July 2011

Periodically updating the content in mobile Ad hoc network with secured synchronization

R. Banu Priya
banupriya.ar@gmail.com

V. Komath
komathivenkatesh@yahoo.com

A. Karthika
karthika_ashokan@yahoo.com

Follow this and additional works at: <https://www.interscience.in/ijssan>

Part of the [Digital Communications and Networking Commons](#), and the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Banu Priya, R.; Komath, V.; and Karthika, A. (2011) "Periodically updating the content in mobile Ad hoc network with secured synchronization," *International Journal of Smart Sensor and Adhoc Network*: Vol. 1 : Iss. 1 , Article 4.

DOI: 10.47893/IJSSAN.2011.1003

Available at: <https://www.interscience.in/ijssan/vol1/iss1/4>

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Smart Sensor and Adhoc Network by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

Periodically updating the content in mobile Ad hoc network with secured synchronization

R.Banu Priya¹, V.Komathi², A.Karthika³

Email Id: banupriya.ar@gmail.com¹, komathivenkatesh@yahoo.com², karthika_ashokan@yahoo.com³

Abstract: The content and presence updates were used to transmit updates to nearby nodes over an ad hoc network using duty cycles; our algorithms synchronize the transmissions of nodes periodically. By this approach, nodes can save battery power without missing updates from their neighbours by switching off their network cards. The algorithms and protocols used here provides more security.

INTRODUCTION:

The mobile phones that have rich media and wireless networking capabilities has ushered in a new paradigm in mobile computing with new emerging social behaviours. New enabling technologies now allow users to search, locate, download, and share dynamically created content with friends and family from their mobile devices. With ad hoc networking capabilities in mobile devices, we are beginning to see the above trend shift from wide-area communities of users to dense local area social situations such a shift presents opportunities to design proximity aware systems that deliver novel social experiences. For example, fans watching a football game can automatically share pictures taken on their mobile phones with each other, while commenting/rating pictures being taken around them.

Designing systems for ad hoc environments presents several interesting research challenges, including the difficult problem of providing scalable, energy efficient presence and content updates. To keep information fresh in such environments, the distribution mechanisms have to focus on frequent, small metadata updates rather than large infrequent payloads, which could also be a cause of significant battery drain from a mobile device.

EXISTING SYSTEM:

- Nodes don't Know the transmission time of their neighbours
- single malicious user can easily disrupt network stability and synchronization

PROPOSED SYSTEM:

- We propose a Content and Presence Multicast Protocol (CPMP) which nodes use to send updates to their neighbours.
- We propose to address the energy efficiency problem by synchronizing the transmission times of all the nodes in the system.
- We devise a rating based algorithm (RBA) that rates neighbours based on the consistency of their behaviour. By favouring well-behaved nodes in the synchronization process, we show that RBA quickly stabilizes the synchronization process and reduces the number of lost updates by 85 percent.

MODULES:

1. CPMP Protocol
2. Weight Based Synchronization
3. Rating based algorithm

CPMP PROTOCOL:

We designed the CPMP to support social content consumption experiences. CPMP provides a framework for periodically communicating information about what content is currently being consumed and what content is being sought for future consumption at each participating node. Our goal was to make the protocol efficient and scalable while including features intended to support synchronization of presence message transmissions. CPMP messages are transmitted periodically to inform nearby devices of updated content presence information using IP multicast.

CPMP headers have the following format "CPMP; device identifier; TX;" containing a TX field specifying the number of seconds in which to expect a new CPMP message from the node specified by device identifier. Note that this does not include any redundant transmissions of the current

message—if retransmission is used to improve reliability, each retransmission must update the TX field so that it accurately reflects the delay until the new message will be transmitted. By transmitting CPMP messages at approximately the same time CPMP messages are expected, an implementation can avoid powering on the wireless LAN radio for transmissions—essentially piggybacking CPMP transmission with reception. Ideally, all participating nodes will use this technique simultaneously, resulting in synchronization of CPMP activity.

CPMP packets may also include an HTTP port number on which a host may be listening for HTTP requests to support related services (such as content transfer). CPMP messages are transmitted via multicast UDP/IP frames. Each message is encapsulated in a single UDP/IP message and transmitted to the CPMP multicast address. In order to avoid IP fragmentation, the protocol requires that implementations limit message sizes to the link layer maximum transmission unit.

WEIGHT BASED SYNCHRONIZATION:

We first describe an algorithm that uses the size of synchronization clusters as a catalyst for synchronization. We call the algorithm WBS—weight based synchronization. As mentioned previously, at the end of each active interval, a node uses the slotArray structure to decide its next transmission time. The slotArray structure has s entries, one for each slot of the next (sleep) interval. The node has to choose one of these slots, called winner slot, and synchronize with it. That is, the node has to advertise the time of its next transmission (its TX value in the CPMP update packet) such that the update packet will be placed into that winner slot by its neighbours. WBS requires each node to locally maintain a variable monitoring the size of the cluster of synchronization which contains the node. The variable is called the weight of the node/cluster. Initially, the weight of each node is 1. Each node includes its weight in all its CPMP updates. Certainly, nodes cannot maintain globally accurate weights. Instead, each node needs to use only local knowledge—extracted from packets received from neighbours—to update the value of this variable. During each active interval, a node keeps track of the largest cluster weight seen among all the received packets, including its own. At the end of the active interval, the node chooses the winner slot to be the one storing the packet advertising the largest weight. That is, each node chooses to synchronize with the largest synchronization cluster in its vicinity. When a node joins a cluster, the weight of the cluster increases, thus increasing the cluster's potential to attract even more members.

RATING BASED ALGORITHM:

RBA requires each node to build statistics of its neighbours' transmission stability: a neighbour that consistently sends its CPMP updates in the same slot is considered more stable than a node that regularly changes slots. In effect, each node is building a rating value for each neighbour. Ratings are used only locally and are not propagated to neighbours. After an active interval, a node will synchronize with the slot in which the packet from the highest rated neighbour has been placed. Thus, neighbours with lower ratings (e.g., newly seen or unstable) will have little chance to influence the synchronization process. The ratings are built as follows: When a node A receives a packet from a neighbour B, if the packet's slot coincides with B's previous transmission slot, B's rating is incremented. If not, B's rating is dropped to zero. Note that B's rating is set to zero also if B misses one transmission or if B is a newly seen neighbour. In order to prevent impersonation attacks, we also extend the CPMP to include minimal overhead authentication information. While this additional data does not prove that the node is who it claims to be, it allows its neighbours to make a reasoning of the form: "this packet was sent by the same device that has sent a similar packet one interval ago."

SOFTWARE REQUIREMENTS:

Operating System : Windows XP/Vista/7
Language : C#.NET (WPF)
Documentation Tool: Ms word 2003
DotNet Framework : V3.5

HARDWARE SPECIFICATION:

RAM : 256 MB and above
Processor : P4 and above
Hard Disk : 40 GB and above

REFERENCES:

- [1]. P. Persson, J. Blom, and Y. Jung, "DigiDress: A Field Trial of an Expressive Social Proximity Application," *Proc. Ubiquitous Computing (UbiComp)*, 2005.
- [2]. The Nokia Sensor Project, <http://europe.nokia.com/A4144923>, 2010.
- [3]. H. Caituiro-Monge, K. Almeroth, and M. Del Mar Alvarez-Rohena, "Friend Relay: A Resource Sharing Framework for Mobile Wireless Devices," *Proc. ACM Int'l Workshop Wireless Mobile Applications and Services on WLAN Hotspots*, Sept. 2006.

- [4]. B. Carbunar, M. Pearce, S. Mohapatra, L. Rittle, and V. Vasudevan, "Byzantine Resilient Synchronization for Content and Presence Updates in Manets," *Proc. IEEE Int'l Conf. Network Protocols (ICNP)*, 2008.
- [5]. W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *Proc. IEEE INFOCOM*, 2002.
- [6]. T.V. Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," *Proc. ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys)*, Nov. 2003.