. If no space is available for new arriving items, the items with the lowest *Goodness* values are deleted.
- The data which are to be sent out next when a significant change occurs is determined within the DataManager by employing a double sided heavy tailed distribution over all data items. The maximum of the distribution lies where the focus of the DataManager lies (e.g., the focus is set to the highest *Goodness* data items when a significant change of the communication environment has been detected), see Fig. 4.
- When no significant change has been detected, the focus of the DataManager starts going down the data cache to the data items with lower *Goodness*

values, until it reaches the very bottom. The focus remains there until significant change is detected again and then, it jumps back to the top of the data cache.

- The actual selection of data items to be send out is performed in cooperation with the ResManager, which defines how many items can be send out, and performing a stochastic selection based on the above heavy tailed distribution.


Double sided heavy tailed
distribution for selecting the data items to send out
Probability of selection
Focus if no significant change has occurred for some time
Focus if significant change has occurred in the communication environment
Lowest *Goodness* value
Highest *Goodness* value
Number of items in the data cache
Goodness Value
Updated Date
Empty space in the cache.

	Antique Lamp	32	15 Jul 2015
	Lawn Mower	29	02 Jul 2015
	Kitchen Knife	20	28 Jun 2015
	Dining Table	15	22 May 2015
	Sofa	11	12 Jul 2015
	Pink T-Shirt	8	05 Apr 2015
	Potable FM Radio	5	05 May 2015

Fig. 4. Data manager operations

Figure 4 shows the structure of the cache populated with data from the Uni Recycler application. The data are ordered based on the *Goodness* value. The size of the cache (9 entries) is determined by the Resource Manager (ResManager).

When determining which items are to be sent out, the following procedure is followed: First, the Communication Manager (CommManager) signals whether the communication environment (neighbours) has changed significantly since the last time or not. If yes, the DataManager places its focus on the highest *Goodness* data items. Placing its focus means that it actually defines a double-sided heavy tailed probability function over the whole data cache, with the maximum placed at the highest part of the data cache (where data items with higher *Goodness* values reside). This is shown in Fig. 4 (left of the data cache) where the focus lies approximately on the middle of the cache and the double-sided heavy tailed distribution spans the whole data cache. If no significant change has been observed in the communication neighbourhood, then the maximum of the probability function is moved down the data cache.

The reasoning behind this behaviour is that, if the environment has changed (or is changing fast continuously), the most important data items should be exchanged first. At the same time, we do not want to limit the selection to few data items, but also give a chance to the ones with lower *Goodness* values. Thus, we use a double-sided heavy-tailed distribution for selecting the data items, where the maximum of the probability is "focusing" on some part of the data cache at different times.

After setting the focus, the DataManager asks the ResManager how many items it is actually allowed to send out. This number is typically small and the DataManager randomly (based on the double-sided heavy tailed distribution) selects the data items to send out.

This procedure makes sure that in changing environments preference is given to high *Goodness* data items, while not ignoring completely lower *Goodness* ones. At the same time, it also makes sure, through the random selection that if some of the neighbours remain for a longer time in the neighbourhood of the node, they will not receive the same data items repeatedly.

3.2 Resource Manager

The *Resource Manager* (ResManager) is responsible for controlling and providing information related to resources in an ODD deployed node. Access to different resources such as caches, communication links, etc. have to be controlled due to the different characteristics that each of these resources have and the current conditions. Other parts of the ODD model is allowed (or disallowed) to perform actions only under the guidance of the ResManager. Some examples include:

- The ResManager may allow only a limited transmission of cached data items due to current communications being only possible over Bluetooth LE.
- The ResManager hinders the use of the WiFi Direct network interface due to the battery level being below a certain percentage.
- The ResManager allows only a very limited number of data items to be send out because of ongoing video streaming by the user.
- The ResManager disallows any activity between 9 am and 10 am, when it has learned that the user heavily uses the system (e.g. for checking emails and reading news).

To perform these decisions in the ResManager, it is armed with a collection of policies and a set of algorithms that employ scheduling and machine learning techniques. The algorithms use the policies to make decisions initially and the gained experiences are subsequently used to improve the quality of the decisions.

3.3 Communication Manager

The *Communication Manager* (CommManager) is responsible for two main tasks. The primary task is to send and receive data over the network interfaces that are currently usable in an ODD deployed node. The data sent by the CommManager are passed to it by the DataManager. Similarly, the data that is received by the CommManager is passed to the DataManager for further processing.

The secondary task that the CommManager has is the activity of making fuzzy-based decisions related to determining whether the communication neighbourhood has changed. Changes in the neighbourhood are required to be known by the DataManager to consider re-propagation of data.

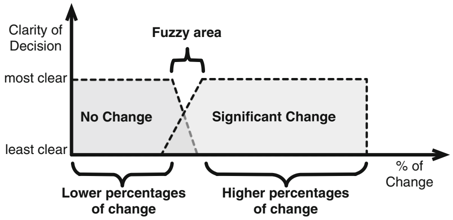


Fig. 5. Determination of neighbourhood changes by the communication manager

Figure 5 shows the fuzzy process of indicating whether a neighbourhood change has occurred or not. When the percentages of change are lower or if the percentages of change are higher, a clear determination is made to consider as the neighbourhood did not change or the neighbourhood changed, respectively. But the overlapping percentages require a fuzzy decision making process to take over. The fuzzy-based decisions assist in correctly and flexibly determining changes in the environment. A simple threshold is hard to identify and not flexible enough.

3.4 Applications

The applications sit on top of the ODD model and interacts directly only with the DataManager. They use the data in the data cache to represent it and interact directly with the user. They can also create new data items (e.g. sensor readings, pictures, messages from the user, etc.).

The applications also implement one of the most important, even if tiny, components of the ODD model, namely the reward function (shown as cakes in Fig. 3). The reason is that only applications can actually identify the *Goodness* of the data items by evaluating them and giving rewards to the DataManager. Different applications might give different rewards for the same data items and these rewards will be combined by the DataManager and its reinforcement learning model into the final *Goodness* value in the data cache. For example, the computation of the *Goodness* value for tapping the data item of the red sofa should be handled by the reward function.

4 The Story of the Red Sofa

In this section we would like to give an intuition of how our proposed ODD model works by relating the story of one particular announcement of the Uni Recycler application, i.e., a *Red Sofa* for sale.

In this story, there are two main actors, Alice and Bob, with their smart devices. Alice has a phablet with a quite large memory and good batteries, while Bob has an older smartphone with an unreliable battery. Alice has a red sofa to sell. She writes an announcement through her Uni Recycler application and saves it. From now on, the ODD model takes care of her message to be delivered to other users, such as Bob or others. The time behaviour of the system have been depicted in Fig. 6.

Now, the red sofa has been saved by Alice and the application passes it to the Data Manager to be saved in the cache. Its *Goodness* value has been set to some constant C, which means that this is a new data item worth spreading (e.g. to 10). The Resource Manager evaluates the usage of the device and sees that it is a good time to check around for new connections and to enhance some data. It fires up the Communication Manager, which scans the environment and decides that it has changed significantly and signals this to the Data Manager. Since the environment has changed, the Data Manager decides that its data items with the highest *Goodness* values should be sent first and puts its focus on the highest 25 % of the ordered cache, which happen to be exactly 20 data items. It also requests the Resource Manager how many items it is allowed to send and through which interfaces. The Resource Manager answers that it can send exactly 3 data items through Bluetooth. The Data Manager randomly selects three data items out of the 20 previously focused ones and the red sofa happens to be among the three selected ones. The Communication Manager sends out the data items and all of them happen to be received by Bob's smartphone (step A in Fig. 6).

The Communication Manager of Bob's smartphone saves the red sofa (and the others, but we are only interested in our red sofa) through the Data Manager to the cache. It also receives Alice's *Goodness* value of the red sofa - currently 10. Now, it needs to decide whether this data item is useful or not. The Data Manager sees that it is new. This is a sign that the red sofa is valuable, but it is still unclear whether it is interesting or not. Thus, it leaves the *Goodness* value as it is (step B in Fig. 6). In other words, it stays neutral towards this data item until more information is available. It asks the Resource Manager whether it is allowed to send out some data and gets a positive answer. It sends the new *Goodness* value of the red sofa plus some of its own cached items (selected as above for Alice) to all its neighbours and also to Alice.

Alice receives the new *Goodness* value which, for clarity, we call feedback, from Bob. Since the feedback is the same as her own value, the *Goodness* value stays as 10.

Now, both Bob and Alice have the red sofa, but they part and everyone goes on their ways through life. Let us first follow Alice for just a little longer. Next time she scans the environment the Communication Manager will signal to her Data Manager that the environment has not changed significantly (in fact, only Bob moved away). Thus, her Data Manager will, at this time, not focus on the highest data items in the cache, but will move its focus to the ones below to also give them a chance to be exchanged. Thus, in longer lived connections between devices, more data with all possible *Goodness* values will be exchanged.

Let us now follow Bob. Bob meets people on his way and pushes the red sofa also to them. He always receives neutral feedback from them, exactly as for Bob before. However, hours later Bob opens the Uni Recycler application and sees the red sofa announcement. He is interested and taps for more information.

He really likes the sofa and decides to call to check the availability of the red sofa later. He taps on the star next to the red sofa and closes the application again. Now the application has new information about how interesting the red sofa is for real users. It gives the red sofa 2 points for being tapped and 5 points for being starred. It passes these points to the Data Manager and the Data Manager re-calculates the *Goodness* value to 17 (step C in Fig. 6).

Bob continues to move around and time passes. Once in a while, the Data Manager re-evaluates the *Goodness* value of the data items and the red sofa gets 1 point off because of ageing. Next time, when Bob meets somebody else and sends the data of the red sofa, it will already have a *Goodness* value of 15, because two update periods have passed (step D in Fig. 6). For example, it might get to Charlie. Charlie will first receive the red sofa with a *Goodness* value of 15 (step E in Fig. 6). However, his Data Manager sees that the announcement is not very fresh any more and gives a feedback of 14 to Bob. Bob will re-compute his *Goodness* value to 14.5 to reflect the fact that the interest of somebody else is less important than the interest of himself (step F in Fig. 6).

Later on, when Charlie opens his Recycler, he will tap the red sofa, because it looks nice, but then he will immediately delete it, because he does not need a sofa right now. The application will evaluate his action and send -3 points (i.e. $2 - 5$) points to the Data Manager. Charlie's Data Manager will re-compute the *Goodness* value of the red sofa to 11 (step G in Fig. 6).

In this way, the *Goodness* value of the red sofa will continue changing over time and with different users. The tendency will be towards smaller values as time passes and the red sofa will start going down the data caches until it needs to get deleted because of space limitations and new data items coming in.

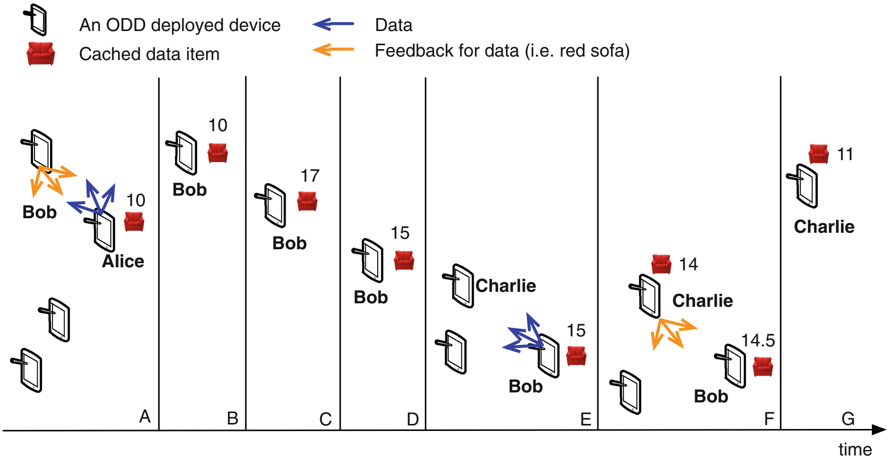


Fig. 6. Evaluation of goodness value for the red sofa (Color figure online)

5 Discussion of the ODD Model

Some interesting questions arise from the story of the red sofa:

- *Will the red sofa reach all users in the network?* Eventually yes, but it will not be flooded. The Resource Manager dictates how much data can be sent out and thus does not increase the traffic with increasing data cache size. Thus, with large amounts of data items in the network, not all data items will get everywhere. The data items with lower *Goodness* values will be deleted.
- *Can we ever delete a data item?* Not really. Instead, as any old rumour, you let it die. People will stop looking at it and the ageing function will make sure that it will slowly degrade its *Goodness* value.
- *Will we not start sending the same data item back and forth?* This can indeed happen, but the randomised selections keep this under control. Prohibiting it by, for example, saving information related to what went where, will introduce more overhead than actually saving resources. The Data Manager takes care of this as detailed in Sect. 3.

In the previous sections we have focused our discussion on one of the proposed applications, the Uni Recycler. However, the ODD model serves very nicely also other applications with similar properties. For example, in the natural fire notification system from Sect. 2, the data items are not things for sale, but sensory data, such as air quality, wind direction and wind strength. They are produced by either the authorities with deployed devices all over the city and the nature reserve, or by citizens with their smartphones, tablets and laptops. The application needs to implement its own reward function. This needs to take into account the freshness of the data and its geographical location. For example, a wind speed data item from 3 min ago is more valuable than something from the day before; tan air quality measurement from a location very close to where we already have a fresh measurement is less useful than from a new and far-away location. Similarly, problematic measurements, such as very high wind speed, fire detections or very bad air quality, are more important than data items which signal that all is good.

In summary, the ODD model offers us a novel way to look at data dissemination in very large distributed environments. It is not meant for targeted heavy-load communications, such as streaming services or phone calls. Instead, it focuses on the type of data which the IoT needs to serve: tiny pieces of information, produced continuously by billions of devices.

6 Conclusion

In this paper, a novel data dissemination model called the Organic Data Dissemination (ODD) model is introduced to exploit the benefits of utilising local data. The data flow in this model, inspired by human communications, is influenced by the data needs of applications and the feedback provided by these applications (i.e., the users of these applications). As application areas of the ODD model, 2 scenarios are considered. One of these scenarios, the “The Story of the Red Sofa” is used to elaborate the operations of the ODD model.

The focus of communications of the ODD model is on local data. Therefore, the underlying communications model that ODD uses is based on direct peer-to-peer communications where the nodes that are deployed with the ODD model is able to operate in networks without infrastructure. When considering a deployment in infrastructure based networks, the ODD model takes aspects such as the availability of the communicating parties into consideration when performing communications.

The next step of the development of the ODD model is to evaluate the performance. To evaluate performance, we are currently building a test-bed with hundreds of smart phones and tablets. The underlying communications are performed using WiFi Direct and Bluetooth Low Energy. To evaluate large scale deployments, a simulation model is being built in the OMNeT++ [9] environment.

References

1. International Data Corporation (IDC): IVIEW, Digital Universe in 2020, December 2012
2. Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., Ohlman, B.: A Survey of Information-Centric Networking (Draft), Number 10492. Dagstuhl Seminar Proceedings. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany (2011)
3. Shelby, Z., Hartke, K., Bormann, C.: IETF RFC 7252, The Constrained Application Protocol (CoAP), June 2014
4. Murray, P., et al.: FP7 SAIL Project, D-5.2 (D-D.1) Cloud Network Architecture Description, January 2013
5. Förster, Anna, Timm-Giel, Andreas, Görg, Carmelita, Giordano, Silvia, Kuladinithi, Koojana: DICE: A Novel Platform to Support Massively Distributed Clouds. In: Pesch, Dirk, Timm-Giel, Andreas, Calvo, Ramón Agüero, Wenning, Bernd-Ludwig, Pentikousis, Kostas (eds.) Monami. LNICST, vol. 125, pp. 273–286. Springer, Heidelberg (2013)
6. Sutton, R.S., Barto, A.G., Learning, R.: An Introduction. MIT Press, Cambridge, MA (1998)
7. Villadiego, K., Velay-Dabat, M.A.: Outdoor thermal comfort in a hot and humid climate of Colombia: a field study in Barranquilla. Elsevier Build. Environ. J. **75**, 142–152 (2014)
8. Sundararajan, A.: From Zipcar to the sharing economy. Harvard Business Review, January 2013
9. Udugama, A., Kuladinithi, K., Foerster, A., Goerg, C.: Federating OMNeT++ simulations with test-bed environments. In: OMNeT++ Community Workshop, September 2015