Fast, self-adaptive timing-synchronisation algorithm for 802.11 MANET

H. Tanaka, O. Masugata, D. Ohta, A. Hasegawa and P. Davis

In multihop ad hoc networks that use conventional IEEE 802.11, long transient resynchronisation states are often generated when multiple IBSSs merge. A simple modification of the conventional timing synchronisation method is proposed to reduce such synchronisation bottlenecks. When the proposed modification is applied, a self-adaptive synchronisation ability is observed in simulations, which makes resynchronisation times much shorter and reduces energy consumption.

Introduction: The ad hoc mode of the IEEE 802.11 standard (hereinafter abbreviated as 802.11) supports the formation of independent basic service sets (IBSSs) in the absence of an access point. Based on the 802.11 media access control (MAC) protocol [1], large mobile ad hoc networks (MANET) are now being developed. In addition, as the nodes (mobile PCs or PDAs) become smaller, more efficient, power-saving control is required, and the transmission ranges of nodes should be limited if possible. In such networks, the timing synchronisation function (TSF) is one of the essential integral components, since timing synchronisation is required for frequency-hopping spread spectrum and synchronous power-saving control. In this 802.11 TSF, each node transmits a beacon that carries timing information only when the node is elected through the contention of neighbouring nodes.

This contention is based on a random process and is supposed to provide a fair election of the beacon transmission node. Most conventional studies on this timing synchronization problem have assumed closed, isolated networks within a single IBSS [2, 3]. In contrast to such ideal static cases, timing synchronisation in dynamic environments is currently being considered (e.g. for the cases of merging two IBSSs [4] and of resynchronisation in large, multihop networks [5]). In this Letter, we consider MANET in such dynamic situations, and propose a simple modification of the 802.11 TSF. Interestingly, this simple modification greatly reduces the transient resynchronisation time, especially for larger networks with hundreds of nodes that maintain a relatively small total number of awake nodes. The reason for the efficiency of the present method is also investigated by systematic simulation and analysis, revealing a kind of self-organised adaptive process that enables fast resynchronisation.

Synchronisation bottlenecks: We consider the simple, but generic examples shown in Fig. 1. Systematic simulations of the synchronisation process were carried out by changing the number of nodes as well as the transmission range (R) of nodes.

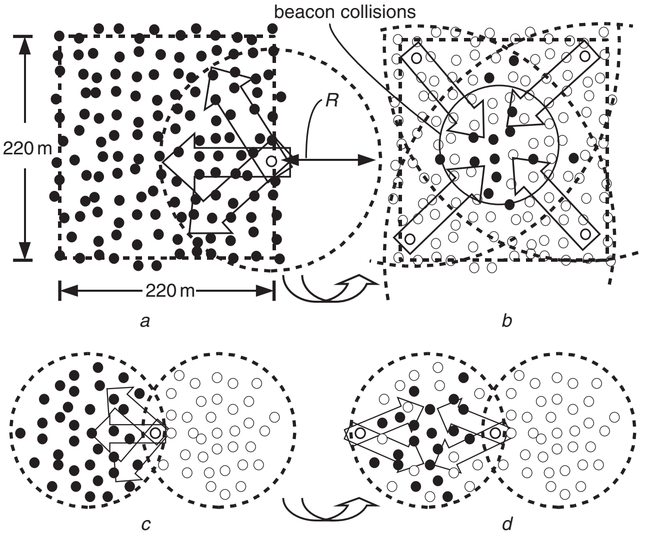

Fig. 1 *Beacon propagation in multihop networks*

a Initial state
b Beacon collisions during synchronisation bottleneck
c Initial state for merging two-cell case
d Beacon collisions during bottleneck

As shown in Fig. 1*a*, 143 synchronised nodes (denoted by •) belonging to the same IBSS are randomly placed in a 220 × 220 m grid, and a node (denoted by ○) belonging to another IBSS with a later TSF timing is then moved to a certain position. Even though this node is fixed, its transmission range can temporally become wider. This has the same effect as the above case of node movement. Both the above situations initiate a resynchronisation process until all the nodes (144 nodes in this example) eventually become resynchronised to the later timing (○). Since this resynchronisation time depends on the transmission range of the nodes as well as on the position of the introduced node, we systematically change the transmission range (R) from 24 to 360 m (by 4 m intervals) and 10 000 trials are carried out for each R with a randomly positioned joining node. To consider the worst-case scenario of the synchronisation process, the power-saving timing of the introduced node is initially set 180° out of phase to that of the original nodes in these simulations. Namely, the introduced node is awake when most of (but not all of) the other nodes are in the power-saving state. Note that only awake nodes can receive beacons, and a node remains in its awake state for one beacon period (~100 ms: awake/power-saving cycle) directly after the node has transmitted a beacon by beacon contention (BC). This is the reason why the resynchronisation process becomes complicated and often requires a long time.

As a limiting case of Fig. 1*a*, we also consider the case of merging two IBSSs (cells), as shown schematically in Fig. 1*c*. This situation arises when two small rooms (in which any two nodes can directly communicate) are joined by a short corridor.

To clarify the essential mechanism of the beacon propagation, the present simulations assume that node movement can be neglected in the resynchronisation process. In addition, the following simplifying assumptions are made: (1) the transmission range is time-constant and uniform over the nodes; (2) the transmission delay of the beacons carrying the TSF timing is negligible compared to a single beacon slot (50 μs in this case); and (3) small mismatches in clock frequencies are negligible compared to the timescale of the awake/power-saving cycle (beacon period), as a first approximation. Later, we consider factors (2) and (3) to be non-negligible and analyse their effects.

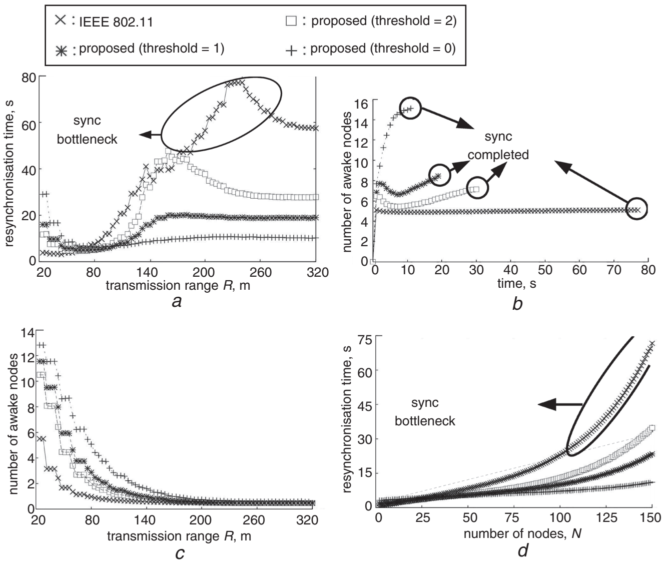

Fig. 2 *Simulation data*

a Average resynchronisation time with respect to transmission range, for case of Fig. 1*a*
b Number of awake nodes during resynchronisation process
c Number of awake nodes in static state
d Average resynchronisation time against total number of nodes (N) in an IBSS, for case of Fig. 1*c*

The simulation results obtained are summarised in Fig. 2, in which the averaged resynchronisation times are plotted against the transmission ranges (data plotted with × in Fig. 2*a*). This data set is obtained using a regular array of 144 nodes. For other node configurations, including the totally random case and the randomly disturbed array case, we have observed a pattern similar to that shown in Fig. 2*a*. For the case in which the range of R is larger than 140 m, as shown in Fig. 2*a*, a synchronisation bottleneck emerges. This long resynchronisation time can be explained by a spatio-temporal pattern of beacon propagations and frequent beacon collisions, as shown in Fig. 1*b* (for details, see [5]). In other words, this bottleneck comes from the combination of the multihop network structure and the beacon contention (BC) mechanism in the 802.11 TSF. We analysed this BC process in detail when the bottleneck emerged and found that the fairness in beacon transmission is lost because nodes near the boundary of the network can transmit beacons more frequently than other nodes. This is because the 802.11 TSF adopts a local BC, which is no longer fair when the local node density is not uniform over the network.

Simple method to reduce synchronisation bottleneck: Since loss of fair beacon transmission causes synchronisation bottleneck, an intuitive method of bottleneck reduction is to regulate the beacon transmissions of nodes near the boundary. In the present simulations, nodes with larger BT values near the boundary are selected, whereas other nodes are selected with smaller BT values. This is because BC becomes less competitive near the boundary. Therefore, we should examine what happens when each node regulates the beacon transmission for the case in which the BT is larger than a certain threshold. To examine this idea concretely and ensure that it is logically consistent with the 802.11 TSF, the following assumptions are introduced: (i) for a node selected by BC, the beacon transmission is cancelled if its BT exceeds a given threshold; and (ii) such a node maintains its awake state over one beacon period directly after the cancellation of the beacon transmission.

In Fig. 2, we compare three numerical data sets (plotted by $+$, $*$, and □ symbols) obtained using the proposed method to that obtained using the original 802.11 TSF ($\times$). These three data sets are obtained by averaging over 10 000 trials when the BT threshold is set to one, two and three beacon slots, respectively. It is noted that, even in the worst instances of these three data, the synchronisation time is shorter than that of the 802.11 case. In Fig. 2*a* (and Fig. 2*d*), the data plotted by the $+$ symbols show a performance that is several times better than the 802.11 TSF (plotted by the $\times$ symbols), except for certain short transmission ranges ($R < 60$ m). In these simulations, the number of awake nodes is also analysed systematically both for the resynchronisation process (Fig. 2*b*) and for the static states after the resynchronisation has been completed (Fig. 2*c*), where R is set to 240 m. In Fig. 2*b*, all three data sets ($+$, $*$, and □) have a larger number of awake nodes (averaging approximately five awake nodes) than that for the 802.11 ($\times$). However, the total number of awake nodes required until the resynchronisation is complete becomes smaller in these three cases, because the resynchronisation time is much shorter than in the 802.11 case. In Fig. 2*c*, the number of awake nodes is approximately the same ($\sim$ one node on average) for R larger than 180 m. In addition, in the proposed method, the total number of beacon transmissions is always less than that of the 802.11 TSF, which is a direct consequence of assumptions (i) and (ii). Thus, the total energy consumption in the proposed method is expected to be smaller than that of the 802.11 TSF.

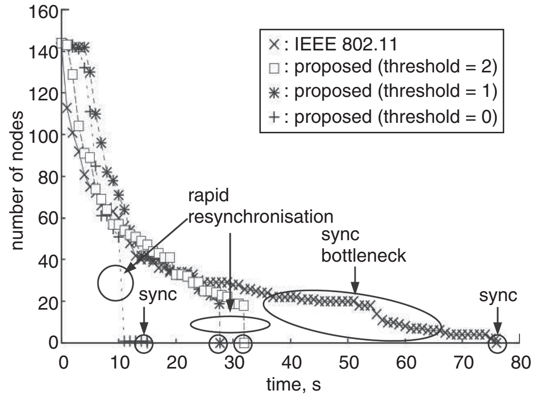

Fig. 3 *Resynchronisation process*

Bottleneck reduction mechanism: In this Section, we analyse the reduction of the synchronisation bottleneck in the proposed method, and the observed efficiency is explained. Fig. 3 shows the number of remaining, non-resynchronised nodes (● in Fig. 1) against time for the four cases ($+$, $*$, □ and $\times$) corresponding to the data shown in Fig. 2. Initially, there are 143 nodes, and the number of nodes decreases monotonically as the synchronisation proceeds. In the 802.11 TSF a long bottleneck always emerges. In contrast, in the proposed method, such a bottleneck is reduced ($+$; $*$, and □ symbols in Fig. 3), where a rapid resynchronisation emerges in several dozen nodes simultaneously, leading to total resynchronisation. This rapid resynchronisation, which is a characteristic feature of the proposed method, is a byproduct of modifications (i) and (ii) above. A brief explanation of this mechanism is as follows: (*a*) as the resynchronisation proceeds, the density of the non-resynchronised nodes is gradually lowered; (*b*) for such a low node density, BC becomes less competitive and modifications (i) and (ii) often cause some of the nodes to remain awake simultaneously; and (*c*) in this case, if some of the nodes within another IBSS (○) transmit beacons, then they are received by these non-resynchronised nodes, whereupon resynchronisation is completed in a single beacon cycle.

Thus, the bottleneck reduction in the proposed method is realised by a kind of self-adaptive, rapid synchronisation process, which is achieved through a simple modification of the 802.11 TSF. The above observation is also verified through a modified simulation protocol, as per [3], including simplification factors (2) and (3) mentioned above. We expect the proposed self-adaptive algorithm to be beneficial also in the activity-scheduling problem in flooding algorithms [6], which is now under investigation.

Acknowledgments: This work was partially supported by NICT and the Ministry of Education, Science, Sports and Culture, Grant-in-Aid, 18560370 (2006), and stimulated by the activities of the AdHoc Wireless Network Consortium.

4 May 2006
Electronics Letters online no: 20061390
doi: 10.1049/el:20061390

H. Tanaka, O. Masugata and D. Ohta (*Department of Electronic Engineering, The University of Electro-Communications (UEC), 1-5-1 Choufugaoka, Choufu-shi, Tokyo 182-8585, Japan*)

A. Hasegawa and P. Davis (*ATR Adaptive Communications Research Laboratories, 2-2-2 Hikaridai, Keihanna Science City, Kyoto 619-0288, Japan*)

References

1 IEEE Std. 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, http://standards.ieee.org/getieee802/download/802.11-1999.pdf

2 Huang, L., and Lai, T.-H.: 'On the scalability of IEEE 802.11 ad hoc networks'. Proc. 2002 ACM Int. Symp. on Mobile Ad Hoc Networking and Computing, pp. 173–182

3 Sheu, J.-P., Chao, C.-M., and Sun, C.-W.: 'A clock synchronization algorithm for multi-hop wireless ad hoc networks'. Proc. 2004 Int. Conf. on Distributed Computing Systems, pp. 574–581

4 Hasegawa, A., and Davis, P.: 'Long coalescence times for ad hoc cells using 802.11 MAC protocol'. Proc. 2004 IEICE Spring Society Meeting, p. 407

5 Ohta, D., Tanaka, H., Hasegawa, A., and Davis, P.: 'Timing-sync issues in multi-hop networks using 802.11 MAC Protocol (2)'. Proc. 2005 IEICE Fall Society Meeting, p. 460

6 Stojmenovic, I., and Wu, J.: 'Broadcasting and activity scheduling in ad hoc networks' *in* 'Mobile ad hoc networking' (IEEE/Wiley, 2004), pp. 205–229