Sailing over Data Mules in Delay-Tolerant Networks

Samir Medjiah, *Member, IEEE,* Tarik Taleb, *Member, IEEE,* and Toufik Ahmed, *Member, IEEE*

***Abstract*—In this paper, we address the problem of efficient routing in delay tolerant networks. We propose a new routing protocol dubbed as ORION. In ORION, only a single copy of a data packet is kept in the network and transmitted, contact by contact, towards the destination. The aim of the ORION routing protocol is twofold: on one hand, it enhances the delivery ratio in networks where an end-to-end path does not necessarily exist, and on the other hand, it minimizes the routing delay and the network overhead to achieve better performances. With ORION, nodes are aware of their neighborhood by the mean of actual and statistical estimation of new contacts. ORION makes use of autoregressive moving average (ARMA) stochastic processes for best contact prediction and geographical coordinates for optimal greedy data packet forwarding. Simulation results have demonstrated that ORION outperforms other existing DTN routing protocols such as PRoPHET in terms of end-to-end delay, packet delivery ratio, hop count, first packet arrival and queues occupancy.**

***Index Terms*—DTN, geographic routing, predictive routing, trajectory-assisted routing, mobile networks, time series analysis, ARMA process.**

I. Introduction

DELAY/TOLERANT Networks (DTN) may often refer to sparse mobile ad-hoc network, where an end-to-end routing path does not necessarily exist. In DTNs, both nodes and links may be inherently unreliable. Due to these constraints, these networks are referred to as "challenged networks" [1] [2]. Many other emerging communication networks fall into this paradigm. Vehicular ad-hoc networks (VANETs), mobile sensor networks, and nomadic community networks are few examples.

An interesting DTN example is the city bus network, in which nodes consist of buses (cars, taxis, trams…) and communicate using short-range radios. With this type of networks, we can envision a lot of new applications: urban sensing, information dissemination (advertisement, traffic information, buses software update…) or even Internet access. Since this type of networks does not rely on an existing infrastructure, and they are formed in an ad-hoc fashion, they may be an excellent solution for information dissemination in certain cases when there is no communication infrastructure or the existing one is down. Recently, revolution movements have gained some countries in North Africa. To counteract these protestation movements, government actors have shut down network infrastructures and disconnected the country from the internet in order to prevent people from accessing social networks. Many alternatives solutions have risen in the web. These propositions (such as the OpenMesh Project [3]) all agree to provide mechanisms to establish networks in ad-hoc fashion and to disseminate easily important information.

The proper functioning of such applications relies essentially on the efficiency of the routing task. However many challenges affect the routing in DTNs such as the changing network topology due to intermittent connectivity which is inherent to mobile networks as well as to static networks (in the case of low duty cycle of the nodes), and it results in low delivery ratio and high end-to-end delay. The problem of intermittent connectivity can be mitigated if the exact schedule or the dynamics of the network is known in advance. However, this is not often the case in DTNs as building this knowledge is an important issue. Thus, the efficiency of a DTN routing protocol relies essentially on the amount of network knowledge or "oracles" (information about contacts, queues or even data traffic) available to perform routing decisions.

Several routing protocols have been proposed for DTNs. These protocols differ by the amount of implemented oracles. Depending on the application, some oracles may not be used. For example, in a city bus network, it may not be possible to embed the entire schedule of contacts between buses in each node due to various reasons: (1) the memory space needed to store such information may be of huge size for a communicating node, (2) the "frozen" schedule may not reflect the actual networks dynamics since the schedule of a bus is not "certain" as it may be shifted during the day, (3) the exploitation of such information (for example, computing the best end-to-end route) can be highly computational costly. Then, the challenge in designing an efficient routing protocol for such networks is to make the communicating nodes smarter by using little information about the network and in a reliable distributed fashion.

In this paper, we propose ORION, a routing protocol for mobile DTNs that capitalizes on the localization information of the nodes (geo-coordinates) and the nature of contacts between this type of nodes (buses, cars, taxis, trams) in an urban area. In our earlier work [4], we presented initial results of inter-nodes contact behavior analysis and modeling. In this paper, we provide more details about a potential target application that can benefit from our routing protocol. We also provide details about the contact model derivation steps. Moreover, extensive simulations have been conducted and new metrics have been computed in order to efficiently prove the

Manuscript received March 19, 2012; revised February 19, June 17, and September 7, 2013; accepted November 4, 2013. The associate editor coordinating the review of this paper and approving it for publication was T. Hou.

S. Medjiah and T. Ahmed are with CNRS-LaBRI/Univ. de Bordeaux, 351, Cours de la Libération F-33405, Talence Cedex, France (e-mail: medjiah@ieee.org; tad@labri.fr).

T. Taleb is with NEC Laboratories Europe, Kurfürsten-Anlage 36 69115, Heidelberg, Germany (e-mail: talebtarik@ieee.org).

Digital Object Identifier 10.1109/TW.2013.123013.120398

performances of the proposed routing protocol.

The contribution presented in this paper is twofold. First, we have investigated deeply the inter-nodes encounter behavior. Second, based on this behavior analysis, we proposed ORION, a routing protocol that relies on predicting future contacts between nodes and greedy geographic forwarding of data packets. Thus, with ORION protocol, a communicating node will incrementally build knowledge about its network regarding the inter-nodes encounters behavior and nodes positions. Thereby, it should be able to predict when it will be in contact with other nodes and for how long (i.e. duration). In this paper, we have also investigated the requirements of ORION protocol in terms of computation and memory space for time series analysis and forecasting, and storage requirements for bundle carrying in the context of a store-and-forward routing protocol.

The remainder of this paper is organized as follows. Section II presents the state of the art for DTN protocols and the use of stochastic processes and time series analysis in network communication modeling. Section III presents details about the proposed ORION protocol. In this section, we introduce the target application (Subsection III.A), and then we present our inter-nodes encounter behavior analysis (Subsection III.B). Based on these analysis' results, we define our routing protocol (Subsections III.C and III.D). Section IV provides extensive simulation results and related discussion. Finally, Section V concludes the paper and highlights our future work.

II. Related Work

In order to overcome the mentioned challenges in DTNs, it is important to design an efficient routing protocol that uses small network topology knowledge to maximize the delivery ratio and minimize the delay. Several routing protocols have been proposed for DTNs. These protocols can be classified into two main categories; replication-based and prediction (forwarding)-based protocols. With replication-based protocols, the contacts are assumed to be totally opportunistic and the required topology knowledge at each node is minimal. In this case, the simplest way to deliver a message is to send a copy to each encountered node. This is repeated until the destination receives the message. The Epidemic Routing protocol [5] envisions this strategy. With prediction-based protocols, only a single copy exists across the network at a given time. The protocol needs to be supplied with more knowledge about the network. Given the unavailability of topology information, some protocols try to use probabilities to predict the contact. However, such prediction can be at the price of reduced delivery ratio. Most of the existing prediction-based routing protocols focus mainly on whether two nodes would be in contact in the future, without paying much attention to "when" the contact will happen or "for how long" the contact will last. This lack of contact timing information degrades the contact prediction accuracy and negatively impacts the routing performance.

A. DTN Routing Protocols Taxonomy

As mentioned earlier, the replication-based routing strategy can achieve high delivery ratio while operating with minimal knowledge. This can be suitable for networks where contacts between nodes are unpredictable and random. However, this strategy is not optimal in terms of transmission and buffer size. It also suffers from the lack of scalability. Some protocols, adopting this strategy cope with this problem by bounding the number of copies in the network trading delay for buffer occupancy. To limit the replication, two solutions are used:

- Fix the number of copies and spread them through distinct nodes. Spray & Wait routing protocol [6] uses this solution, also called quota-based DTN routing protocols.
- Use metrics based on historical encounters between nodes to decide whether to send a copy or not. PRoPHET [7] (Probabilistic Routing Protocol using History of Encounters and Transitivity) protocol uses this solution.

The PRoPHET protocol utilizes an algorithm that makes use of the non-random aspect of the real world. This is done by maintaining a set of delivery success probabilities to known destinations, and by replicating messages during opportunistic contacts. Replication is done only for an encountered node which does not have a copy of the message and has a good probability to deliver the message to its final destination. Given a node i, the probability of node i to encounter another node j is denoted as $P(i,j)$. The delivery probabilities are computed during each contact driven by the following three rules:

1) *Updating*:
$$P(i,j)_{new} = P(i,j)_{old} + (1 - P(i,j)_{old}).L_{encounter}$$
where $L_{encounter}$ is an initializing constant.

2) *Aging*:
$$P(i,j)_{new} = P(i,j)_{old}.\gamma^n$$
where γ is an aging constant and n is the number of time units elapsed since the last aging.

3) *Transitivity*:
$$P(i,k)_{new} = P(i,k)_{old} + (1 - P(i,k)_{old}).P(i,j).P(j,k).\beta$$
where β is a scaling constant.

In the prediction (forwarding) based protocols, a node is associated with a forwarding quality/probability metric for each destination, which is usually a direct (*one*-hop) forwarding quality such as contact frequency [7], or time elapsed since last contact [8][9][10].

One drawback of prediction-based routing protocols for DTNs lies in the fact that good forwarding cannot be guaranteed. Indeed, the forwarding quality of a future contact should not only consider the date of the inter-nodes contact but should also takes into account other information such as the predicted contact's trajectory, its reliability, the radio link quality, etc.

B. Times Series in Network Modeling

The proposed ORION protocol makes use of time series to predict contacts. A time series is an ordered sequence of values of a variable $\{Y_t\}_{t \in T}$ indexed by an ordered set $T = \{t_1, t_2, ..., t_N\}$. The time series analysis serves two purposes: (1) Obtain an understanding of the underlying forces and structure that have produced the observed data, and (2) Fit a model and proceed to forecasting, monitoring or even feedback and feedforward control. Time series analysis is used for many applications such as economic forecasting, sales forecasting, budgetary analysis, stock market analysis, yield projections, process and quality control, etc. Recently, it starts being used in the field of computer communications. Indeed, time series have gained the attention of many researchers for the modeling of the Internet and wireless mobile networks traffic. In [11], Basu *et al.* have modeled the Internet traffic using ARMA process of order (p, q). Using this model, they predict the traffic generated by a TCP source using FDDI protocol. In [12], Liu *et al.* have proposed an energy efficient technique for data collection in Wireless Sensor Networks. A sensor is hold from transmitting redundant data. The data are not sent if they can be predicted by the sink node. For prediction, they utilize ARIMA model of order (p, d, q) [13] due to its outstanding model fit and small computational cost. In [14], Herbert *et al.* extend this idea to the hierarchic routing protocol LEACH [15] by providing verification at the cluster head. This approach has shown great communication cost savings. In [16], Banerjee *et al.* used a *birth and death* process to model the network's dynamics. A node entering in the transmission range of a source node is considered as a *birth*. Similarly, a *death* refers to when it leaves this range. Finally, in [17], Singh *et al.* extend this idea by using an AutoRegressive (AR) process to model the number of a node's neighbors in a mobile ad-hoc network. When dealing with stochastic processes, values of the involved random variables are taken over time forming the time series for further analysis. An important step while analyzing time series is to determine the suitable model (or class of models) fitting the observed data. A common approach to analyze time series is the use of ARMA (AutoRegressive Moving Average) analysis. An ARMA process is a combination of an autoregressive process (AR) and a moving average (MA) process. In an AR process, a random variable is "explained" by its past values rather than other variables. While with MA process, a random variable is supposed to be explained by its actual mean, augmented by a weighted sum of the errors (random shocks) that tainted the previous values. ARMA analysis was introduced by Box and Jenkins [18] and they have identified three steps to model and forecast time series:

1) *Model Identification*: this step is performed to estimate a model structure by using two essential functions: the autocorrelation function (ACF) and the partial autocorrelation function (PACF).
2) *Parameter Estimation*: this step is performed for fitting the identified model to the observed data. This is achieved by determining the coefficients of the linear combination.
3) *Forecasting*: the final objective is to predict the future values of the time series based on the already observed data and the linear combination estimated at the second step. And so, $ARMA(p, q)$ model is defined as:

$$y_t = c + \mu + \sum_{i=1}^{p} \phi_i . y_{t-i} + \sum_{j=1}^{q} \theta_j . \epsilon_{t-j} + e_t \quad (1)$$

where:

- p, q non-negative integers; orders of the AR and MA processes respectively.
- ϕ_i, θ_j time-invariant coefficients off the AR and MA models respectively.
- μ expectation of Y (often assumed to be equal to zero) and c a constant (often omitted).
- e_t samples of white noise with mean zero and σ^2 and ϵ_t the white noise error terms.

To be considered for ARMA analysis, a time series must be stationary. To verify the stationarity two conditions must hold:

$$E(y_i) = \mu \text{ is constant and independent of instant } t \quad (2)$$

$$Cov(y_t, y_{t-j}) = \gamma^j \text{ only depends on time lag } j \quad (3)$$

The first condition means that for a stationary time series, the expectance of the studied random variable is independent is constant and independent from the period of study. The second condition means that the covariance between two values of the random variable depends only on the lag size between the two instances and it is free from the time instants values.

III. ORION Routing Protocol

A. Target Application

In this paper, we have considered a city-bus network in which the communicating nodes consist of buses, trams, cars and hotspots. The buses and trams are assumed to be "*regular*" mobile nodes, where the cars are assumed to be "*random*" mobile nodes and finally the hotspots and access points to be fixed nodes. The regular nodes move across the area along a certain trajectory (i.e. predetermined routes based on a trace file containing the set of points that describe the path), and many nodes may share the same route moving at both directions. Random nodes move freely across the urban area. In this scenario, all mobile nodes move with a non-constant speed. Indeed, for regular nodes (i.e. Buses/Trams), between every two stops, a certain speed is chosen and kept constant in this part of the route, another speed is chosen for the next route portion. All these speed values are chosen around a certain average value (e.g. 5, 10, 15 m/s). For random nodes, the mobility follows the random way-point mobility model with the same nodes' speed values. For inter-nodes contact analysis, we have used the same number of regular and random nodes for a total number of nodes of 100, 200, 300 nodes.

Each communicating node is assumed to be equipped with localization hardware. Consequently, we propose to utilize geographic addressing and achieve data packets forwarding in a greedy fashion based on distance and/or angle calculus. With geographic Routing/Greedy Forwarding, the network address of each node includes geographic coordinates (for example a node [A], at (x, y) will have the address ID-A.X.Y). Forwarding is then said greedy if we use the geographic information for choosing forwarder nodes (e.g., choosing the closest neighbor to the destination node as the next hop).

Following the Box and Jenkins steps to model the times series using ARMA model, we have conducted some simulations of our city-bus network. In these simulations, instead of using synthetic mobility models [19][20][21], we studied the two time series in pseudo-realistic environment mapped on a real city map, namely Bordeaux in France. Indeed, nodes were assumed embedded in transportation

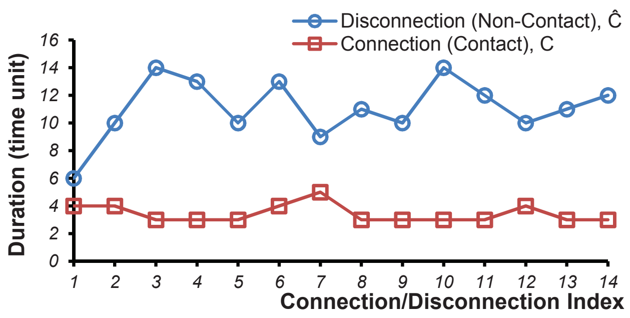


Fig. 1. Variation of C_i and $\bar{C}_i$ over time.

vehicles and were moving across the different routes over the city map. Moreover, nodes are having the same speeds and schedules as the actual vehicles. Cars and people were, in the other hand, moving freely (random way point) across the city and with different speeds.

In the following, we explain our methodology to perform the inter-node contact prediction.

B. Contact Behavior Analysis

For our analysis and in order to use efficiently time series, we propose to discrete the time into small periods of time Δt. In the description, we further denote $\{1\Delta t, 2\Delta t, ..., n\Delta t\}$ as the time instants $\{t_1, t_2, ..., t_n\}$.

In networks with intermittent connectivity, a node becomes aware of an eventual contact by the mean of periodically exchanged *HELLO* messages. Consequently, contact (*connection*) duration C with a certain node is the sum of consecutive periods of time Δt over which the node received at least one *HELLO* message from the other node (during this period, the node is called connected neighbor). Respectively, the duration of the non-contact (*disconnection*) $\bar{C}$ is the sum of consecutive periods of time over which the node did not receive any *HELLO* messages from the other node. The duration of a contact C and a non-contact $\bar{C}$ are two random variables. In order to study the contact behavior, we construct the two times series $\{C_t\}_{t\in\mathbb{N}}$ and $\{\bar{C}_t\}_{t\in\mathbb{N}}$, where C_t denotes the duration of the i^{th} contact (connection), and $\bar{C}_t$ represents the duration of the i^{th} non-contact (disconnection). $\mathbb{N}$ is the set of natural integers.

Examples of $\{C_t\}_{t\in\mathbb{N}}$ and $\{\bar{C}_t\}_{t\in\mathbb{N}}$ chronograms are shown in Fig. 1. To apply the Box-and-Jenkins approach, we had to verify the stationarity of the two time series. Thus, we run the stationarity test (see equations 2 and 3). The results showed that the two stationarity conditions hold for almost all the time series ($\sim$99%) obtained from the simulation (at the rate of two times series C_i and $\bar{C}_i$ by contacted node at each node). Consequently, C_i and $\bar{C}_i$ can be analyzed using ARMA. Based on this information, a node can predict the future value of the contact's duration (connection's duration), and also the future value of the non-contact duration (disconnections' duration). Consequently, the node will be able to predict when the next contact will be and for how long it will last. This knowledge will be extremely beneficial to perform routing decisions.

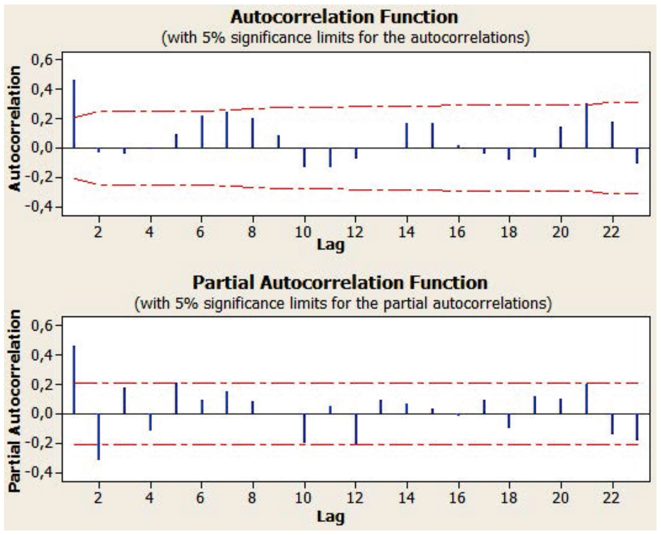


Fig. 2. Autocorrelation and Partial Autocorrelation functions' (ACF & $PACF$) plots for contact's duration.

C. ORION Contact Model Construction

After running the simulation, we extracted two time series, namely the $\{C_t\}_{t\in\mathbb{N}}$ and $\{\bar{C}_t\}_{t\in\mathbb{N}}$, for each frequently contacted node. It was interesting to notice that all the time series were quite similar in terms of pace, even if the mobile nodes were moving with different and non-constant speeds. The rest of this section describes the Box-and-Jenkins steps applied to model the proposed times series. We dubbed this model as "*ORION Contact Model*" as it is related to the targeted application scenario.

1) ***Step 1: ORION Contact Model Identification****:* We have used Minitab [22] to analyze the obtained time series. The autocorrelation function (ACF) and the partial autocorrelation function (PACF) are plotted in Fig. 2. According to the ACF and PACF plots, the results indicate that the best fitting model is the $ARMA(2,1)$ since PACF presents two significant peaks (i.e. this confirms the $AR(2)$ part), and the ACF presents one significant peak (i.e. this confirms the $MA(1)$ part). For all the analyzed time series, this model was the most frequent (78%), followed by $ARMA(2,2)$ with 7%, and$ARMA(3,2)$ with 5% and other models having higher degrees.

Based on these results, the ORION Contact Model obtained can be written as:

$$C_i = \mu + \phi_1 C_{i-1} + \phi_2 C_{i-2} + \theta_1 \epsilon_{i-1} + \epsilon_i \quad (4)$$

where:

- μ denotes the mean value C_i.
- ϕ_1, ϕ_2, θ_1 denote the ORION Contact Model parameters (ϕ_1, ϕ_2 are related to the autoregressive part and θ_1 is related to the moving average part of the process).
- $\epsilon_i, \epsilon_{i-1}$ are assumed to be independent, identically distributed random variables sampled from a normal distribution with zero mean $\epsilon_i \sim N(0, \sigma^2)$ where σ^2 is the variance.

2) ***Step 2: ORION Contact Model Parameters Estimation****:* The second step after identifying the orders of the ORION Model is to estimate its parameters. For the AR part, the parameters can be obtained by the Yule-Walker equations [23].

The principle of the Yule-Walker equation relies on the fact that there is a direct correspondence between the parameters ($\phi_i : i = 1..p$) and the covariance function of the process. This correspondence can be inverted to determine the parameters from the ACF which leads to the Yule-Walker equations:

$$\gamma_m = \sum_{k=1}^{p} \phi_k \gamma_{m-k} + \sigma_\epsilon^2 \delta_m \tag{5}$$

where $m = 0,..,p$ yielding $(p+1)$ equations. γ_m is the autocorrelation of Y. σ_ϵ is the standard-deviation of the input noise process, and the δ_m is the *Kronecker Delta* function. This equation is usually solved by representing it as a matrix, getting the following equation solving all ϕ.

$$\begin{bmatrix}\gamma_1\\ \gamma_2\\ \gamma_3\\ \vdots\end{bmatrix} = \begin{bmatrix}\gamma_0 & \gamma_{-1} & \gamma_{-2} & \cdots\\ \gamma_1 & \gamma_0 & \gamma_{-1} & \cdots\\ \gamma_2 & \gamma_1 & \gamma_0 & \cdots\\ \vdots & \vdots & \vdots & \ddots\end{bmatrix} \cdot \begin{bmatrix}\phi_1\\ \phi_2\\ \phi_3\\ \vdots\end{bmatrix} \tag{6}$$

This equation provides a way to estimate the $AR(p)$ parameters by replacing the theoretical covariance with estimated values. For the MA part, the single parameter is obtained by identification based on the estimated AR parameters and the last estimation error.

*3) **Step 3: Forecasting**:* Since the orders of the ORION Model are fixed in time due to the mobility model, the computational cost of resolving linear systems can be avoided by extracting generic formulas. The node will have to compute the model parameters based on simplified mathematical expressions. Moreover, since these formulas include only aggregated data (sums, means, standard-deviations, variances…), there is no need to store all the past data; only few values are maintained at each node. The following subsection explains the different steps to derive these simplified mathematical expressions and how they will be used for forecasting. Since the model is $ARMA(2,1)$, it is composed of two parts: $AR(2)$ and $MA(1)$. For $AR(2)$ process we have :

$$x_{t+1} = \phi_1 x_t + \phi_2 x_{t-1} + \xi_{t+1} \tag{7}$$

We multiply both sides by one lag value and take the expectation:

$$\langle x_t x_{t+1}\rangle = \phi_1 \langle x_t x_t\rangle + \phi_2 \langle x_t x_{t-1}\rangle + \langle x_t \xi_{t+1}\rangle \tag{8}$$

We eliminate the zero correlation forcing term $\langle x_t \xi_{t+1}\rangle$ and we divide by $N-1$:

$$\frac{\langle x_t x_{t+1}\rangle}{N-1} = \phi_1 \frac{\langle x_t x_t\rangle}{N-1} + \phi_2 \frac{\langle x_t x_{t-1}\rangle}{N-1} \tag{9}$$

We then have:

$$c_1 = \phi_1 c_0 + \phi_2 c_1 \text{ (c_i is the covariance of lag i)} \tag{10}$$

We divide both sides by c_0, we obtain the equation:

$$r_1 = \phi_1 r_0 + \phi_2 r_1 \text{ since } r_i = c_i / c_0 \tag{11}$$

Doing the same thing with lag 2, we find the second equation. Then we have:

$$\begin{cases} r_1 = \phi_1 r_0 + \phi_2 r_1 \\ r_2 = \phi_1 r_1 + \phi_2 r_0 \end{cases} \tag{12}$$

Leading to the Yule-Walker equations:

$$\underbrace{\begin{bmatrix}r_1\\ r_2\end{bmatrix}}_{r} = \underbrace{\begin{bmatrix}r_0 & r_1\\ r_1 & r_0\end{bmatrix}}_{R} \underbrace{\begin{bmatrix}\phi_1\\ \phi_2\end{bmatrix}}_{\Phi} \text{, and so } \Phi = R^{-1} r \tag{13}$$

Knowing that $r_0 = 1$:

$$\begin{bmatrix}\phi_1\\ \phi_2\end{bmatrix} = \frac{1}{1-r_1^2}\begin{bmatrix}1 & -r_1\\ -r_1 & 1\end{bmatrix}\begin{bmatrix}r_1\\ r_2\end{bmatrix} \tag{14}$$

Finally, we get:

$$\phi_1 = \frac{r_1(1-r_2)}{1-r_1^2} \text{ and } \phi_2 = \frac{r_2 - r_1^2}{1-r_1^2} \tag{15}$$

Considering the $MA(1)$ process:

$$x_t = \theta \xi_{t-1} + \xi_t \tag{16}$$

where ξ is the white noise $N(0,\sigma^2)$.

We multiply both sides by one lag value and take the expectation:

$$x_{t-1} x_t = (\theta \xi_{t-2} + \xi_{t-1})(\theta \xi_{t-1} + \xi_t) \tag{17}$$

$$\langle x_{t-1} x_t\rangle = \langle \theta^2 \xi_{t-2} \xi_{t-1}\rangle + \langle \theta \xi_{t-2} \xi_t\rangle + \langle \theta \xi_{t-1}^2\rangle + \langle \xi_{t-1} \xi_t\rangle \tag{18}$$

Knowing that:

$$\langle \xi_i \xi_j \rangle = \begin{cases} 0 & \text{if} \quad i \neq j \\ \sigma^2 & \text{if} \quad i = j \end{cases}$$

we eliminate the zero terms and we divide by $N-1$:

$$\frac{\langle x_{t-1} x_t\rangle}{N-1} = \frac{\langle \theta \xi_{t-1}^2\rangle}{N-1} \tag{19}$$

And we get:

$$c_1 = \theta \sigma^2 \tag{20}$$

By repeating the same operations for zero lag value:

$$x_t x_t = (\theta \xi_{t-1} + \xi_t)(\theta \xi_{t-1} + \xi_t) \tag{21}$$

We obtain:

$$c_0 = \theta^2 \sigma^2 + \sigma^2 \rightarrow c_0 = \sigma^2(1+\theta^2) \tag{22}$$

Knowing that, and from the two obtained equations, we have:

$$r_1 = \frac{c_1}{c_0} = \frac{\theta}{1+\theta^2} \tag{23}$$

By solving the quadratic equation we find :

$$\theta = \frac{1}{2r_1} \pm \sqrt{1 - 2r_1^2} \tag{24}$$

From the three formulas, we can see that the estimation of ORION Contact Model parameters relies on the estimated values of autocorrelation of order 1 and 2. (i.e. r_1 and r_2). Statistically, autocorrelation of order p (i.e. r_p) is estimated by the following expression:

$$r_p \cong \frac{E[(X_t - \mu)(X_{t-p} - \mu)]}{Var(X_t)} \cong \frac{E[(X_t - \mu)(X_{t-p} - \mu)]}{E[X^2] - E[X]^2} \tag{25}$$

```
Require: $v$ new value of the time series
Ensure: $x_{t+1}$ estimated future value of the time series
 1: $N++$;
 2: $Update\_Sliding\_Window(v, x_t, x_{t-1}, x_{t-2})$;
 3: $E_X = Compute\_New\_Average(E_X, N, x_t)$;
 4: $E_{X^2} = Compute\_New\_Average(E_{X^2}, N, x_t)$;
 5: $E_{X,X_1} = Compute\_New\_Average(E_{X,X_1}, N, x_t, x_{t-1})$;
 6: $E_{X,X_2} = Compute\_New\_Average(E_{X,X_2}, N, x_t, x_{t-2})$;
 7: $r_1 = Compute\_AutoCorrelation(E_X, E_{X^2}, E_{X,X_1})$;
 8: $r_2 = Compute\_AutoCorrelation(E_X, E_{X^2}, E_{X,X_2})$;
 9: $\phi_1 = Compute\_Phi\_1(r_1, r_2)$;
10: $\phi_2 = Compute\_Phi\_2(r_1, r_2)$;
11: $\theta = Compute\_Theta(r_1)$;
12: $x_{t+1} = Estimate\_Future\_Value\_ARMA21(\phi_1, \phi_2, \theta)$;
```

Fig. 3. Online model parameters update, and future value forecasting.

where:

$$E_X = E[X] = \mu = \frac{1}{N-1}\sum_{i=1}^{N} x_{t-i+1} \; ;$$

$$E_{X^2} = E[X^2] = \frac{1}{N-1}\sum_{i=1}^{N} x^2_{t-i+1} \; ;$$

$$E_{X,X_p} = \frac{1}{N-1}\sum_{i=1}^{N} (x_t - \mu)(x_{t-p} - \mu)$$

We can see that the calculus of the autocorrelation terms relies on aggregated terms (averages). Thus, these values can be incrementally computed without keeping the entire data set as follows:

$$E_{X_{(t)}} = \frac{(N-1)E_{X_{(t-1)}} + x_t}{N} \; ;$$

$$E_{X^2_{(t)}} = \frac{(N-1)E_{X^2_{(t-1)}} + x_t^2}{N} \; ;$$

$$E_{X,X_{p(t)}} = \frac{(N-1)E_{X,X_{p(t-1)}} + (x_t - \mu)(x_{t-p} - \mu)}{N}$$

Finally, in order to adaptively estimate the parameters of the $ARMA(2,1)$ process and in real-time, we only need to store the following variables (rather than the entire time series):

- N The number of the considered values so far $[0, t)$.
- x_t, x_{t-1}, x_{t-2} The last three values of the time series.
- E_X The last average of the values (at $t-1$).
- E_{X^2} The last average of the squared values (at $t-1$).
- E_{X,X_1}, E_{X,X_2} The last values of E_{X,X_1} and E_{X,X_2} respectively.

Since the data are evolving in time, we propose to compute these parameters in an incremental fashion. For two different instants $t_1 < t_2$ the estimated parameters are different because the estimation at t_1 takes into account the data up to t_1 (i.e. $[0, t_1]$) and similarly the estimation at t_2 takes into account all the data in $[0, t_2]$. This approach makes the estimation in real-time and more accurate as new data are collected. The process of model parameters' update and future value forecasting is explained with the pseudo code in Fig. 3. Fig. 4 presents, for an arbitrary time series extracted during the learning phase and which follows the identified contact model (i.e. $ARMA(2,1)$), a comparison between forecasting based on "offline" parameters estimation (i.e. the parameters are estimated considering the entire data set, then the future values are computed at each index using the obtained model) and forecasting based on "online" estimation (i.e. the parameters are estimated for each new observation, considering the available data so far). We can clearly see that the proposed online estimation is better than the offline estimation in predicting the future data.

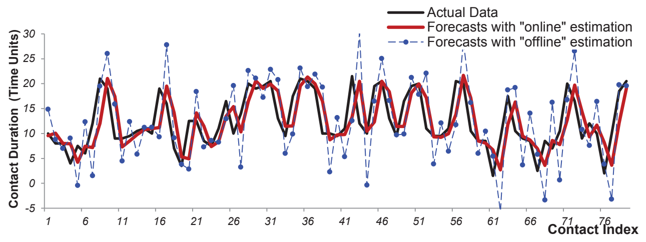


Fig. 4. Forecasting with ***"online"*** and ***"offline"*** parameter estimation.

D. Forwarding Algorithm

The forwarding algorithm used in ORION is based on three criteria: (1) in order to forward a packet in a greedy manner, a node will look for the closest connected neighbor to the destination, (2) if such a node is not available, the forwarder node will look for the most advancing connected neighbor toward the destination, and finally (3) if there is no such a node, the forwarder node will schedule the data packet for the best future connected neighbor. The best future connected neighbor is chosen according to an objective function. This function takes into account (1) the contact frequency, (2) the expected node trajectory, and (3) the period of time until the forecasted date of contact. Based on these contact's parameters, the objective function gave a score to each forecasted contact node. The node that has the highest score will be chosen for the packet forwarding. These three contact's parameters are considered as follows:

- Contact frequency (CF) is expressed as the number of contacts during a fixed period of time. The higher this value is, the higher the score of this future contact node becomes.
- Expected trajectory is expressed as the cosine of the oriented angle between the two vectors (1) the current direction of the forwarder node and (2) the expected direction of the future contact node. The expected direction of the future contact node is simply considered the opposite direction of its last known direction. The higher this value is (this value ranges from -1 "opposite direction" to $+1$ "same direction"), the higher the score of this future contact node becomes.
- Time to the next contact is expressed as the period of time until the next contact date. The lower this value is, the higher the score of this future contact node becomes.

Consequently, and after standardization of the three parameters, the score of a future contact node is given by the following function:

$$f_{obj}(N_i) = \alpha[CF_{N_i}]^* + \beta[cos(\vec{D}, -\vec{D}_{N_i})]^* + \gamma[T_{N_i} - T]^* \quad (26)$$

where:

- N_i: Neighbor i for which information has been collected and predicted.
- $CF_{N_i}, \vec{D}_{N_i}, T_{N_i}$ denote: (1) Contact frequency, (2) Last known direction vector, and (3) the predicted date of the next contact with neighbor N_i, respectively.
- $\vec{D}$ is the current vector towards the destination node, and T is the current time.
- α, β, γ: weight of each parameter in the objective function.

```
1: $nextHop \leftarrow Closest\_Connected\_Neighbor\_to\_Dest()$;
2: if ($nextHop \neq null$) then GOTO 8;
3: $nextHop \leftarrow Most\_Advancing\_Neighbor\_towards\_Dest()$;
4: if ($nextHop \neq null$) then GOTO 8;
5: $nextHop \leftarrow Best\_Future\_Connected\_Neighbor()$;
6: if ($nextHop \neq null$) then GOTO 9;
7: $Store_Packet()$; end.
8: $Send_Packet_to(nextHop)$; end.
9: $Schedule_Packet_for(nextHop)$; end.
```

Fig. 5. Pseudo code for ORION forwarding algorithm

Packet_Queue: a data structure where all the packets to be sent are stored.
Connected_Neighbors_Set(CN): the set of all neighbors that are currently in contact with this node.
Estimated_Neighbors_Set(EN): the set of all estimated neighbors, i.e. the nodes whose contacts we have a historical data about.
Forwarder_Node(FN): the address of the next hop.
f_{obj}: This function gives a score to a neighbor based on its next contact date, contact frequency and expected moving direction.

```
1: if($Packet\_Queue$ is not Empty)
2:   $pk \leftarrow Packet\_Queue.pop()$;
3:   if($pk.NextHop = null$) // packet was not scheduled
4:     $FN \leftarrow arg_N \max\{Distance(N_{pos}; pk_{dest_{pos}}\}; N \in CN$
5:     if($FN \neq null$)
6:       $Send\_Packet(pk, FN)$;
7:     else // Most advancing neighbor towards the destination
8:       $FN \leftarrow arg_N \max\{Distance(N_{old\_pos}; pk_{dest\_pos})$
9:             $-Distance(N_{new\_pos}; pk_{dest\_pos}\}; N \in CN$
10:      if($FN \neq null$)
11:        $Send\_Packet(pk, FN)$;
12:      else // best future contact
13:        $FN \leftarrow arg_N \max\{f_{opt}(N_{contact\_frequency};$
14:                  $N_{next\_direction}; N_{next\_contact\_date})\}$;
15:                  $N \in EN$
16:        $pk.NextHop \leftarrow FN$;
17:        $Packet\_Queue.push(pk)$;
18:      end if
19:    end if
20:  else // packet was scheduled for a certain node
21:    if($pk.NextHop \in CN$) // is the predicted node connected?
22:      $Send\_Packet(pk, FN)$;
23:    else // the predicted neighbor is not connected
24:      $pk.NextHop \leftarrow null$; // unscheduling the packet
25:      $Packet\_Queue.push_back(pk)$; // storing the packet
26:      $Forward\_Packets()$; // Starting over
27:    end if
28:  end if
29: end if
```

Fig. 6. A detailed description of the ORION forwarding algorithm.

- $[]^*$ denotes the standardized value of the parameter.

This objective function can be tuned by giving different weights to the three parameters. The value of each coefficient (α, β, γ) highly depends on the considered scenario. For a simple objective function, these coefficients were all set to one. A pseudo code of the algorithm is shown in Fig. 5. A detailed version is shown in Fig. 6.

IV. Performance Evaluation

A. Simulation Environment

For simulation purpose, we have considered a homogeneous wireless mobile network in which mobile nodes are deployed through an area of $14,5km$ x $11km$ with nodes' transmission range up to $200m$. Network nodes are composed of nodes moving according to bus/tram trace file (50% of the nodes) or random way-point model (50% of the nodes). Two nodes are selected randomly at the beginning of the simulation to act as source and destination. The source sends periodically data packets to the destination. The simulation is run for 1800 seconds (letting sufficient start-up time for PRoPHET, i.e. 600 seconds). To demonstrate and evaluate the performance of ORION, we used OMNeT++ 4 [24]. As a comparison term, we use the PRoPHET protocol. PRoPHET is similar to ORION in a way that both protocols keep track about the encountered nodes. However, ORION gathers more information than PRoPHET in order to enhance the routing performances.

For simulation, we considered different network topologies by varying (1) the number of nodes (i.e., 100, 200, and 300 nodes) and (2) the nodes speed (i.e., $5m/s$, $10m/s$, $15m/s$ and $20m/s$). For each topology, we measured various parameters: (1) the average hop count from the source to the destination, (2) the packets delivery ratio, (3) the first packet arrival, and finally (4) the average end-to-end delay. Due to space limitation, results relative to the 200 nodes topology are not shown.

We also prove that ORION routing protocol does not require more space memory to perform store-carry-and-forward routing by measuring the message queue occupancy during all the simulation period.

1) Simulation Results Discussion:

a) ***Hop Count (HC)***: from Fig. 7, we can clearly see that ORION delivers packets along fewer hops than PRoPHET and this is the case for all the two topologies and with all nodes speeds. This is achieved thanks to the twofold forwarding strategies of ORION protocol (store-and-forward and store-carry-and-forward) while PRoPHET is just a store-and-forward protocol.

b) ***Packet Success Ratio(PSR)***: Since the selection of the next forwarder node in ORION is based on three criteria (i.e. closest neighbor, most advancing neighbor, and the best future contact) rather than just one criterion (Probability of delivery success) in the case of PRoPHET, the successful node selection in ORION prevents packets from being lost; i.e., sent to nodes that cannot forward them. This allows ORION to successfully deliver more packets than PRoPHET as shown in Fig. 8. From this figure, we can also notice that the impact of nodes' speed is more important in ORION than in PRoPHET. With a high speed, the accuracy of the ARMA predictions is affected since the contacts' durations will be at the same scale as prediction error margin. Thus, packets loss will be more frequent. However, the packet delivery ratio is still higher than PRoPHET's.

c) ***First Packet Arrival (FPA) and End-to-End Transmission Delay (EED)***: Because of the greedy nature of ORION, packets will always choose either the shortest, the "earliest" next hop or a next hop moving toward the destination, making packets arrive more quickly at the destination node (Fig. 9) and experiencing shorter end-to-end delay (Fig. 10)

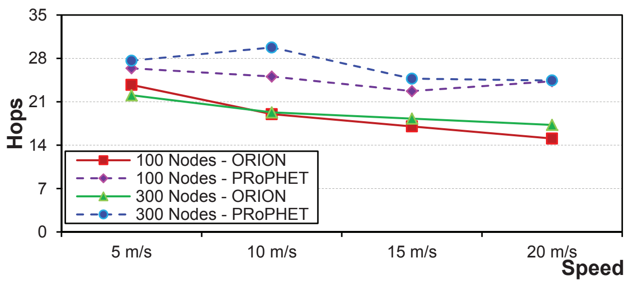


Fig. 7. Average hop count in 100 and 300 nodes topologies with variant speed.

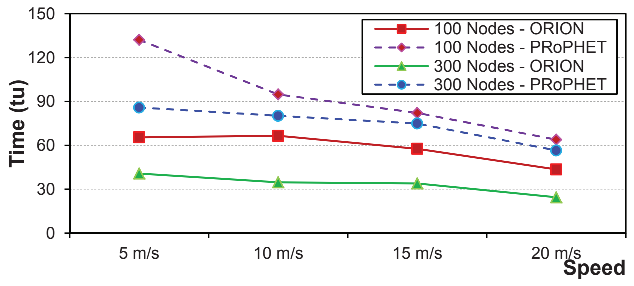


Fig. 9. First Packet Arrival in 100 and 300 nodes topologies with variant speed.

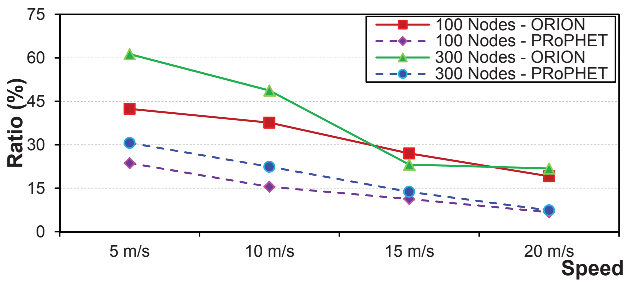


Fig. 8. Packet Success Ratio in 100 and 300 nodes topologies with variant speed.

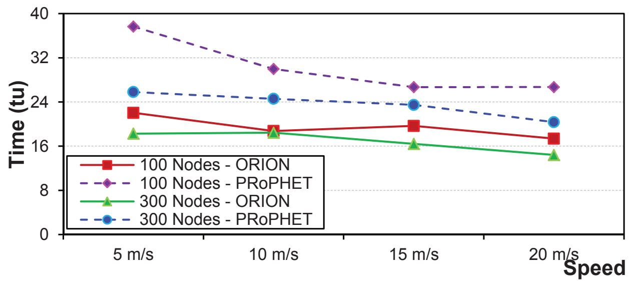


Fig. 10. Average E2E Delay in 100 and 300 nodes topologies with variant speed.

compared to PRoPHET where the next hop is chosen based on only its success delivery probability.

d) ***Bundle Queue Occupancy***: From Fig. 11 we can clearly see that ORION routing protocol does not require excessive memory space for message queue in order to achieve store-and-forward routing. In our simulations, and in small network topology (100 nodes), the average message queue occupancy is high (40 messages in case of 5 m/s node mobility speed) when the node mobility speed is slow. Due to the rarity of the contacts, nodes are obliged to keep messages for a long period of time. However, with higher node mobility speeds, the average queue occupancy decreases accordingly (average of 25 messages in the case of 20 m/s node mobility speed). Furthermore, as in the case of PRoPHET, packets experience higher hop count than ORION, the average queue size is higher than in the case of ORION.

V. Conclusion

In this paper, we have described a new routing protocol, dubbed as ORION, which is suitable for mobile delay tolerant networks. Since the network dynamics are often not so random such as in city-wide inter-hotspot network interconnected through taxis, buses and vehicles, the contacts between two communicating nodes can be analyzed and, moreover, predicted. ORION routing protocol is based on greedy geographic forwarding and contacts predictions, switching between store-and-forward and store-carry-and-forward strategies in such a way, that packet forwarding is always optimal. ORION uses ARMA model online parameter estimation to predict future contacts due to its outstanding fit to this kind of network dynamics. Simulation results show that ORION routing protocol outperforms PRoPHET in terms of different metrics such as first packet arrival delay, end-to-end transmission delay, hop count, and packet delivery ratio. Moreover, ORION routing protocol does not require excessive memory space to achieve efficient delay-tolerant routing.

References

[1] F. Kevin, *et al.*, "A delay-tolerant network architecture for challenged Internets," in *Proc. 2003 ACM SIGCOMM.*.

[2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-tolerant networking architecture," RFC4838, Apr. 2007.

[3] "The openmesh project," 2013. Available: http://www.openmeshproject.org/

[4] S. Medjiah and T. Ahmed, "Orion routing protocol for delay tolerant networks," in *Proc. 2011 IEEE Int. Conf. Commun.*, pp. 1–6.

[5] A. Vahdat, D. Becker, *et al.*, "Epidemic routing for partially connected ad hoc networks," Tech. Rep. CS-200006, Duke University, 2000.

[6] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proc. 2005 ACM SIGCOMM Workshop Delay-Tolerant Netw.*, pp. 252–259.

[7] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," in *Service Assurance with Partial and Intermittent Resources*. Springer, 2004, pp. 239–254.

[8] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and focus: efficient mobility-assisted routing for heterogeneous and correlated mobility," in *Proc. 2007 IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, pp. 79–85.

[9] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: routing for vehicle-based disruption-tolerant networks," in *Proc. 2006 IEEE INFOCOM*, vol. 6, pp. 1–11.

[10] A. Balasubramanian, B. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem," in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, 2007, vol. 37, pp. 373–384.

[11] S. Basu, A. Mukherjee, and S. Klivansky, "Time series models for Internet traffic," in *Proc. 1996 IEEE INFOCOM*, vol. 2, pp. 611–620.

[12] C. Liu, K. Wu, and M. Tsao, "Energy efficient information collection with the ARIMA model in wireless sensor networks," in *Proc. 2005 IEEE Global Telecommun. Conf.*, vol. 5, pp. 2271–2274.

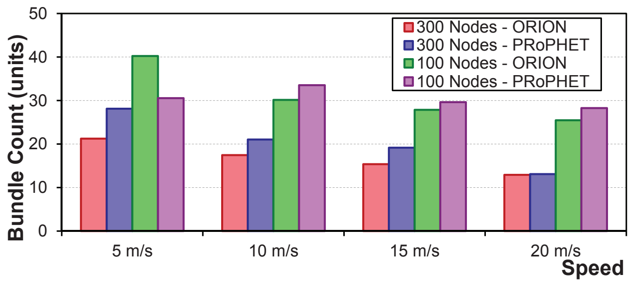


Fig. 11. Average message queue occupancy in 100 and 300 nodes topologies with variant speed.

[13] P. J. Brockwell and R. A. Davis, *Time Series: Theory and Methods*. Springer, 2009.

[14] D. Herbert, G. Modelo-Howard, C. Perez-Toro, and S. Bagchi, "Fault tolerant arima-based aggregation of data in sensor networks," in *Proc. 2007 IEEE Int. Conf. Dependable Syst. Netw.*

[15] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, 2002.

[16] A. Banerjee, K. Majumder, P. Dutta, and K. Mon-Dal, "Implementation of the behavior and structure of the multihop mobile environment as a pushdown automata and birth-and-death based statistical model," in *Proc. 2004 Mobile Comput. Netw.*, pp. 45–51.

[17] J. P. Singh and P. Dutta, "Temporal behavior analysis of mobile ad hoc network with different mobility patterns," in *Proc. 2009 ACM Int. Conf. Advances Comput., Commun. Control*, pp. 696–702.

[18] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. Holden Day, 1970.

[19] D. N. Alparslan and K. Sohraby, "A generalized random mobility model for wireless ad hoc networks and its analysis: one-dimensional case," *IEEE/ACM Trans. Netw.*, vol. 15, no. 3, pp. 602–615, 2007.

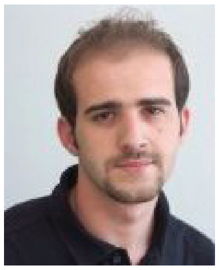
[20] E. Hyytia, P. Lassila, and J. Virtamo, "A Markovian waypoint mobility model with application to hotspot modeling," in *Proc. 2006 IEEE Int. Conf. Commun.*, vol. 3, pp. 979–986.

[21] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. Mobile Comput.*, vol. 2, no. 5, pp. 483–502, 2002.

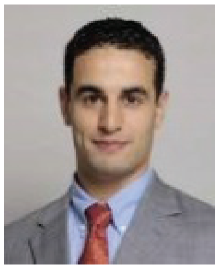
[22] I. Minitab, "Minitab statistical software," 2013. Available: http://www.minitab.com/

[23] M. B. Priestley, *Spectral Analysis and Time Series*. Academic Press, 1981.

[24] A. Varga, et al., "The OMNET++ discrete event simulation system," in *Proc. 2001 European Simulation Multiconf.*, vol. 9, p. 185.



Samir Medjiah is a Teaching and Research Assistant (ATER) at IPB (Institut Polytechnique de Bordeaux) in the ENSEIRB-MATMECA school of engineering. He conducts his research at the CNRS-LaBRI Lab, UMR 5800, at the Univ. de Bordeaux. Samir Medjiah received his M.S in computer science with distinction from the National Institute of Computer Science (ESI ex-INI), Algiers, Algeria, in 2009, and his Ph.D. with high honors from LaBRI/University of Bordeaux-1, France, in 2012. His main research interests are multimedia communications over challenged networks including routing, transport and congestion control of multimedia streams in wireless multimedia sensor networks, delay-tolerant networks, satellite and space communications, and various overlay networks. He has won national competitions in the field of mathematics (Mathematical Olympiad 2004) and computer science (Microsoft Imagine Cup 2008). Dr. Medjiah is an IEEE/ComSoc member.



Tarik Taleb is an IEEE Communications Society (ComSoc) Distinguished Lecturer and a senior member of IEEE. He is currently a Senior Researcher and 3GPP Standards Expert at NEC Europe Ltd., Heidelberg, Germany. Prior to his current position and until Mar. 2009, he worked as an assistant professor at the Graduate School of Information Sciences, Tohoku University, Japan, in a lab fully funded by KDDI, the second largest network operator in Japan. From Oct. 2005 through Mar. 2006, he worked as a research fellow with the Intelligent Cosmos Research Institute, Sendai, Japan. He received his B.E. degree in information engineering with distinction, and his M.Sc. and Ph.D. degrees in information sciences from GSIS, Tohoku Univ., in 2001, 2003, and 2005, respectively.

Dr. Taleb's research interests lie in the fields of architectural enhancements to mobile core networks (particularly 3GPP's), mobile cloud networking, mobile multimedia streaming, congestion control protocols, handoff and mobility management, inter-vehicular communications, and social media networking. Dr. Taleb is also directly engaged in the development and standardization of the Evolved Packet System as a member of 3GPP's System Architecture working group. Dr. Taleb is a board member of the IEEE Communications Society Standardization Program Development Board. As an attempt to bridge the gap between academia and industry, Dr. Taleb founded and has been the general chair of the "IEEE Workshop on Telecommunications Standards: from Research to Standards," a successful event that was awarded "best workshop award" by the IEEE Communications Society (ComSoC). Dr. Taleb is/was on the editorial board of *IEEE Wireless Communications Magazine*, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE COMMUNICATIONS SURVEYS & TUTORIALS, and a number of Wiley journals. He serves as vice-chair of the Wireless Communications Technical Committee, the largest in IEEE ComSoC. He also served as Secretary and then Vice Chair of the Satellite and Space Communications Technical Committee of IEEE ComSoc (2006–2010). He has been on the technical program committee of different IEEE conferences, including Globecom, ICC, and WCNC, and chaired some of their symposia.

Dr. Taleb is the recipient of the 2009 IEEE ComSoc Asia-Pacific Best Young Researcher award (June 2009), the 2008 TELECOM System Technology Award from the Telecommunications Advancement Foundation (March 2008), the 2007 Funai Foundation Science Promotion Award (April 2007), the 2006 IEEE Computer Society Japan Chapter Young Author Award (December 2006), the Niwa Yasujirou Memorial Award (February 2005), and the Young Researcher's Encouragement Award from the Japan chapter of the IEEE Vehicular Technology Society (VTS) (October 2003). Some of Dr. Taleb's research work has been also awarded best paper awards at prestigious conferences.

Toufik Ahmed is a Professor at IPB (Institut Polytechnique de Bordeaux) in the ENSEIRB-MATMECA school of engineering. He conducts his research at the CNRS LaBRI Lab, UMR 5800, at the University of Bordeaux 1. Toufik Ahmed received the B.Sc. in computer engineering, with high honors, from the I.N.I (National Institute of Computer Science) Algiers, Algeria, in 1999, and M.Sc. and a Ph.D degrees in computer science from the University of Versailles, France, in 2000 and 2003, respectively. In November 2008, he obtained the HDR degree (Habilitation á Diriger des Recherches) at the University of Bordeaux -1 on adaptive streaming and control of video quality of service over wired/wireless IP networks and P2P architectures. Toufik Ahmed was a visiting scientist at the School of Computer Science of the University of Waterloo in 2002 and a research fellow at the PRiSM laboratory of the University of Versailles until 2004. His main research activities concern quality of service for multimedia wired and wireless networks, end-to-end signaling protocols, P2P networks, and wireless sensors networks. His work on quality of service and video delivering has led to many publications in major journals and conferences.