Bio-inspired Proximity Discovery and Synchronization for D2D Communications

Shih-Lung Chao, Hsin-Ying Lee, Ching-Chun Chou, and Hung-Yu Wei, Member, IEEE

Abstract—In this paper, a distributed mechanism for application-aware proximity services (ProSe) in device-to-device (D2D) communications is presented. The method, which is derived from the bio-inspired firefly algorithm, can achieve proximity discovery and synchronization at one time. To be precise, this mechanism not only enables neighbor discovery and service discovery simultaneously, but achieves synchronization in physical communication timing and service interests in the meanwhile. However, the basic firefly algorithm is not perfectly suitable for larger scale systems such as LTE-A D2D. In most network topologies, the property that each node may not be able to hear all its neighbors makes the basic version fail. Thus, we further propose the firefly spanning tree (FST) algorithm which can solve the topology-dependent divergence problems.

Index Terms—D2D, bio-inspired algorithm, application-aware services, proximity discovery, synchronization.

I. INTRODUCTION

Device-to-device (D2D) communication is an emerging technology. This technology enables a wireless communication device to connect to another device directly. Recently, 3GPP started discussing proximity based services (ProSe)[1][2] for D2D communication in LTE. Proximity discovery is a key component for D2D communication. Take public safety application as an example, devices are usually distributed in a broad area as shown in Fig. 1. Infrastructure nodes may be unavailable due to the damage from a disaster. When some devices are outside the base station’s (BS) coverage, a distributed mechanism is required for devices to achieve proximity discovery and synchronization. The proximity discovery in general can be categorized into two different problems, physical communication[8] and application level discovery. The integration of two proximities is desired because the signaling procedures can be reduced. For example, physical-level proximity discovery requires signal exchange among devices. Application-level proximity discovery requires the devices to find other devices with the same interests. Thus, integration of proximity discovery in view of physical communication and application is needed.

Previous works applied this sync method in different manners. Wener-Allen et al. implemented decentralized RFA (reachback firefly algorithm) on TinyOS-based motes, and provided theoretic improvement[4]. Tyrrell et al. studied an exquisite design[5] and its application[6]. Lucarelli and Wang described the general model, and the convergent condition[7]. Nevertheless, in all conventional work above, the topology (physical layer radio connectivity topology) model is always idealized.

Our LTE-A D2D simulator is constructed based on the LTE simulator [9]. It includes a link level simulator and a system level simulator. As observed in simulations, the network topology significantly affects the convergence. As shown in Fig. 2, if we naively apply the basic firefly algorithm in randomly generated topologies, a substantial ratio of systems are even unable to achieve the state sync. In current systems, like D2D communications in LTE-A, the significant growth of involved devices makes the complexity of the network marking increase. This implies that the basic firefly algorithm is no longer suitable to emerging communications. Thus, it is imperative to create a new replacement which is adaptable to all kinds of topology. In this paper, a distributed topology-adaptive algorithm named firefly spanning tree (FST) is proposed. FST is an algorithm based on the basic firefly algorithm and a spanning tree algorithm, it is able to transform an divergent graph into a convergent graph as shown in Fig. 1. In simulation, the algorithm not only preserves all the
benefits from the original one, additionally it outperforms other common sync methods [13] under the LTE-A D2D circumstance.

II. BASIC FIREFLY ALGORITHM

In the real world, synchronization between one another is one of the the largest problems that a swarm of fireflies have to face every day. This is a totally self-organized synchronization problem, resembling the one we face in communications. Surprisingly, by simply applying their innate flashes, the tough problem has an elegant solution.

Basically, the synchronous flashing progress can be modelled as the behaviour of a population of identical integrate-and-fire oscillators. It can be expressed as a phase function \( \phi_i(t) \) which is integrated from zero to a certain threshold \( \phi_{th} \). When \( \phi_{th} \) is reached, the oscillator flashes, or “fires.” After firing, \( \phi_i(t) \) is reset to zero. If not coupled with others, i.e. no firing signal is detected, the oscillator will naturally fire with a period equal to \( T \). This may be described as

\[
\frac{d\phi_i(t)}{dt} = \alpha \phi_i + \beta \tau_i(t) + \Delta \phi_j(t) + \epsilon_i \quad \text{if} \quad \phi_i(t) < \phi_{th}\]

\[
\phi_i(t) \rightarrow \phi_{th} \quad \text{when} \quad \phi_i(t) \geq \phi_{th} \quad \text{and} \quad \phi_{th} \quad \text{is reset to zero. If not coupled with others, i.e. no firing signal is detected, the oscillator will naturally fire with a period equal to} \quad T. \]

When coupled to others, an oscillator is receptive to the firing signal of its neighbours. When a given oscillator fires, it pulls the others up by a fixed amount \( \epsilon \), or brings them to \( \phi_{th} \), whichever is less, i.e., \( \phi_i(t) = \phi_{th} \Rightarrow \forall j \neq i : \phi_j(t^+) = \min(\phi_{th}, \phi_j(t) + \epsilon) \). If the threshold \( \phi_{th} \) is normalized to 1, when a node \( j \) fires at \( t = \tau_j \), the above equation can be transformed into a piecewise linear function,

\[
\phi_i(\tau_j) + \Delta \phi(\phi_i(\tau_j)) = \min(\alpha \phi_i(\tau_j) + \beta, 1) \quad \text{with} \quad \alpha = e^\epsilon \quad \text{and} \quad \beta = \frac{\phi_{th}}{\epsilon}
\]

\[
\Delta \phi = \frac{\phi_{th}}{\epsilon}, \quad \Delta \phi \quad \text{and} \quad \beta \quad \text{denote} \quad \text{the phase increment function and the dissipation factor, respectively.}
\]

It has already been proven that, provided that certain constraints (e.g. \( \alpha > 1, \beta > 0 \), and full meshed) on the coupling between entities are met, for an arbitrary number of entities and independent of the initial condition, the network always synchronizes. Plenty of aspects such as the effect of propagation delay, channel attenuation, and noise have also been addressed in the literature [5] [6]. Optimal selection of the parameters has also been completely discussed in some solid works. Thus, the scope of this paper will not cover the above issues again.

III. PROXIMITY DISCOVERY AND SYNCHRONIZATION

The basic concept of firefly algorithm can be applied to D2D networks as shown in Fig. 3. In this paper, we proposed a distributed mechanism to achieve proximity discovery and synchronization. This mechanism enables neighbour discovery and service discovery simultaneously. In addition, it also achieves synchronization in physical communication timing and service interests in our application.

The mechanism achieves proximity discovery by sending and detecting proximity signals (PSs) among devices as shown in Fig. 4. Each device possesses a counter and broadcasts PSs periodically. The counter value increases by time and a threshold is set for the counter. Devices detecting the PSs will increase the counter value by a fixed rate. Once the counter reaches the threshold, the device will broadcast the PS, and then reset the counter back to the initial value like zero. The PS is broadcast to the neighbouring devices, and those devices increase their counter values accordingly. This process of PS broadcast and detection will continue until all the devices achieve synchronization in the period of signal broadcast. Note that the rate of increment and threshold may be different for different applications/services/groups. Take LTE-A system as an example, predetermined RACH codes (or other CDMA codes) or preambles can be used to transmit the PS in the proposed scheme. Different codes indicate different application interests. This implies that a device may transmit multiple codes to indicate its interest in these multiple applications. Transmission in different radio transmission opportunity (e.g. RACH resource) can also be used to indicate different application interest.

After synchronization is reached within the group, several communication activities can be done. For example, the mechanism might be served as a keep-alive signaling method. It can also be directly applied to activate data session between devices. For instance, the PS may use two different codes (e.g. a pair RACH code). One code indicates standby and continue to the server as the synchronization/keep-alive purpose. The
other code triggers other events. We further assume that user equipments are full-duplex devices which are capable of simultaneously transmitting and receiving proximity signals. LTE-A RACH preambles are OFDM symbols, and different preambles may be transmitted in parallel without inter-group interference. However, intra-group proximity signal interference may still arise due to collision or misalignment. Such interference will influence the counter value increment. That is, devices may only detect 1 preamble while more than two devices are sending the proximity signal. Firefly algorithm still holds in this case.

For the purpose of reducing the signal processing in both physical and application levels of proximity discovery, we introduce the concept of firefly synchronization into D2D communications. The properties of the algorithm are quite suitable to an idealized system, like a full meshed network. However, in real systems, an idealized topology is not a common case. To our best knowledge, currently there is no such research that focuses on applying similar algorithms on any randomly generated topology.

IV. FIREFLY SPANNING TREE

A D2D network can be formulated into a graph $G(V, E)$, where vertices $V$ are independent devices and edges $E$ are communication links. Links can be weighted by the strength of communication links. Links can be weighted by the strength of $k$-node tree $\phi_k$. One new connection is established between $L(V, P)$ and any of the nodes $L(P, \phi)$ can be achieved by inserting a new node $L$ to a $k$-node tree $T$.

Theorem 1: For any acyclic graph (i.e., tree, connected graph without simple cycles), by applying the firefly algorithm, the synchronization of nodes is always achieved and sustained.

Proof: Mathematical induction can be used to prove that the above statement in the theorem always holds.

The basis: Show that the statement holds for the node number $n = 2$. Based on [3], when there are only two nodes (i.e., full meshed network with two nodes), the system can be synchronized. The inductive step: Show that if the trees with $k$ nodes (i.e. $n = k$) hold, then any acyclic structures with $n = k + 1$ nodes will also hold.

A new tree $T'$ with $k + 1$ nodes can be achieved by inserting a new node $L$ to a $k$-node tree $T$. One new connection is established between $L$ and any of the nodes $P$ in $T$. When $\phi(P) = \phi_\alpha(P)$, $\forall P \in T$, the derivation can be divided into three conditions based on the impact of phase shift $\phi(P)$ from $\phi_\beta(P)$ after the insertion of $L$

1) No phase shift occurs (i.e. $\phi(P) = \phi_{\alpha}(P)$): this means that the synchronizing situation of $T'$ is equal to $T$. That is, the convergence is immediately completed.

2) Phase shift occurs (i.e. $\phi(P) \neq \phi_{\alpha}(P)$), and $P$ reaches the threshold: $P$ receives a PS from $T = \{P\}$ at $t_0$, which means the entering of $L$ makes $P$ temporarily unsynchronized from $T \backslash \{P\}$. If the PS makes $P$ saturate (i.e. $\alpha_{\phi}(P) + \beta \geq 1$), $L$ will be also lifted to saturation by $P$'s PS since $\phi(P) = \phi_{\alpha}(P)$... As a result, the condition of $T'$ is equal to $T$.

3) Phase shift occurs, but $P$ does not reach the threshold: If the PS at $t_0$ does not make $P$ saturate (i.e. $\phi(P) = \phi_{\alpha}(P) + \beta < 1$), $P$ and $L$ are temporarily unsynchronized. At the moment just before $P'$s next saturation, say $t_1$, the phase of $L$ is $\phi_{\alpha}(P) + \beta < 1$. After that, the PS from $P$ is received by $L$, we have $\phi_{\alpha}(P) + \beta < 1$. Since $\alpha > 1$ and $\alpha_{\phi}(P) + \beta < 1$, we have $\alpha(1 - \alpha_{\phi}(P) + \beta - 1) = (1 - \alpha)(\alpha_{\phi}(P) + \beta - 1)$. Using the fact that $\alpha > 1$, we get $\phi_{\alpha}(P) + \beta < 1$. As a result, the condition of $T'$ is equal to $T$.

Since both the basis and the inductive step have been performed, by mathematical induction, the statement always holds.

V. NUMERICAL RESULTS

In this section, system level simulations are carried out. Referring to [12] and [9], we construct a LTE-A D2D network simulator. The parameters are given in Table I. As shown in Fig. 5, the performance results are presented for different network sizes. Each case is evaluated in average convergence time within 1000 trials (randomly generated topologies with fixed nodes). Note that the convergence time here is the measure of how fast the devices reach the state of complete synchronization. The average convergence time is the mean of the 1000 trials. Since currently there is no other related work which is under the framework of LTE-A, the compared
The same time, in both physical and application level. Some is able to achieve proximity discovery and synchronization at network ProSe, especially in LTE-A systems. The mechanism messages exchanged.

FST even narrowly defeated CS-MNS in terms of number of facts, when in large networks (e.g. the case of 500 nodes), While FST largely outperforms CS-MNS in convergence time, the trade-off of fast convergence is the number of exchanging messages. However, we can see that the deal is quite worthwhile. In Fig. 6, we can see that the proposed FST outperforms CS-MNS in mutual network synchronization (CS-MNS) [13]. As we can investigate, we only found the algorithm of clock-sampling or unable to converge in random topologies. After thorough selection, we found that the algorithm of clock-sampling mutual network synchronization (CS-MNS) [13]. As we can see in Fig. 5, the proposed FST outperforms CS-MNS in any size network. Also, we can observe that FST is even more efficient in larger systems due to the linearithmic time complexity.

Another performance evaluation is the exchanging messages during the converging progress. In Fig. 6, we can see that the trade-off of fast convergence is the number of exchanging messages. However, we can see that the deal is quite worthwhile. While FST largely outperforms CS-MNS in convergence time, the discrepancy of the two algorithms is always subtle. In fact, when in large networks (e.g. the case of 500 nodes), FST even narrowly defeated CS-MNS in terms of number of messages exchanged.

VI. CONCLUSION

In this paper, we proposed a distributed mechanism for D2D network ProSe, especially in LTE-A systems. The mechanism is able to achieve proximity discovery and synchronization at the same time, in both physical and application level. Some essential operations can also be accomplished at the same time. In addition, a practical algorithm called FST is also given for resolving the problems caused by different topologies. Finally, the numerical results on LTE-A simulator show that the algorithm outperforms the other existing algorithms.

REFERENCES