Performance Analysis of AeroRP with Ground Station Advertisements*

Hemanth Narra, Egemen K. Çetinkaya, and James P.G. Sterbenz
Information and Telecommunication Technology Center
Department of Electrical Engineering and Computer Science
The University of Kansas, Lawrence, KS 66045, USA
{hemanth, ekc, jpgs}@ittc.ku.edu

ABSTRACT

Data transmission in highly-dynamic airborne networks is challenging, thus domain-specific routing protocols are necessary. AeroRP is one such protocol designed to operate in disruption-tolerant networks. Its multi-modal design enables it to operate in varying mission requirements. In this paper, we discuss the operation of various AeroRP modes and analyse their performance using the ns-3 network simulator. We compare the performance of beacon, beaconless, and ground station (GS) modes of AeroRP. The simulation results show the advantages of having a domain-specific routing protocol and also highlight the importance of ground station advertisements to disseminate geolocation information in discovering routes.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*routing protocols*

General Terms

Algorithms, Design, Performance

Keywords

geographic routing, AeroRP, aeronautical networks, ns-3 simulation, disruption-tolerant network (DTN)

1. INTRODUCTION

The iNET (Integrated Network Enhanced Telemetry) program has recognised the requirement for multihop transmission of data in emerging highly-dynamic airborne networks [1]. Multihop routing is necessary to cope with the limited spectrum and increased requirement for bandwidth [9]. An example scenario is shown in Figure 1 [9], in which airborne nodes (ANs) move at velocities of up to Mach 3.5 and move towards each other with relative velocities of about Mach 7.0. In this scenario, ANs route application data to a ground station (GS) via another AN or through a relay node (RN) in a multihop fashion. However, these high velocities lead to frequent link breaks.

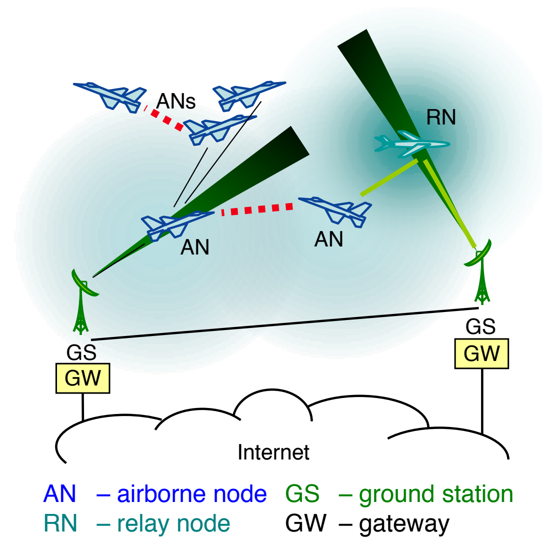

Figure 1: Aeronautical network architecture

The Aeronautical Routing Protocol (AeroRP) is a domain-specific routing protocol that was designed for highly dynamic airborne networks [4, 7, 8, 9]. AeroRP uses the node's current or predicted geolocation information for discovering routes. Neighbour discovery in AeroRP is multi-modal, in which various modes can be used to discover neighbours depending upon the mission needs including stealth requirements. One of those neighbour discovery modes is GS update mode. In this paper we present the design and implementation of GS update mode of AeroRP in the ns-3 network simulator [6]. We compare the results of different modes of AeroRP neighbor discovery modes and show that a priori knowledge of the network topology in this highly-dynamic environment significantly improves the performance of the routing.

The rest of the paper is organised as follows: In Section 2 we present an overview of AeroRP and AeroNP. The AeroRP design with GS advertisements is explained in Section 3. We present the simulation results comparing different modes of AeroRP in Section 4. Finally, conclusions are presented in Section 5.

*Distribution A - Approved for public release; distribution is unlimited.

2. BACKGROUND

AeroRP is a geographic routing protocol designed for highly-dynamic airborne networks [4]. Contrary to other MANET routing protocols that discover end-to-end paths, AeroRP makes only per-hop routing decisions. This is reasonable as the nodes in the network move at very high velocities often leading to breakage of links after an end-to-end path has been determined. AeroRP can operate in various modes based on the mission requirements [9]. AeroRP can discovery neighbours in beacon and beaconless modes. Moreover, depending on the mission requirements, geolocation information of ANs may or may not be present in the AeroNP (Aeronautical Network Protocol) header. For example in a given mission, ANs can be programmed to be in stealth mode so that they are not making their location available to other ANs.

2.1 Operational Aspects of AeroRP

The operation of AeroRP can be divided into two phases. The first phase of operation is the *neighbour discovery* phase. In this phase, an AN gathers as much information as it can about the network topology in the following ways: *Active snooping* is a mechanism in which the nodes snoop packets that are being exchanged among other nodes, extract the location information from them, and build or update their topology tables. In *beacon mode*, Hello beacons are transmitted periodically; this ensures that its neighbouring ANs are aware of its presence. *Ground station advertisements* are optional messages transmitted by the GS during some missions having a pre-determined flight plan. GS advertisement messages are broadcasted periodically and are exchanged among all the ANs.

These AeroRP modes affect the various neighbour discovery processes. In beaconless mode the hello messages are not sent by any of the ANs. Therefore, neighbour discovery relies on *overhearing* the packets in the medium. Depending on the mission needs, AeroRP messages can have location information of ANs. If the AeroRP routing messages include location information, then the ANs can determine the network topology. Otherwise, without the location information, ANs can only be aware of their neighbours. If the GS is involved in the neighbour discovery phase, it can broadcast two types of advertisement messages: GSLink or GSTopology. GSLink messages carry only the active or predicted links among all the nodes, while GSTopology messages carry the geolocation information.

The second phase of AeroRP operation is *data forwarding*. In this phase, the sender node determines the next hop to forward a packet by using the topology table built in the *neighbour-discovery* phase. The *time-to-intercept* (TTI) metric is used in determining this next hop neighbour [7]. TTI is calculated for every node from the topology table as:

$$\text{TTI} = \frac{\Delta d - R}{s_{\text{d}}}$$

whereΔ d is the Euclidean distance between the current location and a predicted location of a node based on the recorded location coordinates and velocity components from the table, R is the common transmission range of all the nodes, and s_{d} is the recorded speed component. The assumption made here is that the nodes move at a constant speed during the interval for which we calculate the TTI value. TTI gives the estimate of time taken by the neighbours to reach within the transmission range of the destination. The neighbour with the lowest TTI value is chosen as the next hop neighbour and packets are forwarded to this neighbour.

2.2 AeroNP

AeroNP is an IP-compatible network protocol specifically designed for highly-dynamic airborne networks. Application systems and other devices on ANs are IP -based. In addition to replicating the services provided by IP, AeroNP provides QoS (quality of service), flow control, and error detection to the AeroTP transport layer [9]. QoS is provided by tagging packets based on priorities; AeroNP provides four levels of priorities. The AeroRP control packets are always given the highest priority. Mission specific command and control data is given higher priority compared to application data. The packets tagged with a particular priority are queued in specific buffer classes. The flow-control mechanism is accomplished by implementing a cross-layering mechanism with the iNET TDMA MAC layer. The congestion indicator (CI) field in AeroNP specifies the congestion level at a node. If a node identifies that the MAC buffer is full, it increases its congestion indicator value in the AeroNP header. Neighbouring nodes do not forward packets to a node with high CI value, unless it is the destination. The wireless medium is error-prone leading to packet corruption, and detecting these errors at the destination and waiting for the source to resend the packet increases the end-to-end delay. The AeroNP corruption indicator and HEC-CRC (header error check – cyclic redundancy code) fields provide error detection. Depending on the mission requirements geolocation information can be included in the AeroNP header. We designate the AeroNP header with the geolocation information as the *extended header*, whereas the AeroNP header without the geolocation information as is referred to as the *basic header*.

3. GS ADVERTISEMENTS MODE

Introducing GS updates to the AeroRP routing protocol adds an additional mechanism for neighbour discovery in AeroRP. The GS tracks movements of all the ANs and broadcasts either their TopologyInformation or LinkInformation to other ANs in the network [5].

3.1 GS Update Mechanism

A GS broadcasts updates for all the ANs periodically over a time-interval called the periodicUpdateInterval. Depending on velocities of the ANs and the frequency at which they change direction, the periodicUpdateInterval is set. The frequency at which the GS sends these updates will affect the protocol's performance significantly. If the periodicUpdateInterval is low, the GS broadcasts updates frequently resulting in increased control overhead. On the contrary, if the periodicUpdateInterval is high, the GS will broadcast updates rarely resulting in the ANs not having up-to-date information about the other ANs. Velocities of all the ANs are unlikely to be uniform and some of the ANs may change their direction more frequently than others. Therefore, sending updates of all the ANs for every periodicUpdateInterval alone is not sufficient. There is a need for a mechanism in which the GS can broadcast updates in-between the periodicUpdateIntervals as well, called the trigger update mechanism. Whenever the GS identifies a change in direction of any AN, it will immediately broadcast the changed location and velocity information of the AN. Highly dynamic changes result in the GS sending trigger updates more frequently, which leads to more control overhead and increased packet collisions in the network. To avoid this, trigger updates over a triggerUpdateInterval are aggregated into a single update and sent together. On the other hand, the GS verifies if a trigger update can be cancelled and the change be sent in the next periodic update. If the time duration to the next periodic update is less than the triggerUpdateInterval, then the trigger update is not sent.

To make sure the GS advertisements do not violate the maximum transmission unit (MTU) of 1500 B set by the MAC layer, they are fragmented into multiple packets. Each packet is uniquely identified by the time at which this packet was originated and by a 16-bit fragment number.

3.2 Types of GS Advertisements

The GS sends two types of advertisements: GSTopology and GSLink. The GSTopology and GSLink advertisements are explained in the following sections.

3.2.1 Topology Advertisements

Topology information of all the nodes is advertised by the GS using GSTopology advertisements. This advertisement carries a TypeHeader over multiple GSTopologyHeaders containing the geolocation coordinates and the velocity components in x, y, and z directions. A GS also maintains a topology table just like other ANs. The difference is that the ANs fill their table on receiving AeroRP updates, whereas the GS topology table is assumed to have updated information of the entire network based on the predetermined flight plans. The GS uses this information to send out topology updates. It sets the EnableBroadcast flag to *true* so that this message can be rebroadcasted among all the ANs. Depending on the update mechanism used, the GS sends out updates for all the ANs or only for the ANs with changed information since the last update. The GS creates multiple topology headers, one for each node and adds them to a packet. It then creates a single TypeHeader, populates the necessary fields and adds it on top of the topology headers. This packet is then broadcast in the network. Every AN receiving the packet will process the TypeHeader, identify the type of headers present in the packet, and based on the header length it will process each header separately. Figure 2 shows the header format for GSTopologyHeader.

node id	reserved
x-coordinate	x-velocity
y-coordinate	y-velocity
z-coordinate	z-velocity
start time	end time

Figure 2: Packet format for GSTopologyHeader

Topology updates have an option for adding start time and end time. These fields are used when an AN is flying in a pre-determined path. When the start time is populated, it is interpreted as the AN is located at the location coordinates specified by the topology header. The end time field is the time from which the ANs should stop using this location information. This field can also be set to next periodicUpdateInterval or the predicted time after which the AN might go out of the network.

3.2.2 Link Advertisements

GSLink advertisements carry multiple GSLinkHeaders for all the links formed among nodes in the network. Figure 3 shows the header format for GSLinkHeader. Each GSLinkHeader will carry two 16-bit node-id addresses of the ANs forming the link, the start and the expire times for that link, and the link cost. The start and expire times are calculated based on node's geolocation and velocity information. A link is said to be established between two nodes if the euclidean distance between the two is less than their transmission range. The assumption here is that all nodes have the same transmission range. Based on the nodes geolocation coordinates the euclidean distance is calculated. The link expire time is predicted based on the geolocation and velocity components. The expire time is increased until the euclidean distance between the new predicted locations of the two nodes is greater than their transmission ranges. Thus the GS calculates this information for all the nodes in the network. If there are n nodes in a network, considering the best case scenario in which every node is connected to every other node, the total number of links are $n \times (n-1)/2$.

node1 id	node2 id
link start time	
link end time	
link cost	

Figure 3: Packet format for GSLinkHeader

3.3 Processing of AeroRP Updates

AeroRP uses a protocol id of 251 and works in conjunction with the AeroNP network protocol. The AeroNP protocol on receiving any packet from the MAC layer will identify an AeroRP packet by looking at the protocol id field in the AeroNPHeader, and if appropriate will deliver it to the AeroRP routing protocol along with the extracted AeroNPHeader. A GS node does not require any AeroRP control information as it always keeps track of all the nodes in the network. Each AN examines the sourceTimestamp field present in the AeroNPHeader. It compares it with the stored, last-received timestamp value for that neighbouring node in its topology table. If the received timestamp is newer, the protocol will update the node's topology table with the geolocation information present in the AeroNPHeader. AeroRP then compares its own GSTimestamp value with that of the neighbouring node's GSTimestamp value. If it identifies that the neighbour node does not have a newer update from the GS, it unicasts those updates to the neighbour node. This mechanism will make sure that every node will be as consistent as possible. AeroRP uses three types of control messages to broadcast its updates as explained in Section 2. They are hello messages, GSLink advertisements, and GSTopology advertisements. A hello message does not have any information to be used by the AeroRP routing protocol. It is only used to inform a node's neighbours of its presence. Upon receiving a GSTopology advertisement, AeroRP unpacks all the GSTopologyHeaders one by one and updates the topology table entry for that node. GSLink advertisements will also be processed the same way as a GSTopology advertisement by extracting the GSLinkHeaders. However, after all the GSLinkHeaders are updated in the topology table, the protocol will determine if the Dijkstra shortest-path algorithm should be run based on the changed link costs. If there is a change in the links, AeroRP runs the Dijkstra algorithm, otherwise the algorithm is scheduled to run at the time determined by the creation of a new link based on the predicted geocoordinates and velocity components. A timer schedules the Dijkstra algorithm by keeping track of the new link formations or breakages based on the information from the topology table.

4. SIMULATION RESULTS

We performed the simulations over an area of 150 km^2. All the simulations were averaged over 10 runs with each simulation running for 1000 s and plotted with 95% confidence interval bars. Simulations were performed with varying node densities ranging from 5 to 60 nodes. The communication model is peer-to-peer with as many flows as the number of nodes in the network. All the ANs are configured to send 1 packet/s. Using this lower packet rate, we can correctly evaluate the performance of the protocol by minimising the probability of congestion at MAC layer. We use the ns-3.10 to perform these simulations. The On-Off application in ns-3 is used to generate CBR (constant-bit rate) traffic. The 802.11b MAC is used over the Friis propagation loss model to limit the transmission ranges of nodes. The transmit power is set to 50 dBm to achieve a 27800 m (15 nautical mi) transmission range. The mobility model is 3D Gauss-Markov [3] with node velocities ranging from 50 – 1200 m/s. Setting α between zero and one allows us to tune the model with degrees of memory and variation. We set the α to 0.85 to have some predictability in the mobility of the nodes, while avoiding abrupt AN direction changes. Ferrying of packets is enabled by default in AeroRP using a drop-tail queue implementation. The maximum buffer size is set to 400 and the maximum buffering time is set to 30 s. Any packet that reaches its buffer expiry time is purged. Active snooping is enabled by default.

4.1 Performance Metrics

The performance metrics for the evaluation of AeroRP are packet delivery ratio (PDR), routing overhead, and delay. PDR is the number of packets received divided by the number of packets sent by the application. Routing overhead is the fraction of bytes used by the protocol for AeroRP control messages. Delay is the time taken by the packet to reach the destination node's MAC protocol from the source node's MAC protocol.

4.2 Simulation Analysis

We compare the AeroRP performance when the geolocation information of the nodes (topology information) or no location information is present in the AeroNP header (link information). Figure 4 shows the performance of different modes of AeroRP in terms of PDR as the node density increases from 5 to 60 nodes, moving at a velocity of 1200 m/s. The GS update mode with LinkInformation performed better until the node density reached 35 nodes and later it started to degrade as the node density further increased. The increase in node density led to more link breakages. This may have resulted in breakages in the end-to-end paths determined by the node. Though the GS tries to update the changes in link information by sending out trigger updates, there is a chance that some nodes might not receive all the updates leading to inconsistent routing of packets. As the node-density increases, the AeroRP update mode without GS updates and using geolocation information begins to perform comparatively better as the nodes in this mode exchanged geolocation information.

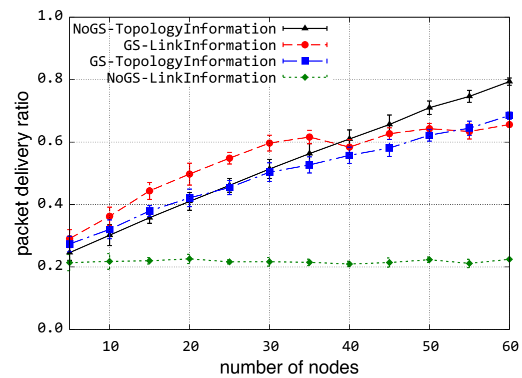

Figure 4: Effect of node density on PDR

Figure 5 shows the variation of PDR as the velocity increases from 50 m/s to 1200 m/s for 35 nodes. At 35 nodes the average network connectivity is 75%. The average network connectivity is determined by using the geolocation information of nodes and their transmission ranges. Average network connectivity of 75% is chosen to ensure we can analyse the PDR performance in a not fully connected network. We see that AeroRP in presence of GS and using LinkInformation performed better. This is because with the 3D Gauss-Markov mobility model ANs move without any abrupt change in their direction. Thus the links predicted by the GS remain stable most of the time. However, as the velocities increase there are many link breakages leading to the GS sending an increased number of updates. As the ANs run the Dijkstra shortest-path algorithm they need to have consistent information from the GS, otherwise this may lead to inconsistent routing of packets. The other modes of AeroRP using TopologyInformation, with and without the presence of GS updates, perform better as the velocities increase. This is because at high velocities the nodes have more chance of identifying a path to the destination as they come across different neighbours which increases their chance to identify a route to the destination. Furthermore, they are aware of the entire topology of the network as they exchange geolocation information in the AeroNP headers. AeroRP mode using LinkInformation in the absence of GS updates does not perform well as it knows only its neighbours.

Figure 6 shows the end-to-end packet delay with varying node densities. We see that the delay is greater while using TopologyInformation with and without the presence of GS updates. This is because the calculation of TTI is based on the node's current and predicted geolocation coordinates. By using TopologyInformation from GS updates, we can predict the node's trajectory information for a longer duration as the expire time provided by the GS is usually greater compared to the neighbourholdTime. This results in having higher TTI values while using TopologyInformation from the GS updates. TTI values specify how long a packet should be buffered before identifying a route to the destination. The other two modes using LinkInformation have very low delay as they do not use the TTI metric.

The variation of AeroRP control overhead with node density in its various modes of operation is analysed in Figure 7.

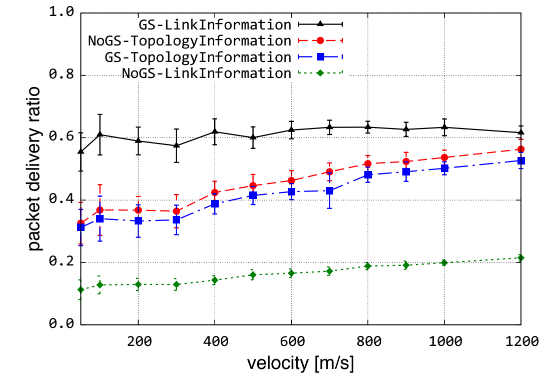

Figure 5: Effect of node velocity on PDR

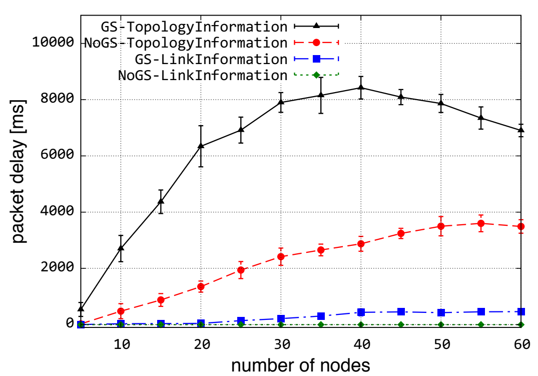

Figure 6: Effect of node density on delay

The control overhead in the absence of GS updates and using LinkInformation is significantly less as there are no AeroRP messages sent out except for the initial bootstrap beacons. Control overhead however, increases exponentially with the increase of node density for AeroRP mode with GS updates enabled and sending out LinkInformation. This is because as the node density increases, so do the number of links between those nodes leading to more updates from the GS whenever a new link is formed or an existing link is broken.

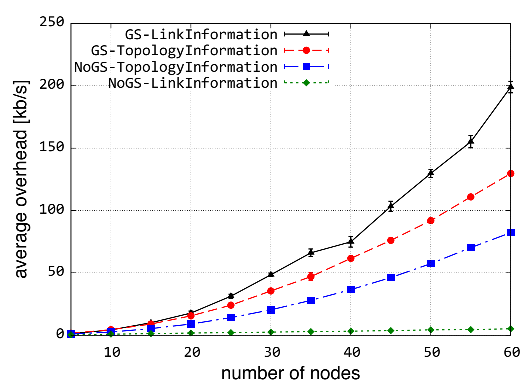

Figure 7: Effect of node density on overhead

5. CONCLUSIONS

In this paper we presented the various AeroRP modes concentrating on the GS update mode using both TopologyInformation and LinkInformation. We presented a detailed analysis of these different modes by modelling them in ns-3 and analysing their performance under various operations scenarios. We showed that location information in the AeroNP header increases packet delivery ratio as the network density increases. Our results also indicate that using GS update mode increases the overhead.

As part of the future work, we plan to analyse AeroRP performance using TDMA MAC model and compare its performance benefits over 802.11 MAC. We are also starting to implement these protocols on prototypes [2].

Acknowledgments

The authors would like to thank the Test Resource Management Center (TRMC) Test and Evaluation/Science and Technology (T&E/S&T) Program for their support. This work was funded in part by the T&E/S&T Program through the Army PEO STRI Contracting Office, contract number W900KK-09-C-0019 for AeroNP and AeroTP: Aeronautical Network and Transport Protocols for iNET (ANTP). The Executing Agent and Program Manager work out of the AFFTC. This work was also funded in part by the International Foundation for Telemetering (IFT). We would like to acknowledge the ResiliNets group members Abdul Jabbar, Justin P. Rohrer, Kevin Peters, and Dan Broyles for useful discussions on this work.

6. REFERENCES

[1] iNET Needs Discernment Report. Central Test and Evaluation Investment Program, May 2004.

[2] M. Alenazi, S. A. Gogi, D. Zhang, E. K. Çetinkaya, J. P. Rohrer, and J. P. G. Sterbenz. ANTP Protocol Suite Software Implementation Architecture in Python. In *Intl. Telemetering Conf. (ITC)*, Las Vegas, NV, Oct. 2011.

[3] D. Broyles, A. Jabbar, and J. P. G. Sterbenz. Design and Analysis of a 3–D Gauss-Markov Mobility Model for Highly-Dynamic Airborne Networks. In *Intl. Telemetering Conf. (ITC)*, San Diego, CA, Oct. 2010.

[4] A. Jabbar and J. P. G. Sterbenz. AeroRP: A Geolocation Assisted Aeronautical Routing Protocol for Highly Dynamic Telemetry Environments. In *Intl. Telemetering Conf. (ITC)*, Las Vegas, NV, Oct. 2009.

[5] H. Narra, E. K. Çetinkaya, and J. P. Sterbenz. Performance Analysis of AeroRP with Ground Station Updates in Highly-Dynamic Airborne Telemetry Networks. In *Intl. Telemetering Conf. (ITC)*, Las Vegas, NV, Oct. 2011.

[6] ns-3 network simulator. http://www.nsnam.org.

[7] K. Peters, A. Jabbar, E. K. Çetinkaya, and J. P. Sterbenz. A Geographical Routing Protocol for Highly-Dynamic Aeronautical Networks. In *IEEE WCNC*, pages 492–497, Cancun, Mar. 2011.

[8] J. P. Rohrer, E. K. Çetinkaya, H. Narra, D. Broyles, K. Peters, and J. P. G. Sterbenz. AeroRP Performance in Highly-Dynamic Airborne Networks using 3D Gauss-Markov Mobility Model. In *IEEE MILCOM*, pages 834–841, Baltimore, MD, Nov. 2011.

[9] J. P. Rohrer, A. Jabbar, E. K. Çetinkaya, E. Perrins, and J. P. Sterbenz. Highly-Dynamic Cross-Layered Aeronautical Network Architecture. *IEEE Trans. Aerosp. Electron. Syst.*, 47(4):2742–2765, Oct. 2011.