Towards Low Cost Prototyping of Mobile Opportunistic Disconnection Tolerant Networks and Systems

Milena Radenkovic, *The University of Nottingham*, Jon Crowcroft, *University of Cambridge*,
Mubashir Husain Rehmani, *COMSATS Institute of Information Technology*

***Abstract*— Fast emerging mobile edge computing, mobile clouds, Internet of Things (IoT) and cyber physical systems require many novel realistic real time multi-layer algorithms for a wide range of domains, such as intelligent content provision and processing, smart transport, smart manufacturing systems and mobile end user applications. This paper proposes a low cost open source platform, MODiToNeS, which uses commodity hardware to support prototyping and testing of fully distributed multi-layer complex algorithms over real world (or pseudo real) traces. MODiToNeS platform is generic and comprises multiple interfaces that allow real time topology and mobility control, deployment and analysis of different self-organised and self-adaptive routing algorithms, real time content processing, and real time environment sensing with predictive analytics. Our platform also allows rich interactivity with the user. We show deployment and analysis of two vastly different complex networking systems: fault and disconnection aware smart manufacturing sensor network and cognitive privacy for personal clouds. We show that our platform design can integrate both contexts transparently and organically and allows a wide range of analysis.**

***Index Terms*—Disruption tolerant networking, Mobile ad hoc networks, Prototypes, Wireless communication, Wireless sensor networks**

I. Introduction

Over the recent years there has been a growing interest in designing and testing novel mobile wireless and opportunistic network communication protocols and systems for a wide range of vastly different application scenarios, such as smart manufacturing, mobile social networks and smart wellbeing domains. Researchers increasingly aim to test their novel network architectures and protocols under realistic constraints after initially optimising theoretical models. Newly emerging services and applications for Internet of Things (IoTs) and Cyber Physical Systems (CPSs) require development of new intelligent communication, storage and processing architectures. We propose a novel platform where IoT ubiquitous devices can host services and communicate in peer-to-peer manner via adaptive mobile delay/disconnection tolerant opportunistic networks. This paper describes a novel multi-layer intelligent Mobile Opportunistic and Disconnection Tolerant Networking (MODiToNeS) platform that supports various mobility and connectivity patterns, adaptive communication protocols, and a wide range of smart algorithms for intelligent content processing. We argue that our platform can be used to help research community test highly complex self-organised cognitive distributed protocols and architectures as well as serve as an educational resource which uses open source software, low cost hardware, simple control interfaces and modelling structures. We believe that MODiToNeS will help advance the research and educational opportunities available to the cognitive DTN and opportunistic network communication communities by providing a "real-world" fully distributed platform where researchers and students can develop and test their cognitive protocols and applications while being able to observe how they behave in a real world hybrid (wireless and wired, mobile and static) environments. We argue that it is very important to allow researchers to validate their core assumptions and hypothesis made when proposing new complex algorithms and systems as early as possible to avoid building inaccurate and unusable protocols. MODiToNeS allows us to tackle exactly that through incremental or evolutionary prototyping. Using simple open source interfaces, researchers are able to rapidly develop distributed algorithms and deploy them in a real environment saving time in developing component evaluation and simulation techniques while providing them with valid feedback about how their components interacted with different kinds of dynamic environments. Additional advantages of using MODiToNeS are multifold and include: first, low cost which refers to the total cost of the platform ownership being less than the cost of a mid-high end laptop (around £2000), and second the platform being fundamental to performing early feasibility tests and quick prototyping before embarking in complex simulations (e.g. NS-3[29] or Mininet[30]).

In this paper, we describe two different example scenarios prototyped in MODiToNeS: first, smart manufacturing Fault

This work is supported in part by the project "Health Monitoring and Life-Long Capability Management for SELf-SUStaining Manufacturing Systems (SelSus)" which is funded by the Commission of the European Communities under the 7th Framework Programme, Grant agreement no: 609382.

Milena Radenkovic, is with the School of Computer Science, The University of Nottingham (e-mail: mvr@cs.nott.ac.uk).

Jon Crowcroft is with the Computer Laboratory,University of Cambridge (e-mail: Jon.Crowcroft@cl.cam.ac.uk)

Mubashir Husain Rehmani is with COMSATS Institute of Information Technology (e-mail: mshrehmani@gmail.com).

Aware and Disconnection aware communication smart sensors protocol (FDASS) [10]); and second, privacy aware mobile personal clouds (CogPriv) [4]. More specifically, we show how our open-source distributed platform allows building, deploying and testing of 1) fault aware and disconnection aware framework prototyping and testing for smart manufacturing and 2) adaptive mobile privacy aware personal clouds prototyping and testing when sharing various kinds of data via different routing protocols via networks with different levels of privacy leakage.

The paper is organised as follows. Section 2 reviews related work on state of the art testbeds for smart data communication in mobile and wireless networks. Section 3 proposes the multi-layer architecture of our MODiToNeS platform, introduces its key control planes and describes support for cross layer data communication that can on-the-fly adapt to dynamic link properties and changing requirements of the users. Section 4 describes smart manufacturing opportunistic and disconnection tolerant sensor network architecture scenario prototyping with smart adaptive routing protocols such as FDASS [10]. Section 5 describes peer to peer mobile clouds scenario which deploys smart protocols [4] for data sharing, monitoring and interaction. We show that both CogPriv and FDASS outperform other competitive and benchmark protocols across a range of metrics in line with previously done simulation based work in [4][10]. In addition, we evaluate realistic resource costs for FDASS and CogPriv across a range of resource metrics in real time. Section 6 gives summary and future work directions.

II. Related Work

As mobile edge computing and cognitive networks are still emerging filed, there are limited simulation and testbed environments which allow prototyping and testing of new emerging applications, protocols and services. We review a range of state of the art testbeds for wireless networks and applications, and identify how our proposal defers from each one of them. Similarly, the majority of the current simulator implementations have limited number of control features as they are based on the basic wireless sensor networks and communication protocols. In this paper, we propose a novel multilayer platform that uses low cost smart devices (e.g. such as Raspberry PIs) to prototype rich set of intelligent and interactive complex communication algorithms and distributed network architectures.

[6] propose the design and architecture for a low cost light weight testbed for a Personal Cloud based on Raspberry Pi with a range of sensors (RasPiPCloud). RasPiPCloud supports multiple on demand virtual containers to host different services and applications that can collect, store and share data with varying different levels of privacy. RasPiPCloud utilizes opportunistic networks communication to communicate with the heterogeneous sensors and other devices. RasPiPCloud can have multiple containers [5] such as: Healthcare, Finance, and Social Network with additional container template ready for rapid on demand deployment. Each container gets installed and runs its purpose specific applications to ensure secure data fencing and protection. [6] do not describe the support for multi-user communication. This paper focuses on multi clouds communication support in MODiToNeS.

In [13], authors propose cognitive testbed for wireless sensor networks as an emerging technology with a potential to avoid traditional wireless problems such as reliability, interferences and spectrum scarcity in wireless sensor networks. [13] argue that cognitive wireless sensor networks testbeds are an important tool for future developments, protocol strategy testing and algorithm optimization in real scenarios. This paper focuses on sparse and potentially disconnected topologies in addition to large dense topologies. State of the art work in [16] proposes Haystack system which aims to allow unobtrusive and comprehensive monitoring of network communications on mobile phones entirely from user space. Haystack correlates disparate contextual information to illuminate mobile phone app performance, privacy and security. While Haystack runs locally on a user's phone and can provide highly useful real world data traces that we can use in our platform, our platform is fully distributed and can run different applications and contexts.

TKN Wireless Indoor Sensor network Test-bed (TWIST) [22], developed by the TKN at the TU Berlin, is one of the largest academic testbeds for experimenting with WSN applications at indoor deployment scenarios. It provides basic services like node configuration, network-wide programming, out-of-band extraction of debug data and gathering of application data. It also presents several novel features such as active control of the power supply of the nodes. The testbed in [11] uses setup which consists of 102 TmoteSky nodes operating at 2.4 GHz and 102 eyesIFX nodes at 868 MHz resulting in a fairly regular grid deployment pattern with an inter-node distance of 3 m. The Virginia Tech COgnitive Radio NEtwork Testbed (VT- CORNET) [21]] is a collection of cognitive nodes deployed in a building on the Virginia Tech campus. The testbed consists of a total of 48 static SDR nodes based on USRP210, located at the ceiling. In addition to the static nodes, low-power mobile nodes are also available in order to provide an environment that accommodates a wide variety of research topics. All devices used in this testbed are based on SDR and are not suitable for WSNs because of their high power consumption. Despite their possibilities for frequency mobility, the solution implemented by this test-bed does not support CWSN implementation. Both [22] and [21] focus on the network layer communications and do not consider other IoT (middleware) layer and different application and data types which MODiToNeS includes.

While there has been extensive research and standardisation work being done in the areas of verification and validation for product lifecycle for different application areas [34][35], in our paper we follow general guidelines for the physical prototyping which has been identified as an open research problem for the intelligent mobile opportunistic research community and complex network protocols and systems they aim to propose [34][8]. Physical testing is still an expected industry practice, frequently linked to product certification. Moreover, as we target complex systems modelled with

complex temporal networks assuming likely loss of connectivity and data, physical prototyping generates valuable knowledge and data that can be utilised to enhance the design of future products or variants.

III. Mobile Opportunistic Delay/Disconnection Tolerant Networks and Systems Platform (MODiToNeS)

A. Overview of the MODiToNeS Platform

We propose a novel platform, MODiTOnNeS, which is highly suitable for fast prototyping of applications that are distributed, cognitive (context-aware), intelligent (able to use various on demand self-organised and adaptive routing and machine learning algorithms), interactive and driven by real world connectivity and application traces. MODiToNsS platform contains five programmable layers for dynamic and on demand control including: 1) cognitive/smart hardware devices which are able to store and process real time data and are equipped with different heterogeneous sensors and different types of types of communication interfaces; 2) topology control plane to enable rich diversity of mobile and fixed network topologies; 3) control plane for enabling different intelligent routing protocols; 4) control plane to enable different real time analytics and machine learning protocols which are suitable for different applications and 5) interactive real time user dashboard to allow user interaction and notifications for different application types. This architecture is shown in Figure 1. We argue that it is important to enable different control interfaces for different layers in order to enable a more complete and useful platform that promotes opportunistic disconnection tolerant networking and mobile edge computing research which is fundamental for pervasive computing, IoTs and CPSs research and services.

While MODiToNeS draws inspiration from the work such as Castalia [2],it focuses on different set of properties as we target cross layer design, opportunistic smart communication protocols in challenged networks and enabling high level analysis and visualisation accessible to the user or at the edges. We provide a modular and simple open source implementation (inspired by ONE [18]) for topology control and monitoring, resources monitoring and analysis and data monitoring and visualisation. Each of our smart nodes has multiple communication interfaces which can be programmed dynamically with different parameters to start or stop as well as to control and monitor different wired or wireless active channels which communicate different types of sensor and user application data. MODiToNeS provides the developer with simple open source functionality to change the default interface used to send data on demand as well as to the change the active channel on the fly. In addition, our platform allows running of different real world and connectivity traces in real time which can be mobile, static or hybrid networks. We assume that connectivity traces are in accordance with the syntax of the StandardEventsReader format used in ONE. MODiToNeS also allows programmable topologies where the user can program the dynamic connectivity among the nodes. Our approach allows testing of various protocols against multiple real world conditions by allowing real world topology information retrieval which is fed to the platform in order to mirror any real world topologies. We enable automated reconfiguration of the platform such as experiment repetitions with controlled parameter changing so that our approaches can be tested against multiple real-life conditions. We also allow a user to in a simple way deploy different routing algorithms on the distributed MODiToNeS nodes to support various routing behaviour and topologies. Overview of the layered architecture of the platform is shown in Figure 1.

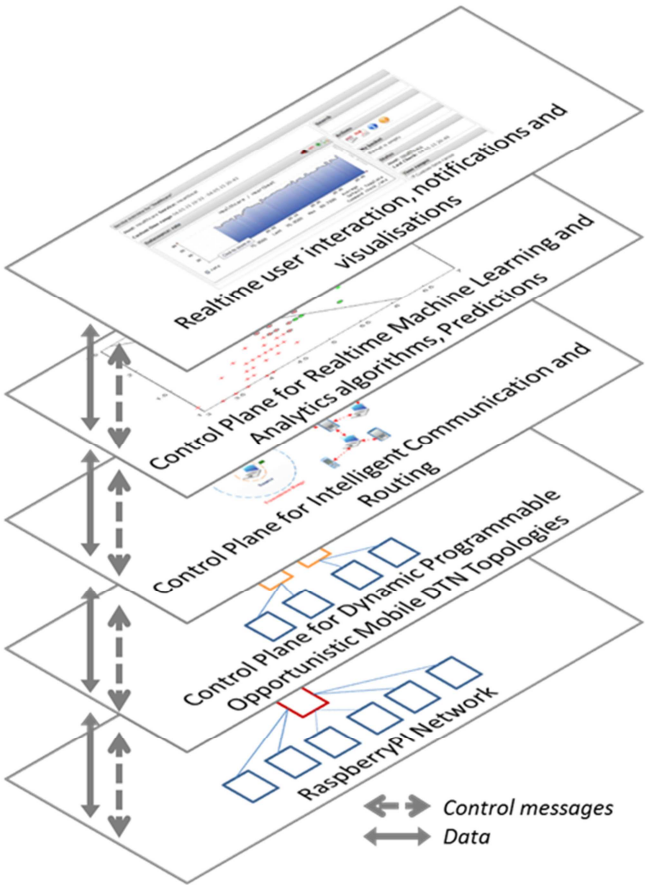

Fig. 1. Overview of the layered architecture of MODiToNeS.

Each MODiToNeS node is a low cost single-board computer (Raspberry Pi model B) which provides good processing power, flexible storage, and has good software support. It can be integrated with a number of wired and wireless sensors using GPIO, I2C, RF modules, 802.11 Wifi, Bluetooth and USB. We currently have over 80 Raspberry Pi nodes. Figure 2 shows one Raspberry PI node which uses on-board Ethernet port, an 802.11n Wi-Fi dongle for wireless network connectivity, and uses IBR-DTN [3,26] to provide P2P DTN capabilities. Figure 3 shows one hierarchical deployment of over 20 Raspberry PIs with different low sensors such as wireless temperature sensor, pressure sensor, magnetometer and 3-axis accelerometer.

Fig. 2. DTNPi with 801.11 adapter and XRF receiver for wireless sensors.

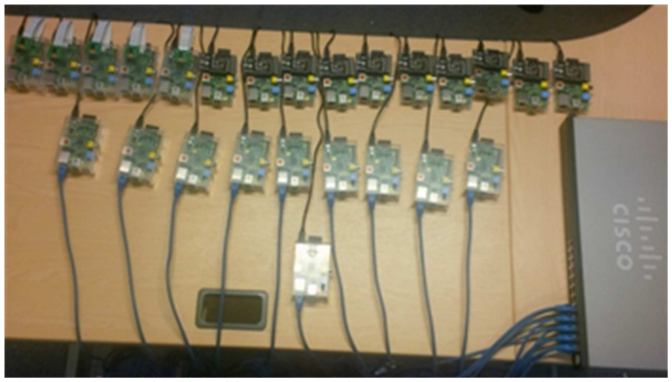

Fig. 3. A hierarchical architecture Raspberry PI sensor network example.

B. *Multi-layer Control Planes*

There is a growing need for generic network platforms that can combine real-world delay and other challenging network conditions with the flexibility of simulators to support a range of different application domains. MODiToNeS enables significantly lower time between the early prototype and production system deployment. Using Raspberry PIs (or similar hardware) allows having large number of hardware nodes in a relatively small physical space at a low cost. For example, influence of mobility on systems performance is complex and is usually evaluated in simulator environments. Contrary to this, our MODiToNeS platform enables the integration of many different distributed mobility patterns.

We have had several diverse projects which benefited from the design and deployment of a generic MODiToNeS platform that satisfies the following requirements:

• Allowing the coexistence of multiple independent projects (e.g. nodes 1-17 for Project A, nodes 20-50 for Project B)

• Allowing experimental orchestration in terms of determining how many run to have, selecting senders and receivers, determining messages rates and sizes, determining which protocols to run.

• Allowing for change of network environment within the scope of a single project to simulated different network topologies for different simulation runs (e.g. nodes 1-5 communicate with nodes 6-10, even ID nodes, communicate with odd ID nodes)

• Allowing for real-time change of network topology emulating DTN [1,27] or MANET where nodes can come in and out of contact with each other within the span of a few seconds.

We address these requirements by proposing real-time programmable interface for network topology configuration. This control plane is used for fast automated and programmable configuration of the network plane. The network layer consists of the head node and variable number of worker nodes directly connected e.g. via a wired Ethernet switch (see Figure 3, Figure 4 and Figure 5). The existence of the head node is fundamental for allowing integrated and holistic view of the design in order to be able to validate in an integrated manner. We have designed, developed and deployed a set of tools on the head node that allow real time configuration changes, command executions, as well as running and deploying distributed smart protocol modules (such as FDASS[10], MWCC [11], CogPriv [4], CafeREP [17], etc) to all nodes. All tools are based on PERL, C/C++, PYTHON and IBR-DTN suitable for our low cost hardware (Raspberry PIs). We use open source PNP4Nagios that visualise RRD files generated by sensor readings and other time series data. Through combination of deploying different routing and content dissemination protocols as well as dynamic/programmable firewall configurations to all nodes, the network topology can be changed and certain nodes can be included or excluded from it. For example configuring a node firewall to drop all incoming and outgoing packets will make this node "invisible" to all other nodes. By changing firewall rules we can achieve any traditional network topologies (such as tree, star or full mash) or any complex temporal networks with our platform (by reading connectivity trace files from *http://uk.crawdad.org* which is a shared wireless network data resources for the research community or generating and capturing similarly formatted connectivity files from other real work devices). We can also exclude a node from the sensor topology while a particular experiment is in mid-run for cases when we want to test a node fault and disconnection. Similarly, we enable real time configuration of different topologies across different experiments for various prototypes. For static topologies, the configured network topology remains the same for the duration of the individual trail. We use a simple format to design and describe the desired network topology. It describes the tuples connectivity that comprises the network topology as well as contains the connection characteristics between the connected tuples as shown below:

```
TimeStamp:Address1:Address2:<UP|DOWN>[,RATE,
DELAY,LOSS]
```

An example connectivity line describing only connectivity is given:

```
0:10.0.10.2:10.0.10.17:UP
```

This executes the below firewall and traffic rate configuration commands:

For Node 10.0.10.2

```
iptables -A INPUT -s 10.0.10.17 -j ACCEPT
iptables -A OUTPUT -d 10.0.10.17 -j ACCEPT
```

For Node 10.0.10.17:

```
iptables -A INPUT -s 10.0.10.2 -j ACCEPT
iptables -A OUTPUT -d 10.0.10.2 -j ACCEPT
```

An example connectivity line describing connectivity with network characteristics is given below:

```
0:10.0.10.2:10.0.10.17:UP:rate 2Mbit,delay
150ms,loss 7%
```

For Node 10.0.10.2:

```
iptables -A INPUT -s 10.0.10.17 -j ACCEPT
iptables -A OUTPUT -d 10.0.10.17 -j ACCEPT
iptables -t mangle -A POSTROUTING -d
10.0.10.2 -j CLASSIFY --set-class 1:100
tc class add dev eth0 parent 1: classid
1:100 htb rate 2Mbit
tc qdisc add dev eth0 parent 1:100 handle
100: netem delay 150ms loss 3%
```

For Node 10.0.10.17:

```
iptables -A INPUT -s 10.0.10.2 -j ACCEPT
iptables -A OUTPUT -d 10.0.10.2 -j ACCEPT
iptables -t mangle -A POSTROUTING -d
10.0.10.2 -j CLASSIFY --set-class 1:100
tc class add dev eth0 parent 1: classid
1:100 htb rate 10Mbit
tc qdisc add dev eth0 parent 1:100 handle
100: netem delay 150ms loss 3%
```

In addition to static topologies, MODiToNEs also supports dynamic topology configuration when the configured network topology is expected to change within the duration of the individual trail. We can use this to emulate DTN [1]and MANET network environments and run prototype experiments with real-world connectivity and data dissemination traces like Infocom [19], Rollernet [9,28], [7] etc. We provide a mechanism that allows us to update the network configuration by using the above mentioned connectivity/network topology format where entries reflect changes in chronological order:

```
TimeStamp:Address1:Address2:<UP|DOWN>[,RATE,
DELAY,LOSS]
7:10.0.10.2:10.0.10.17:UP
128:10.0.10.2:10.0.10.17:DOWN
```

or

```
14:10.0.10.2:10.0.10.17:UP:rate 2Mbit,delay
150ms,loss 7%
58:10.0.10.2:10.0.10.17:DOWN
```

Figures 4a and 4b give an overview of the pseudo code of the master MODiToNeS node and distributed working MODiToNeS nodes respectively.

```
Configure_worker_nodes_DTNRouting(DTNAlgorithm);
Enum MessageGenerator {Sensor, ExternalTrace, PseudoRandom };
Configure_worker_MODiToNeS_nodes_MessageGenaration(MesageGen
erator);

Enum Topology {ExternalTrace, PseudoRandom, Static};
Configure_experiment_topology(Topology);

Enum ContentIntelProc (RealtimeMachineLearning,
PredictiveAnalystic,Heuristics);
Configure_worker_MODiToNeS_node_ContentIntelProc_Algorithm(Con
tentIntelProc);

Dtf= Parse_trace_data_file;
For contact In (<Dtf>)
Do
    FWRule=Generate_FW_rules(contact);
    Configure_worker_MODiToNeS_nodes_FWRules(FWRule);
End For
```

Fig. 4(a). Algorithm for the control planes in the Master node.

```
DTNAlgorithm =
Read_from_master_MODiToNeS_Node_DTNRouting();
Configure_local_DTNRouting(DTNAlgorithm);

MessageGenerator =
Read_from_master_MODiToNeS_Node_Generator();
Configure_local_MessageGeneration(MessageGenerator);

ContentProcAlgorithm =
Read_from_master_MODiToNeS_Node_ContentIntelProc();
Configure_local_ContentProcAlogrithm(ContentProcAlgorithm);

Run_DNT_Thread(DTNAlgorithm);
Run_Generator_Thread(MessageGenerator);
Run_ContentProcAlgorithm_thread(ContentProcAlgorithm);

For FWRule In (Read_from_master_MODiToNeS_Node_FW_rules)
Do
    // node tuples connect or disconnect
Apply_local_FWRules(RWRule);
End For
```

Fig. 4(b). Algorithm for a Worker node.

IV. Smart Manufacturing Scenarios

A. Prototyping Fault and Disconnection Aware Smart Manufacturing

We describe the hardware, software and algorithms we use to prototype smart manufacturing sensor network that we have used in smart manufacturing project EU Selsus [12] and facilitate real world deployment of complex architectures and communication protocols.

We describe the design and implementation for a sensor network prototype in MODiToNEs that can reproduce a production floor sensor network environment and emulate various sensor network topologies and communication

patterns that we then integrate within the EU SelSus project[12]. We assume that we have smart sensing nodes which operate as MODiToNeS nodes and provide sensing, computation, storage and communication together with allowing self-configuration, fault tolerance and self-monitoring. The MODiToNeS platform allows development and evaluation of different novel and benchmark protocols FDASS [10], Prophet [24] and Epidemic/Flooding protocols [23] intended for use with smart sensors.

A common and widely used production environment monitoring sensor network topology is a tree topology of depth 2 – the lowest layer of nodes including heterogeneous sensor nodes, the middle layer including gateways/aggregators/processors and the top layer referring to the head/cloud/master node [30,31,32,33,34]. We build and demonstrate a logical depth 2 tree sensor network topology in MODiToNes comprising of four sensor nodes, two aggregator nodes and one central cloud node (shown in Figure 5).

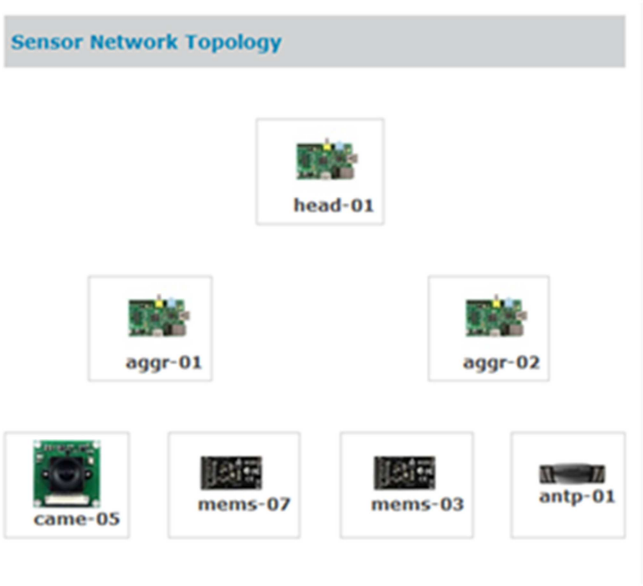

Fig. 5. MODiTONeS topology used in the performance tests for manufacturing scenario.

The MODiToNeS sensor nodes are equipped with a range of sensors including directly attached camera sensor, MEMS Sensor Evaluation Board with Low power 3D magnetometer, 3-Axis digital accelerometer, temperature / high precision pressure sensor, and a remote ANT+ protocols compatible sensor. The MODiToNeS sensor nodes are unable to detect each other's' presence in the network and are only able to communicate with the two MODiToNeS aggregator nodes. Each MODiToNeS aggregator node is able to communicate with the all sensors nodes and the central MODiToNeS cloud node. The MODiToNeS aggregators are also unable to detect each other's presence on the network. The central MODiToNeS node is able to communicate with any of the MODiToNeS aggregators. We assume that, during normal operation of the MODiToNeS sensor node, it captures their corresponding sensors' readings as well its real time local resources utilisation (including CPU load, memory, disk usage, I/O) at configured time intervals. Each node is able to store the measurements locally to be available for localised queries and also generates simple format messages with sensor measurements which are sent to the central cloud node. The only neighbours that any MODiToNeS sensor node detects are the two MODiToNeS aggregators. The individual MODiToNeS sensor nodes can transmit their sensor measurements messages to any of the two MODiToNeS aggregators but in normal operation they "prefer" their local MODiToNeS aggregator. When a MODiToNeS aggregator receives a MODiToNeS sensor reading message, it stores the sensor measurements locally to be available for localised query and also forwards the messages directly onto the central MODiToNeS cloud node. The MODiToNeS aggregator nodes also forward resource measurements to the MODiToNeS cloud node in the same way the MODiToNeS sensor nodes do. Each MODiToNeS aggregator stores the sensor measurements of its belonging sensor nodes and can provide them if queried locally or remotely. In this way, the central MODiToNeS cloud node receives measurements from all MODiToNeS sensors within the sensor network including resource utilisation readings (Fig 6 and Fig 7). All sensor readings are being stored in RRD format as this format is well suited for time-series data like network bandwidth, temperatures, CPU load, etc. The data are stored in a circular buffer based database, thus the system storage footprint remains constant over time. Note that this is distinct from the traditional concept of round-robin scheduling. Readings for separate sensors get allocated their individual RRD databases. Each Raspberry Pi node is running a web service that allows real-time queries of sensor states and reading as well as historical information and visualisation via PNP4Nagios.

Figure 6 and Figure 7 show long term file system utilisation and long term memory utilisation for a MODiToNeS node. We observe that memory utilisation is firmly below the full utilisation and that the file system is not over-utilised as it gets monthly archiving of experiment log to external long term storage.

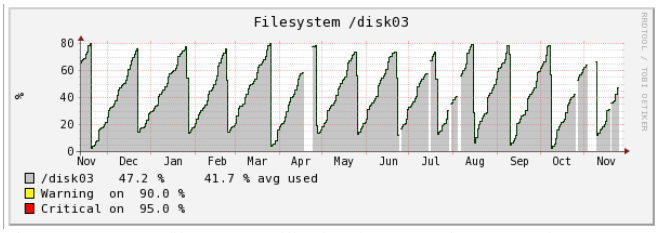

Fig. 6. Long term file system utilisation for a MODiToNeS node.

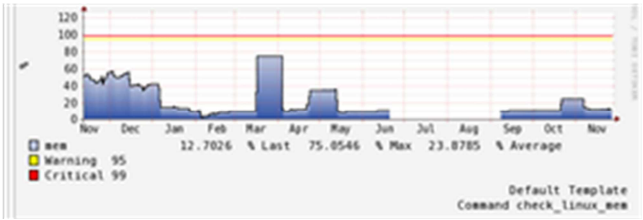

Fig. 7. Long term memory utilisation for a MODiToNeS node.

B. Routing Protocol Evaluation

In order to better understand simulated performance of fault aware disconnection tolerant smart sensor network protocol (FDASS) [10], we prototype and test FDASS against Flooding and Prophet protocols in MODiToNeS. We have developed an intelligent framework that aims to improve reliability of the manufacturing plant in the face of varying network connectivity and non-uniform distribution of different types of faults in the network. Fault and Disconnection Aware Smart Sensing framework (FDASS) [10] is able to detect and identify misperforming nodes in a fully distributed fashion in order to isolate them, reroute the traffic away from them and notify the sinks about the type, location and time of the failures. FDASS builds on and extends multi-path transport approaches to combine fault analytics layer with the complex network topology and resources analytics layer into a complex heterogeneous network for manufacturing environments as shown below. As all MODiToNeS nodes run IBR-DTN on Raspberry PIs, we implemented the FDASS protocol as a IBR-DTN routing component written in C++. IBR-DTN also includes other benchmark protocols (such as Epidemic and Prophet). This allows direct performance comparison between different protocols in a real world environment with MODiToNeS.

Each MODiToNeS node was augmented with a sensor simulator capable of generating varying numbers of pseudo realistic sensor readings on demand. The simulated sensor readings were chosen over real sensors to increase the diversity of sensing ranges and frequencies [31][32][33] compared to the available low cost sensor types we have. The simulated sensor readings were padded to 100 bytes to ensure each reading had a consistent size. Each sensor reading was also timestamped as it was taken with millisecond precision. The head node also timestamped the bundles, with millisecond precision, as they were received. These timestamps were used to calculate the time taken for the bundle to propagate through the network.

Each experimental prototype run lasted 60 minutes with each sensor being polled once every second. Between each run the number of sensors per sensor node was increased by one until ten sensors per node was reached. All nodes were rebooted between each run to ensure the nodes were in a known state. Each experiment was repeated three times and the averages of these three runs were recorded.

Figure 8a shows the recorded bundle delivery success rate achieved and Figure 8b shows the bundle delivery delay observed as the number of sensors per sensor node increases from one to ten. Figures 8a and 8b show that FDASS outperforms both the Flooding and Prophet protocols.

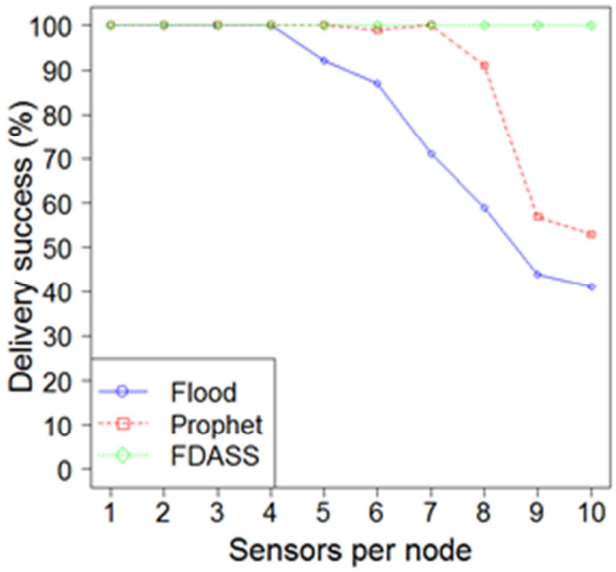

Fig. 8(a). Delivery success with increasing numbers of sensors.

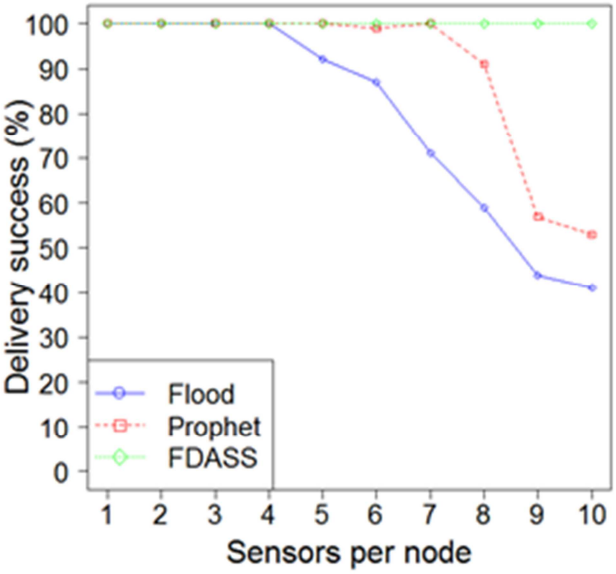

Fig. 8(b). Delivery delays with increasing numbers of sensors.

Figure 9 and 10 demonstrate FDASS robust functionality in MODiToNeS by emulating faults of the MODiToNeS aggregators. Consider MODiToNeS Aggregator 2 fails first by losing network connectivity. We observe in the Figure 9 that all sensor messages get redirected via MODiToNeS Aggregator 1. After a few minutes, MODiToNeS Aggregator 1 loses network connectivity as well. At this stage there is no route between the MODiToNeS sensor nodes and the central MODiToNeS node. During this time, all sensor readings are being stored by the sensor nodes where they are generated. After another few minutes both MODiToNeS Aggregators recover their connectivity and we observe the peak in traffic generated due to the instantaneous delivery of all stored sensor readings.

We aim to deploy MODiToNeS Raspberry PI nodes with

FDASS in the real world manufacturing shop floors to enable further validation with real users and improve reliability of diverse factory communications.

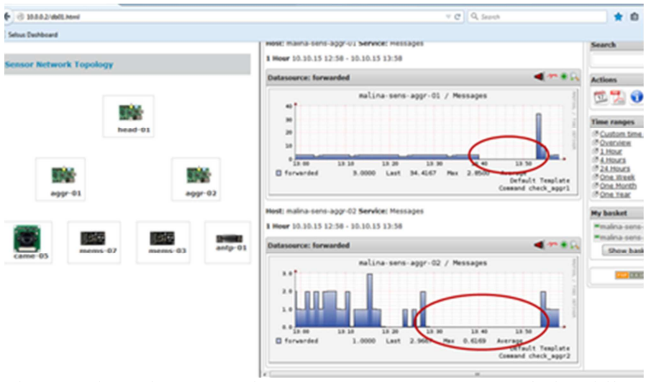

Fig. 9. View of two MODiToNeS aggregators messages during failure disconnections and recovery.

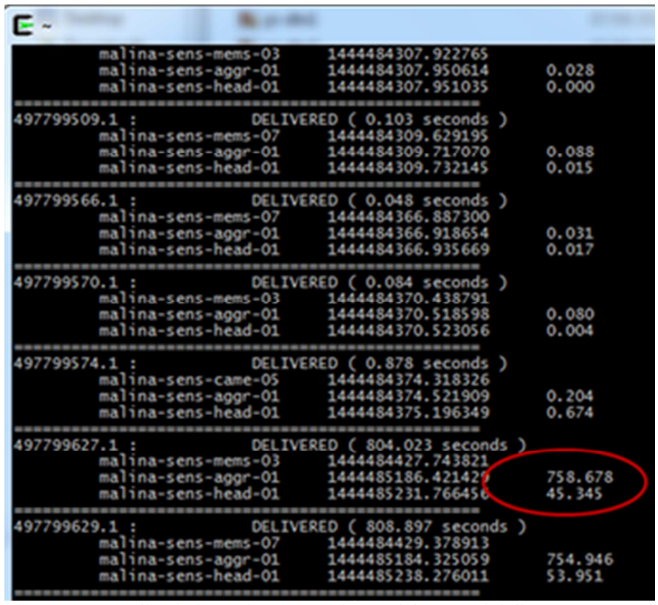

Fig. 10. Centralized view of messages hops, delays, and delivery success.

V. Mobile Clouds Scenario

A. *Prototyping Mobile Clouds Overview*

As an example of a different architecture that can be built and tested in MODiToNeS platform, we describe the design of predictive mobile clouds where mobile sensing and real time predictive analytics algorithms are incorporated in dynamic mobile clusters of MODiToNeS nodes. Each smart MODiToNeS platform node allows intelligent real time decision making that can predict (and change) the behaviour of the network communication of itself and other nodes'. Even though machine learning and analytics techniques have been widely recognised as important for context prediction in mobile computing and many theoretical and simulation based works exist, real world implementations are still scarce and remain interesting future research challenge [8]. The mobile cloud (MC) prototype over MODiToNeS example we describe here supports new paradigm shift that combines anticipatory systems [8] and adaptive collaborative proposals [e.g. 17,20] where computer devices base their actions on the predictive models of themselves, the environment and the other nodes. sMODiToNeS support consideration of multiple criteria including different complex temporal graphs centrality predictions as well as resource, movement and behaviour predictions.

We view mobile clouds (MC) as new approach that bridges the gap between the device(s), environments and the user. In MODiToNeS platform, the prototype of each MC is equipped with a range of sensors (accelerometer, gyroscope, temperature, pressure, heart rate sensor) that can sense the environment and monitor the context, as well as run real time predictive analytics (or other machine learning) algorithms to develop models that predict occurrences of various events. Our MODiToNeS MC also allows rich real time interaction with the users as well as sharing among MCs over different intelligent protocols, different applications and data types. Additionally, each MODiToNeS MC is able to interact with the environment and can adaptively change its behaviour in different situations.

Of particular interest in this platform is to investigate the performance characteristics of our MC smart data communication algorithms in the face of different users' requirements for privacy in different contexts. In [11], we describe a Mobile Wellbeing Cloud Companion (MWCC) testbed prototype which is able to continuously process accelerometer and gyroscope from the physical environment and process the readings using various machine learning algorithms to identify several user activity features. These are analysed in real time and correlated with the heart rate signals to identify if the heart rate is normal or not for the current user activity. In [4], we proposed CogPriv that explored through simulations how different levels of privacy can be supported via adaptively changing network connectivity in both sparse and dense topologies. In this paper, we build CogPriv prototype in MODiToNeS and test it both in terms of quality of the experience metrics (such as achieved end-to-end privacy and delays) and the quality of service metrics (such as memory, I/O, CPU with resource limited devices.).

CogPriv considers users who may be running a social network that allows them to stay in contact with their friends at the same time as regularly monitoring their long term medical condition and being in contact with the hospital. These two types of applications have different privacy requirements and need their data to be stored and shared in different ways in order to adapt to each required privacy requirements dynamically. Figure 11 shows example sensor integration for mobile personal cloud (MPC) prototype in MODiToNeS on a Raspberry Pi with an Xtrinsic sensor board with temperature, pressure, and acceleration sensors. Figure 12 shows MODiToNeS Raspberry PI device that captures, stores and processes a range of user and environment data such as heart rate and pedometer.

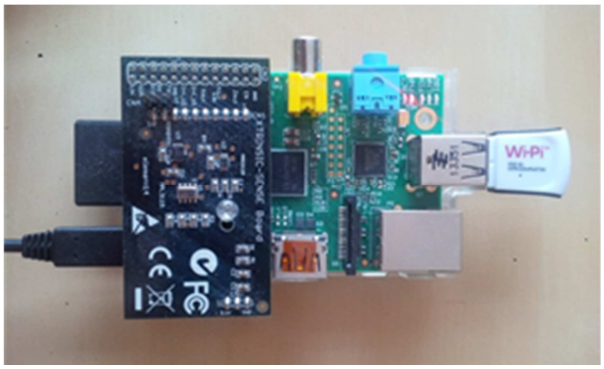

Fig. 11. MODiToNeS Raspbery Pi B with a Xtrinsic sensor board and a WiPi wireless adapter.

Fig. 12. MODiToNeS Raspberry Pi with Suunto and WiPi USB module, Garmin heartrate sensor and a smaprtphone displaying readings.

CogPriv in MODiToNeS extends the bundle protocol based on RFC 5050 [25,26] that provides API for DTN applications to exchange and route bundles among distributed nodes in an intelligent P2P manner. CogPriv P2P DTN (IBR-DTN) module in MODiToNeS provides multi flow real time bundle forwarding based on a range of criteria such as source ID, Virtual Machine (VM) ID, application privacy requirements, destination ID so that different incoming bundles can be matched to the appropriate network interface in real time. At its core, CogPriv comprises multiple stages: it probes local cellular network to identify the likelihood of any middle boxes that may compromise user traffic, requests the remote destination nodes to provide their estimations of the cellular network privacy levels, and collaborates and cooperates with the local network nodes to determine the best local next hop. CogPriv routing protocol can range dynamically and adaptively from providing fully cellular single hop end to end communication to fully localised multi hop mobile opportunistic communication. Through collaborations and cooperation in the local neighbourhoods, each node can understand its environment and neighbours better. More specifically, each CogPriv MODiToNeS node exchanges their own cellular network privacy statistics and predictions to negotiate feasibility of using cellular network for the particular application, analytics of their own resource predictions and social connectivity analytics. Note that both social connectivity traces and middle boxes information are fed to the MODiToNeS master node from external real world traces (e.g. utilising http://uk.crawdad.org/). In this paper, we show measured achieved end-to-end privacy, end-to-end delays, end-to-end number of hops and transitions, I/O, memory and CPU costs. Each CogPriv MODiToNeS node privacy level is important to consider as it is the core criteria for forwarding the data and deciding on the next hop and via which interface. More detailed description of CogPriv Decision Algorithm is described in [4].

B. Cognitive Privacy Experiment Scenario

We carry out evaluation of CogPriv in MODiToNeS against fully cellular communication and fully local social opportunistic networks across a range of different network conditions and user traffic types using a range of metrics. We show how data can be shared with different levels of privacy in light of untrusted infrastructure. We use findings identified in [14,15] that show widespread use of transparent middle boxes such as HTTP and DNS proxies in the cellular infrastructure which are able to analyse and actively modify user traffic and thus compromise user privacy and security. In [4] we provided rich set of simulation based experiments with real world traces of middle boxes [14], connectivity [7], interests [7] and friendships [7]. This paper addresses these scenarios and proposes a way of integrating different layers within our MODiToNeS platform and exploring how different intelligent routing can exploit maximally trusted routes based on the real time probes and collaboration with the MODiToNeS nodes that may be infrastructure nodes or fully ad hoc local nodes based on the local context sensing.

We base our deployment on the real-world data traces of different probes for mobile networks across 112 countries and over 200 mobile providers obtained by netalyzr in [14,15]. We select traces from Germany as its number of mobile networks providers best suits our real world user communication trace [7]. For every mobile node we obtain the probability for the network spying on the web traffic by calculating the percentage of tests returning positive vs the total number of tests performed. For every mobile network, we obtain the probability of it spying on web traffic by averaging the values obtained by all individual mobile nodes on this particular network. Based on the real cellular networks in Germany, we average privacy levels into five evenly distributed privacy threat levels .e.g. minimum (0%) such as ALICE and NETZCLUB, low (25%) such as M-NET, medium (50%) such as BASE, MEDION , high (75%) such as CONGSTAR, maximum (100%) such as FYVE.

While in our previous work, we developed extensions to the ONE simulator [18] that utilise this data in order to return middle boxes presence probability discovered when performing probing of different cellular networks, in this paper we feed this data to MODiToNeS to drive different testbed nodes' behaviour (to act as middle boxes or not). To enable dynamic real world physical connectivity (and disconnections) among MODiToNeS platform nodes, we drive

the MODiToNeS firewall configuration for each MODiToNeS testbed node with the of real world Facebook connectivity traces [7] during the whole time of the experiments. We range the privacy levels of the data being transmitted starting form maximum to minimum privacy requirements with three intermediary levels. We run 5 randomly selected combinations of sources and receivers for each cellular network privacy level.

1) Results

Figure 13 shows that end to end privacy levels remain higher for MODiToNeS CogPriv approach than for cellular only and mobile social ad hoc communication independently of the level of presence of middle boxes in the cellular infrastructure i.e. ranging from no middle boxes to wide range of middle boxes, the performance of cognitive privacy drops from 100% privacy level to 80%. This is in contrast with the cellular network which drops end to end privacy linearly with the amount of the middle boxes in the cellular network. MODiToNeS CogPriv approach also outperforms fully local social ad hoc approach because the delays that are associated with the bundles time out and invoke the nodes to utilise cellular infrastructure that may have privacy leaks.

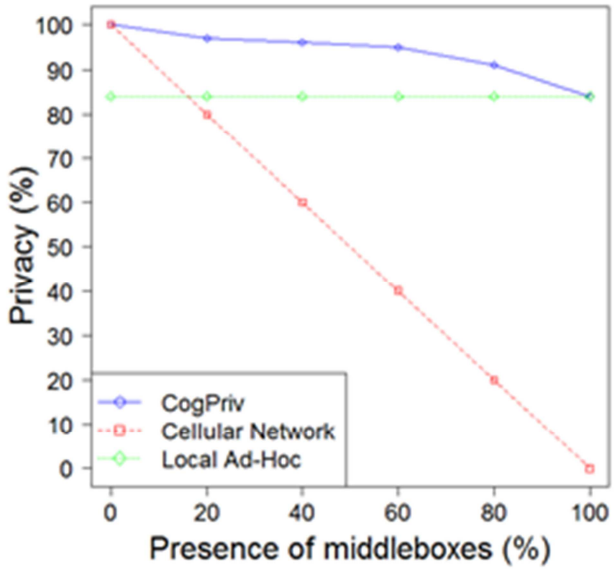

Fig. 13. End-to-end privacy.

Figure 14 shows statistical analyses of MODiToNeS CogPriv number of hops with increased number of middle boxes in the cellular architecture. We observe that the numbers range between 1 and 4 across all levels of middle boxes presence.

Figure 15 shows that MODiToNeS CogPriv delays increase slowly until the infrastructure is fully compromised at which point the delays become the same as the local ad hoc approach. The cellular network approach has the lowest delays but this is due to privacy being compromised and the traffic taking single hop (direct) cellular link between the end nodes.

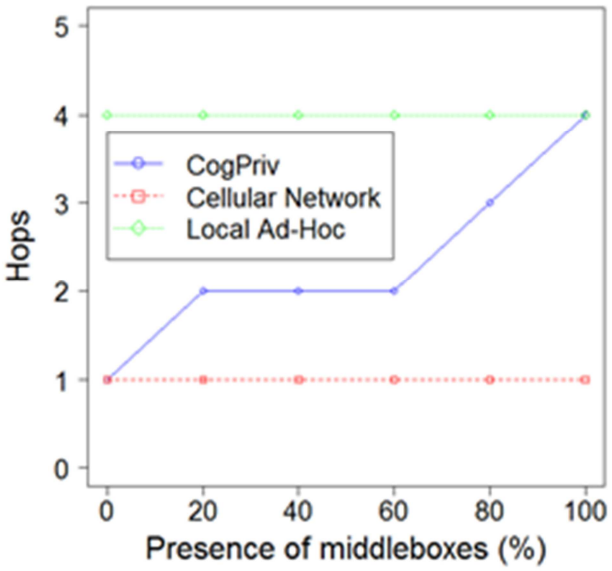

Fig. 14. End-to-end number of hops.

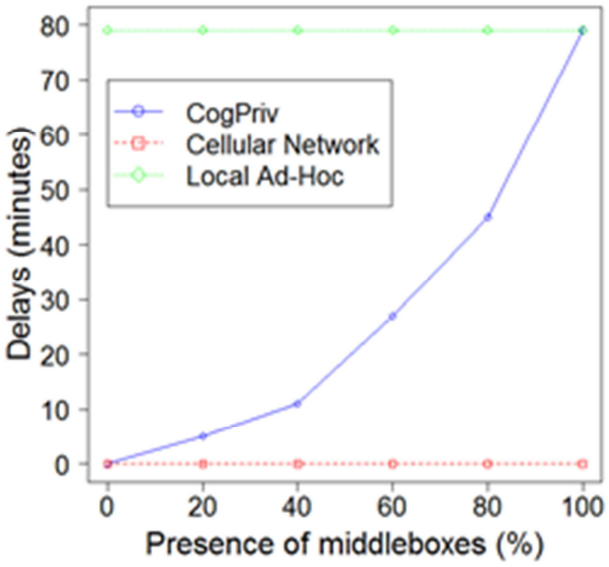

Fig. 15. End-to-end delays.

Figure 15 shows delay distributions for highly private traffic bundles when the cellular infrastructure contains dramatically different amount of middle boxes. We observe that the delays are the lowest when the infrastructure is not compromised as the MODiToNeS CogPriv approach takes cellular single hope router to the destination. As MODiToNeS CogPriv discovers increasing number of middle boxes in the cellular networks, the delays will increase but still be significantly lower than the local ad hoc approach. Even though there are some bundles that may take up to 27 minutes until 60% of surveillance of the cellular network over MODiToNeS, the average still remains low and below 11 minutes. For the cellular network where there is 80% to 100% of middle box presence, the delays range from 45 minutes to 79 minutes. These sorts of delays are appropriate for non-

emergency applications where the users value their privacy and can tolerate delays such as regular daily checks for users with long-term medical conditions.

In Figure 16 we show the number of transitions between i MODiToNeS nfrastructure and MODiToNeS local ad hoc when the security of the cellular network decreases. It is interesting to see that while the number of hops is relatively low (reaching 4 for highly compromised cellular networks), up to 50% of these hops are transitions between the infrastructure and local communication. This shows that supporting adaptive transitioning between infrastructure and local communication is highly beneficial.

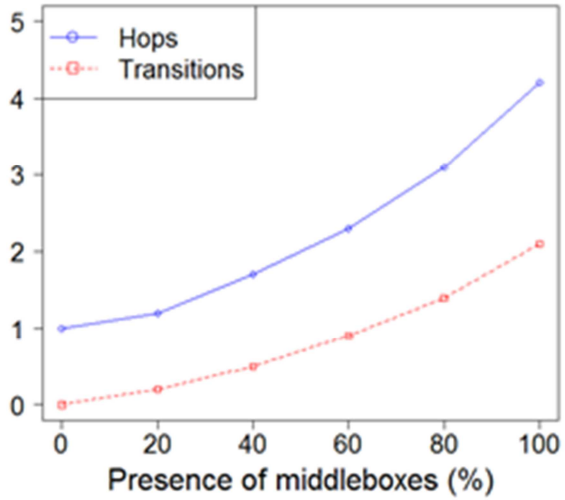

Fig. 16. End-to-end number of transitions.

The previous figures have shown that delays and hop by hop counts increase as MODiToNeS CogPriv moves adaptively from fully cellular mode to the fully opportunistic mode while managing very high levels of end to end privacy. More specifically, we show that the MODiToNeS CogPriv achieves privacy of end to end connections which is almost constant while neither the delays nor the hop count is significantly increased.

Figure 17 shows short term and long term CPU load, memory usage and IO usage for MODiToNeS CogPriv nodes. We observe that, despite complex algorithm and low resources devices, MODiToNeS CogPriv memory usage remains firmly under the full usage. CPU load is in the lower half of the total CPU utilisation for the majority of time while IO at the critical level for the majority of time (note that this critical level has been administratively assigned to b 2K per sec).

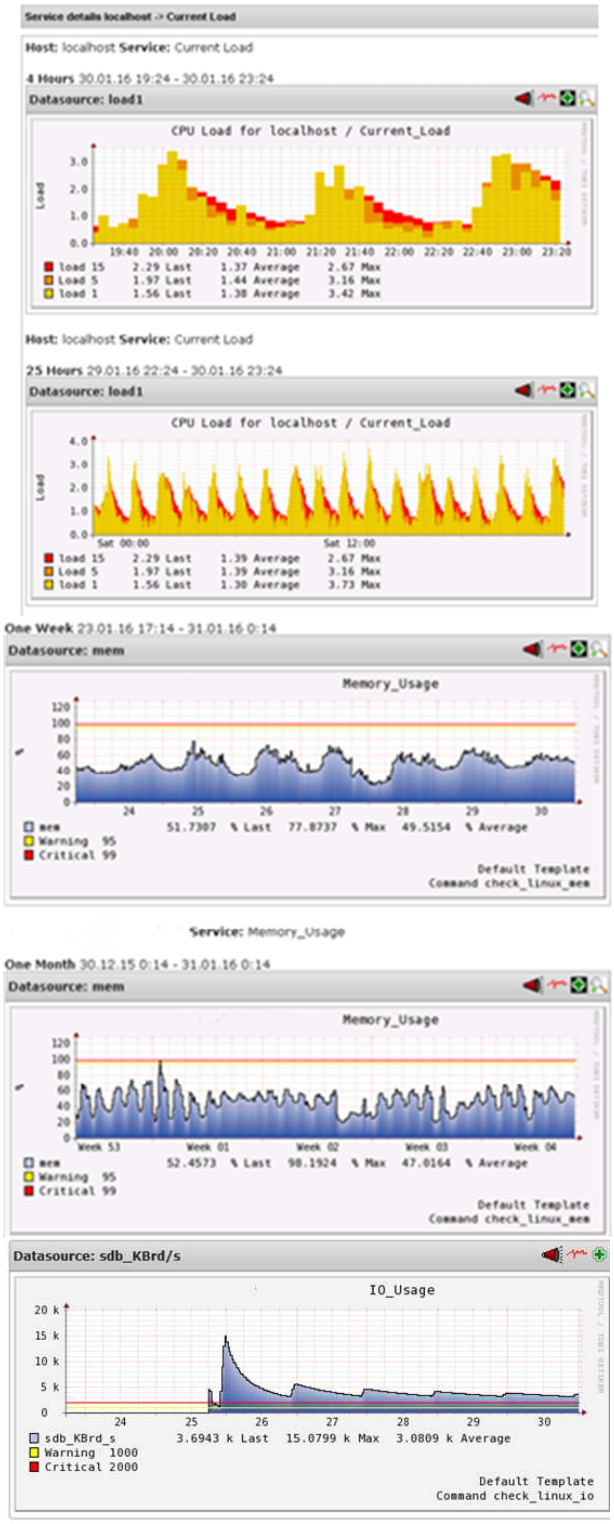

Fig. 17. Short and long term node resource utilisation visualisation.

VI. Conclusions and Future Work

We proposed a novel platform MODiToNeS that supports real time multi-layer and multi-dimensional communication and analysis distributed architectures which can combine various aspects of smart mobile social, transport and other CPS systems with the particular focus on testing real world novel reliable and intelligent communications among potentially low resourced devices.

We envisage increasing need for complex systems of devices including vehicles, humans and infrastructure. Within such systems, various communication paradigms need to be supported including the following: ad hoc communication among people, among vehicles (vehicle to vehicle),

communication between vehicles and infrastructure (vehicles to road side units and vice versa), human and the vehicle (vehicle notifying and guiding the driver as well as the driver providing on the fly information that can potentially differ from the vehicles information) and human and company/home/hospital (human sharing information about their trip/health and getting information or instructions back). In this context, MODiToNeS platform can support the concept of Internet of Things joined with the concept of Internet of vehicles or mobile social networks representing future trends of smart transportation and mobility applications. Current research and services typically allow central remote real time monitoring of various information while MODiToNeS allows users to interact in real time with the prototypes where, query and add additional information on any unexpected events. MODiToNeS builds on and extends existing research to develop a prototype distributed system which allows rich interactivity with the end user and real time localised analytics and predictions as well as remote data communication for non real time analysis. Capturing diverse collection of information locally (which can include any environment and context data), providing real time data analysis and prediction which is visualised and fed back to the users is key for increasing reliability and efficiency of communication in such environments. We envisage that MODiToNeS will play an important role when integrating and testing human behaviour in the design and development of Cyber Physical Systems in mobile social, mobile health care and vehicular networks for critical safety applications.

ACKNOWLEDGMENT

This work is supported in part by the project ”Health Monitoring and Life-Long Capability Management for SELf-SUStaining Manufacturing Systems (SelSus)” which is funded by the Commission of the European Communities under the 7th Framework Programme, Grant agreement no: 609382.

REFERENCES

[1] Scott, K., and Burleigh, S. Bundle Protocol Specification. RFC 5050, November 2007

[2] Castalia wireless sensor network simulator [Online]. Available: http://castalia.research.nicta.com.au.

[3] IBR-DTN - https://trac.ibr.cs.tu-bs.de/project-cm-2012-ibrdtn

[4] M. Radenkovic, Cognitive Privacy for Personal Clouds, Mobile Information Systems, vol. 2016, Article ID 7107103, 17 pages, 2016. doi:10.1155/2016/7107103

[5] LXC – Linux Containers, https://linuxcontaingers.org

[6] Milena Radenkovic and Natasa Milic-Frayling. 2015. Demo: RasPiPCloud: A Light-weight Mobile Personal Cloud. In Proceedings of the 10th ACM MobiCom Workshop on Challenged Networks (CHANTS '15). ACM, New York, NY, USA, 57-58. DOI=http://dx.doi.org/10.1145/2799371.2799373

[7] Annalisa Socievole; Floriano De Rango; Antonio Caputo , Wireless contacts, Facebook friendships and interests: Analysis of a multi-layer social network in an academic environment; In Wireless Days (WD), 2014 IFIP (November 2014), pp. 1-7,

[8] Veljko Pejovic, Mirco Musolesi Anticipatory Mobile Computing: A Survey of the State of the Art and Research Challenges, ACM Computer Survey 47(3): 47 (2015)

[9] Leguay, F. Benbadis, CRAWDAD data set upmc/rollernet (v. 2009-02-02). <http://crawdad.cs.dartmouth.edu/upmc/rollernet>,February 2009

[10] M. Radenkovic, I Kostadinov, B. Wietrzyk, “Increasing Communication Reliability in Manufacturing Environments”, in IEEE IWCMC 2015, Dubrovnik, Croatia, 1377—1383

[11] M. Radenkovic, S. Ha, Low-Cost Mobile Personal Clouds, to be published in IEEE IWCMC 2016

[12] EU FP7 “SelSus:Health Monitoring and Life-Long Capability Management for SELf-SUStaining” FP7-NMP 609382 http://www.selsus.eu/

[13] Elena Romero, Javier Blesa, Agustın Tena, Guillermo Jara, Juan Domingo and Alvaro Araujo “Cognitive Test-bed for Wireless Sensor Networks” IEEE DYSPAN, 2014

[14] Narseo Vallina-Rodriguez, Srikanth Sundaresan, Christian Kreibich, Nicholas Weaver, and Vern Paxson. 2015. Beyond the Radio: Illuminating the Higher Layers of Mobile Networks. In Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '15). ACM, New York, NY, USA, 375-387.

[15] Narseo Vallina-Rodriguez, Srikanth Sundaresan, Christian Kreibich, Vern Paxson “Header Enrichment or ISP Enrichment?: Emerging Privacy Threats in Mobile Networks”, in ACM SIGCOMM otMiddlebox, London, UK, 2015

[16] Abbas Razaghpanah, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Christian Kreibich, Phillipa Gill, Mark Allman, Vern Paxson “Haystack: In Situ Mobile Traffic Analysis in User Space”, arXiv 2015

[17] Milena Radenkovic, Andre Grundy: Efficient and adaptive congestion control for heterogeneous delay-tolerant networks. Ad Hoc Networks 10(7): 1322-1345 (2012)

[18] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. 2009. The ONE simulator for DTN protocol evaluation. In Proceedings of the 2nd International Conference on Simulation Tools and Techniques (Simutools '09), ICST, Brussels, Belgium, Article 55 , 10 pages. DOI=http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5674

[19] James Scott, Richard Gass, Jon Crowcroft, Pan Hui, Christophe Diot, Augustin Chaintreau, CRAWDAD dataset cambridge/haggle (v. 2009-05-29), downloaded from http://crawdad.org/cambridge/haggle/20090529, doi:10.15783/C70011, May 2009.

[20] Radenkovic, M.; Benslimane, A.; McAuley, D., "Reputation Aware Obfuscation for Mobile Opportunistic Networks," in Parallel and Distributed Systems, IEEE Transactions on , vol.26, no.1, pp.230-240, Jan. 2015, doi: 10.1109/TPDS.2013.265

[21] T. Newman, A. He, J. Gaeddert, B. Hilburn, T. Bose, and J. Reed,“Virginia tech cognitive radio network testbed and open source cognitive radio framework,” in Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops, 2009. TridentCom2009. 5th International Conference on , 2009, pp. 1–3

[22] V. Handziski, A. Ķopke, A. Willig, and A. Wolisz, “Twist: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks,” in Multi-hop ad hoc networks: from theory to reality REALMAN ’06 Proceedings of the 2nd international workshop on , 2006, p.63 – 70

[23] A. Vahdat, D. Becker, Epidemic Routing for Partially Connected Ad hoc Networks, Technical Report, Citeseer, 20

[24] A. Lindgren, A. Doria, O. Schelen Probabilistic routing in intermittently connected networks• Lecture Notes in Computer Science (2004), pp. 239–254

[25] Sebastian Schildt, Till Lorentzen, Johannes Morgenroth, Wolf-Bastian Pöttner, and Lars Wolf. 2012. Free-riding the BitTorrent DHT to improve DTN connectivity. In Proceedings of the seventh ACM international workshop on Challenged networks (CHANTS '12). ACM, New York, NY, USA, 9-16. DOI=http://dx.doi.org/10.1145/2348616.2348619

[26] M. Doering, S. Lahde, J. Morgenroth, and L. Wolf. Ibr-dtn: an efficient implementation for embedded systems. In Proceedings of the third ACM workshop on Challenged networks, pages 117–120. ACM, 2008.

[27] Kevin Fall. 2003. A delay-tolerant network architecture for challenged internets. In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '03). ACM, New York, NY, USA, 27-34. DOI=http://dx.doi.org/10.1145/863955.863960

[28] Farid Benbadis, Jeremie Leguay, CRAWDAD dataset upmc/rollernet (v. 2009-02-02), downloaded from

http://crawdad.org/upmc/rollernet/20090202, doi:10.15783/C7ZK53, Feb 2009.
[29] NS3, URL: https://www.nsnam.org/
[30] B. Lantz, B. Heller, and N. McKeown, A network in a laptop: rapid prototyping for software-defined networks. In Proc of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (Hotnets-IX). ACM, New York, NY, USA, , Article 19 , 2010
[31] Alexander Bleakie, Dragan Djurdjanovic, Analytical approach to similarity-based prediction of manufacturing system performance, Computers in Industry, Volume 64, Issue 6, Aug 2013, Pages 625-633
[32] M. Hermann, G. (1990). Artificial Intelligence in Monitoring and the Mechanics of Machining. Computers in Industry vol. 14(1-3), 131-135
[33] Hu W, Starr A, Leung A. A multisensor-based system for manufacturing process monitoring. Procs of the Institution of Mechanical Engineers Part B: Journal of Engineering Manufacture. 2001;215(9):1165-1175.
[34] P.G. Maropoulos, D. Ceglarek, Design verification and validation in product lifecycle, CIRP Annals - Manufacturing Technology, Volume 59, Issue 2, 2010, Pages 740-759, ISSN 0007-8506, http://dx.doi.org/10.1016/j.cirp.2010.05.005.
[35] ISO/IE IS 9000:2015 - Quality management, http://www.iso.org/iso/home/standards/management-standards/iso_9000.htm