International Journal of Wireless & Mobile Networks (IJWMN) Vol. 3, No. 2, April 2011

SAMPCAN: A NOVEL CACHING TECHNIQUE FOR CLIENT-SERVER INTERACTION MODEL IN LARGE AD HOC NETWORKS USING RESAMPLING METHODS

Paramasiven Appavoo

Dept. of Computer Science & Engineering, University of Mauritius, Réduit, Mauritius
p.appavoo@uom.ac.mu

ABSTRACT

The scalability of routing protocols has been a prominent topic to increasing the lifetime of a mobile ad hoc network. One of the underlying features behind scalable protocols is the localised nature of path maintenance. However, nodes' data communications are excluded, despite being the main purpose of the routing protocols. SAMPCAN is a novel caching technique that can be applied to numerous applications of the client-server interaction model for an optimised storage and transmission of data. SAMPCAN uses a subsampling method that can be applied to two data types, mainly (1) XML document, and (2) Image. This novel method promotes the locality of nodes cooperation and interactions to fulfil requests of neighbouring nodes. In this paper, the application of SAMPCAN is discussed for the scalability and increased lifetime of mobile ad hoc networks.

KEYWORDS

Caching, routing, scalability, ad hoc network, client server

1. INTRODUCTION

1.1. Mobile computing with ad hoc network

Mobile computing has revolutionized the world of e-services by allowing people to roam and remain connected allowing them to have access to same service anywhere and anytime. This has mainly led to increased productivity and efficiency. While the driving technologies behind are mainly Mobile IP and wireless communication, miniaturization is allowing powerful devices to be portable. Moreover, according to Moore's Law, the number of active devices that can be placed over a given area of silicon doubles every 18 months. This implies that the evolution and proliferation of mobile computing is far from an end to itself.

Current research is also focusing on the seamless integration of portable devices with their environment in an ad hoc manner. This has been made possible with standard ad hoc mode of wireless communication like mainly Bluetooth (IEEE 802.15.1) and WiFi (IEEE 802.11 b/g), which are quite common with their widespread range of applications in several domains. While Bluetooth is short range and is mainly use for Personal Area Network, with WiFi support, it is possible to have wireless LAN. Heterogeneous devices from different vendors just need to support any of such standards to allow for interoperability. In addition, service discovery

DOI : 10.5121/ijwmn.2011.3214

protocols have been developed alongside with the standards which mainly allow the deployment of services and access to services by service providers and clients respectively.

The multi-hop routing nature of ad hoc network has seemed to be promising to unplanned circumstances like natural disasters, emergency systems, law enforcement… where the infrastructure for communication may have been destroyed or inexistent. Ad hoc network has the potential of a prominent tool for disseminating information efficiently when coordinating operations in a multi-disciplinary emergency system. This has lead to a new generation of application software and protocols to emerge and has been an active area of research since the past five years. Some researchers [1] have brought routing protocols at the level of the user space to simplify the development, deployment and portability of applications.

However most routing protocols designed for ad hoc network focused on data transmission rather than data access. It can be concluded that the usage of the limited resources, available in mobile devices, are not optimized with respect to the data and services that were accessed. As such, a number of cache schemes emerged so as to generally improving the performance and network lifetime by shortening the response-time of queries and limiting multi hop transmissions to a close cache respectively.

1.2. Motivation

Despite the existence of scalable routing protocols for ad hoc network, nodes' requests in a large ad hoc network still experience long delays. This is mainly due to its inherent characteristic of multi hops path with respect to distant sources. Most of the data received are cached locally by the applications to improve performance. However, due to the affinity of nodes' requests in a particular region it is very much likely that a node requests the same object at the same quality level or in some cases at a lower or higher quality. Examples of such objects are local news and weather information with relevant images or map details for navigational purposes. The requested object may be fully or partially cached already by its neighbouring nodes or by nodes which are much closer than the source provider may be. However, existing caching methods do not consider (1) the quality of the required object, (2) how existing cached objects can contribute to fully/partially fulfil similar requests, (3) cache admission and replacement optimizations by keeping a lower quality of the cached object rather than deleting. It is to be noted that localised fulfilment of requests favours the scalability and lifetime of a mobile ad hoc network.

2. RELATED WORK

2.1. Scalability of routing protocols

Most protocols designed for ad hoc networks were not meant for large scale networks. A few routing protocols that have the characteristics of scaling up are MLANMAR [2], CAMP [3], RBM [4], LAM [5], PBM [6], HSR [7], FSR [8],MHMR [9], SENCAST [10] and WSR [11]. The properties of these protocols which contributed to their scalability are (1) reducing the frequency of update messages, (2) reducing the size of the update messages as far as possible, (3) the maintenance of link failures are localized, (4) locality of communication as the network grows, (5) cluster-based routing approach with cluster-head, and (6) GPS assisted routing, which tremendously reduces the number of packets sent for route discovery. It has also been

shown by [12] that delay-tolerant applications can take the advantage of nodes' mobility to scale up.

As the network becomes larger, the following should be noted as well: (1) it is obvious that whenever the data source is far from the requester, the latter will inevitable experience long query delays due to the multi-hops path, (2) The network may also exhibit low data availability whenever one of the node along the path fails to respond, (3) affinity of nodes requests, based on actual locations, may waste bandwidth and eventually reducing the network lifetime, and, (4) just like the notion of localized maintenance is beneficial, so will be the fulfillment of requests as far as possible.

In [13, 14], it has also been shown that sharing of information by protocols across different layers increases network performance. A number of well-known protocols were studied in [15] to show the scalability of routing protocols for heterogeneous and homogeneous MANETs.

2.2. Cache replacement schemes for ad hoc network

Apart from a having a faster response to queries, caching can be used to increase the lifetime of the network as shown in [16] by reducing energy consumption. A number of caching replacement schemes were evaluated in [17, 18, 19] and are summarised as follows:

- LRU policy - Least recently used objects are swapped first.
- LRUMIN policy – since objects may be of different sizes, LRUMIN minimizes the number of objects to be replaced. As such LRUMIN leaves generally smaller objects in the cache for a longer period.
- Size policy – the largest object is replaced with the newly accessed object. Ties, if occurred, are resolved using time since last accessed.
- Key-based policies – objects' attributes like frequency of access, size, time since last access and entry time in cache are the keys determining the replacement. Objects to be replaced are sorted and swapped using different levels of keys.
- Function-based replacement policies - apply a sort of formula that involves a number of objects' attributes, as stated above, to select the object to be disposed.

In addition to a cache replacement scheme, a proper admission policy is also a requirement for effective caching. With respect to the characteristic that ad hoc networks are resource-scarce, it may prove ineffective to cache all objects that are accessed. In [20], it was proposed a cooperative cache-based data access framework to let mobile nodes cache the data, data path or both. Also in [21], the concept of intermediate nodes fulfilling requests by checking local cache was showed without supplementing any admission control or cache replacement scheme.

3. SAMPCAN

3.1. SAMPCAN Communication Model

The SAMPCAN communication model is shown in figure 1. The novel caching technique proposed is SAMPCAN which is based on subsampling data for a more efficient caching in

large ad hoc networks. Downsampled versions of the required object come from several caching nodes to build up the requested data.

Client applications request data via their local proxy which checks the local cache. If the request cannot be fulfilled locally, it is forwarded to neighbouring nodes which in turns forward the request if the latter is unable to contribute to the reply. Intermediate nodes, along the path taken by the request, respond to the client node when they can fully/partially contribute to the reply on behalf of the information source, i.e. the server. The partial replies from the intermediary nodes are then upsampled by the client proxy to fulfil the initial request.

In the worst case, the request reaches the server which replies the sender. Nodes along the returning path may choose to cache the reply fully/partially depending on the outcome of the local cache admission control.

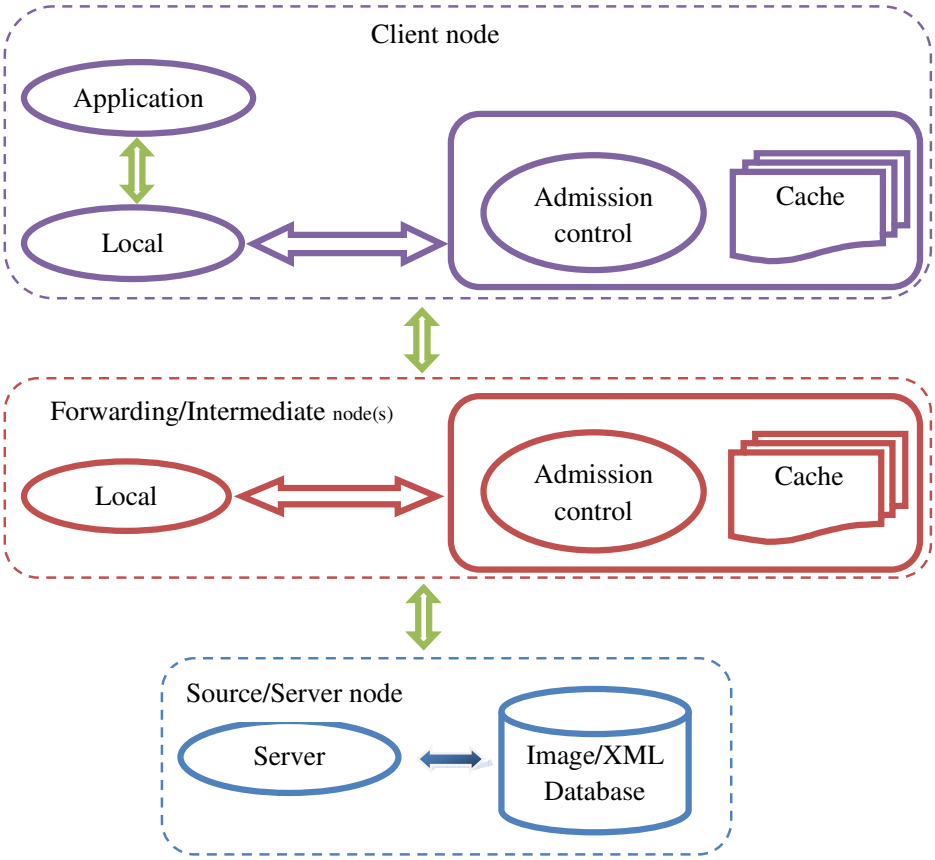


Figure 1. SAMPCAN communication model

3.2. Data types supported

The proposed caching method allows for the efficient transmission of mainly two data types associated with location-based information (1) XML Document, and (2) Images.

3.2.1. XML Document

Database records in XML format as XML was designed to transport and store data. For example, information from an XML weather service may look like table 1.

Table 1. XML weather data example.

```
<?xml version=”1.0?>
<weather_observations>
        <location>New York</location>
        <latitude>40.56</latitude>
        <longitude>-74.54</longitude>
        <weather>Sunny</weather>
        <temp_celcius>28.4</temp_celcius>
</weather_observations>
```

The above can be consumed by any application for a suitable display.

3.2.2. Images in jpeg 2000 format, for places of interests & map

The JPEG 2000 image format is preferred mainly because (1) images encoded JPEG 2000 with multiple quality layers are more error-resilient than JPEG as shown in [22], (2) downloading/streaming of jpeg 2000 images region-of-interest (ROI) is possible using HTTP, as depicted in [23]. Other interesting features are highlighted in [24].

As noted above, it is possible to build an image of higher quality when images of lower quality are available. Moreover, if different parts/tiles of an image are accessed by different nodes, a larger image or a particular ROI can be build from what was retrieved earlier.

As most applications cache whatever is retrieved, the IP layer of intermediary nodes queries the application layer before forwarding any http request. If the latter holds a valid copy of the requested information, a reply on behalf of the source is sent to the requestor.

3.3. Caching algorithm

The caching algorithm is depicted using a scenario where a node, x, requests location-based information from y:

Node x checks its local cache. Given that the required object is not present, the request is forwarded to other intermediate nodes to data source y using any georouting protocol. If the IP of y is unknown, the message is then forwarded to the location of the requested information.

Intermediary nodes check their local cache through their respective proxy. If any one of them holds a valid copy of the data requested, it sends an acknowledgement, on behalf of the source. As the reply moves from the proxy towards the requestor, depending on the routing protocol used, intermediary nodes’ addresses are piggybacked with the acknowledgement message. Alternatively, georouting may be used, instead of stacking a set of IP addresses to maintain a path, to increase efficiency.

The client node chooses the closest proxy for the data. Depending on the outcome of the admission control and replacement policies, discussed in the next section, intermediary nodes may choose to save a copy of the forwarded object. In order to avoid localized cache redundancies, i.e. close neighbours caching the same pieces of data, the cache control of intermediary nodes ensures that the proxy lies in a different track and sector as depicted in the figure 2. As the number of tracks decreases: (1) cache redundancies across neighbouring nodes increases, (2) resiliency increases, and (3) frequency of updating cache records increases as the probability that the proxy lies in a different track increases.

The distance between any node and the information source/server, calculated using the standard two points on a 2D plane formula using the conversion from geodetic to 2D coordinate system using earth model in [25], determines the track. The GPS coordinates of the receiver or caching node with that of the source determines the sector as shown next.

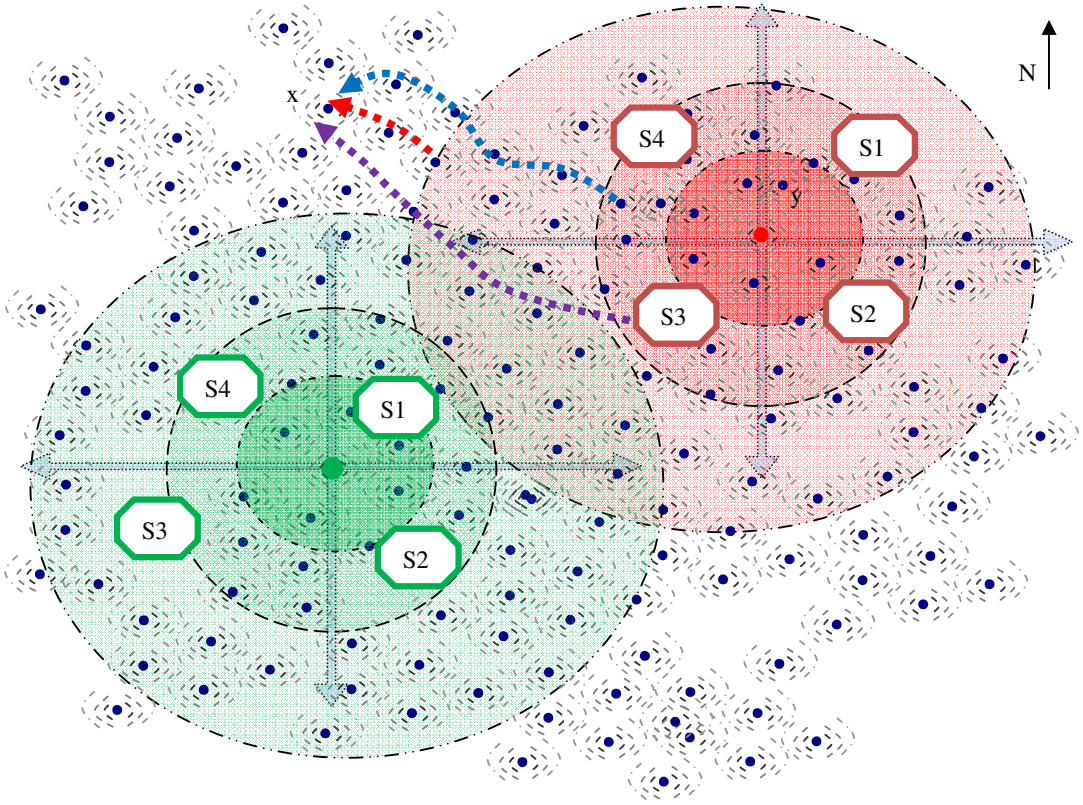


Figure 2. Track and sectors

Given that the longitude and latitude of the source are α and β respectively and four sectors are chosen, a receiver/caching node φ is in:

Sector S1, if longitude(node φ) is > α and latitude(node φ) is > β
Sector S2, if longitude(node φ) is > α and latitude(node φ) is < β
Sector S3, if longitude(node φ) is < α and latitude(node φ) is < β
Sector S4, if longitude(node φ) is < α and latitude(node φ) is > β

3.4. Pseudo codes

Node x requesting for an object from location y

Step 1 - Node x:

Request $object_x$ from location y

Local proxy checks the local cache

If $object_x$ is found

    If valid

        Use $object_x$

    Else delete $object_x$ & forward request to neighbouring nodes

Else forward request to neighbouring nodes, goto step 2

Step 2 - Request forwarded by intermediary node n_i:

Local proxy of n_i checks the local cache

If $object_x$ is found

    If valid

        Piggybacked $object_x$ with reply to node x

    Else delete $object_x$ & forward request to neighboring nodes

Else

Forward request to neighboring nodes

Update requestMap, set value to value+1 where key = $object_x$

Step 3 - Data source from location y sends reply:

$object_x$ is sent to node x

Step 4 - Reply forwarded by intermediary node n_i:

Forward reply towards node x

Local proxy checks actual track and sector wrt to location y

If sender of $object_x$, or subsample of $object_x$, is located in a different track and sector

Run cache replacement function, step 5

(Nb sender can be source of information or a proxy.)

Step 5 - Cache replacement function – image / XML data:

Set TTL = 0

While free space < Space required for $object_x$ or its subsample

  Increment TTL

For i up to number of objects in cache

If time-to-live of $Object_i$ is equal to 0, remove $Object_i$

Else If value in requestMap* related to key $object_x$ > No. of requests for $Object_i$ in cache

OR time-to-live of $Object_i$ is <= TTL

Replace $Object_i$ with subsample($Object_i$)

Save $object_x$ or its subsample

*The request map stores the number of requests received for a particular object (an image or a database/records).

3.5. Object sub-sampling methods for transmission/storage

3.5.1. Image sub-sampling

Objects are basically downsampled to use up less storage space when a lower quality of the former is still fulfilling its purpose. Given that four nodes at different locations retrieved a subsampled $object_x$, these subsampled versions, when cached, are useless when a request for a better quality of the object emerges from any nodes. However, the cached objects of lower quality can be used to reconstruct objects of better quality when the sender uses different offset values and sample periods to generate subsampled objects. The following illustrates:

Given that an image, img, is as shown in figure 3, whose the starting corner of the image is identified by x and y and (2) the number of samples that exist is υ, where there υ is equal to the sample period along the x axis multiply by the sample period along the y axis, then the subsample a is defined by:

x = offset of img for sample a along the x-axis

y = offset of img for sample a along the y-axis

Set i and j to zero

for i is less than img.length

for j is than img.width

sample a = sample a + img.pixel(x+i, y+j)

increment j by the sample period

increment i by by the sample period

Reset j to zero

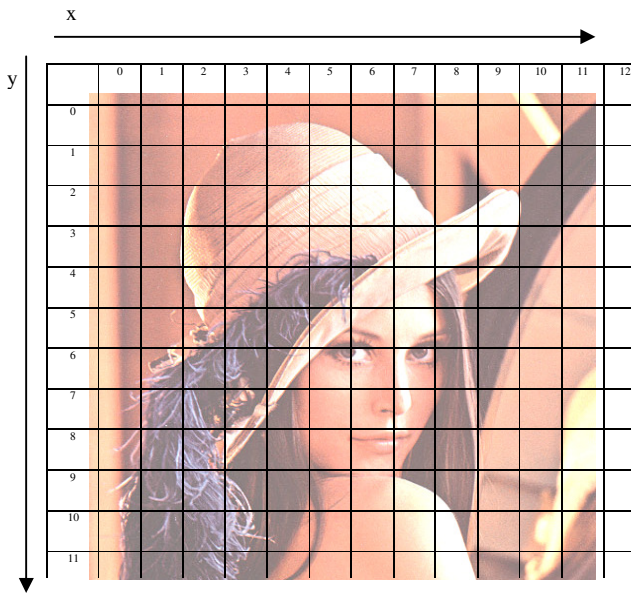


Figure 3. Subsampling an image given an offset and a period

Figure 4 and figure 5 show the image cells/pixels chosen for two samples (a & b) respectively when the above algorithm is applied to img with sample period 2 for both the x-axis and the y-axis (here υ=4), and the subsample offset (x, y) for sample a, b, c and d are (0, 0), (1, 0), (0, 1) and (1, 1):

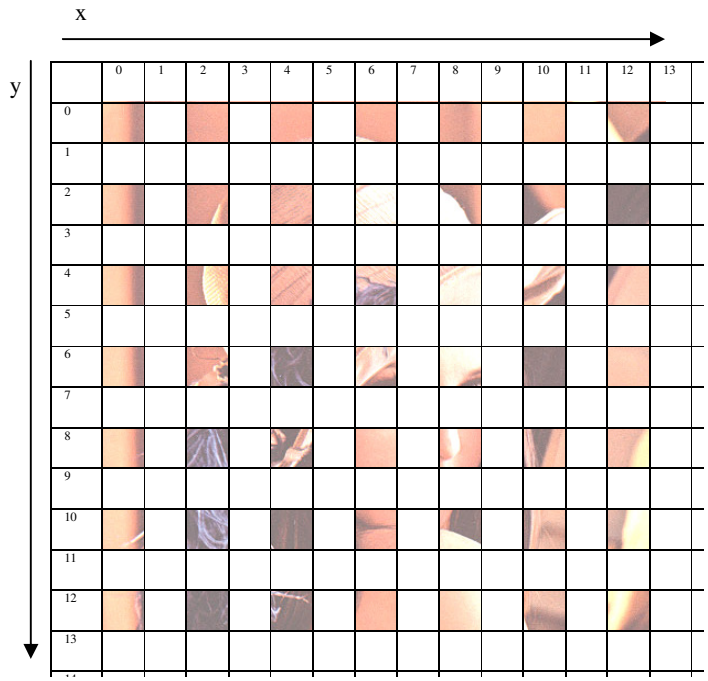


Figure 4. Image cells/pixels used for sample a

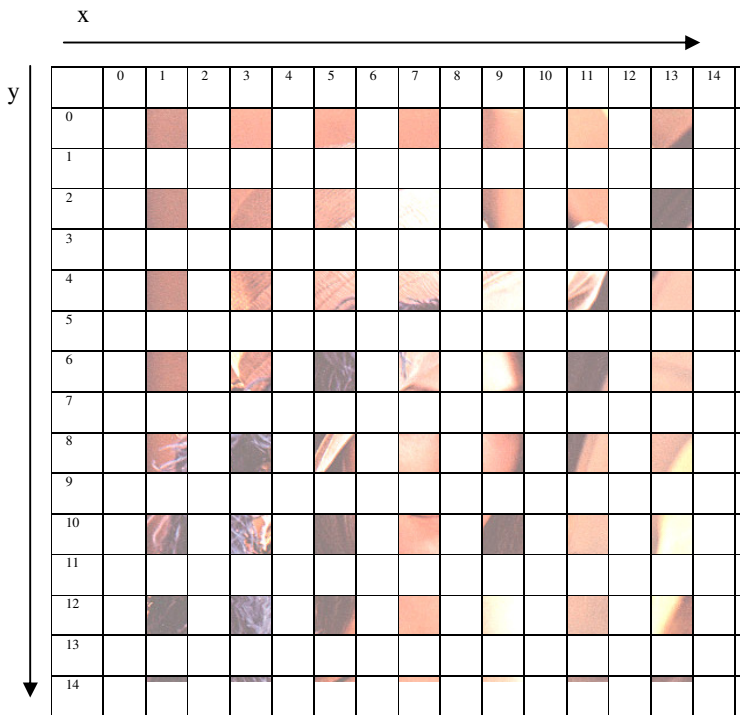


Figure 5. Image cells/pixels used for sample b

Note that the Subsampling period and offset indicate the number of pixel to advance in a direction after choosing a pixel for the sample and the first pixel of the subsample image respectively.

3.5.2. XML data sub-sampling

Records accessed by nodes are stored in XML forms. The root element and the first level branch have the name of the database and table respectively. For example, table 2 shows data that was accessed from the table **observation** found in the database **weather_observations** by two different nodes, x and y.

Table 2. Example of xml data for weather observations.

<?xml version="1.0?> <weather_observations> <observation> <rec_id>1</rec_id> <location>New York</location> <latitude>40.56</latitude> <longitude>-74.54</longitude> <weather>Sunny</weather> <temp_celcius>28.4</temp_celcius> <obserTime>16 :00</obserTime> </observation> … </weather_observations>	<?xml version="1.0?> <weather_observations> <observation> <rec_id>2</rec_id> <location>New State</location> <latitude>45.46</latitude> <longitude>-34.54</longitude> <weather>Cloudy</weather> <temp_celcius>18.4</temp_celcius> <obserTime>16 :10</obserTime> </observation> … </weather_observations>
Node x	Node y

[26] brought forward XQuery to manipulate xml data like database data. As such details of weather observations are easily retrieved from intermediate nodes and merged to fulfill other queries.

Subsampling of an xml document can be done at the level of a record tag, e.g. 'observation', or an attribute/field of a record, e.g. 'location'. In the latter case, the attributes of a record are listed in order of priority/importance, then the downsampling process deletes the last attribute of the record as listed. When node x downsamples the 'weather_observations' document, in a first instance the attribute 'obserTime' of all records gets deleted.

Whenever location information is available, like in the above file, higher priority is given to caching records whose locations is nearest to the node's actual position

4. DISCUSSIONS/EXPERIMENTATION

Java Advanced Imaging [27] was used to manipulate the JPEG 2000 image. Images were downsampled and used to reconstruct their originals as shown in the series of figures below. Using υ=4, downsampled versions (a, b, c & d) of the original lena image were obtained as shown in figure 6. The original image is of size 349 Kb and the downsampled image size given υ=4 was 109 Kb. However, there is no direct relationship between the downsampled image size and υ, as images' attributes largely affect its compression. Figure 7 and 8 shows reconstructed images when subsamples a & b and subsamples a,b & c are available respectively. While figure 9 shows the original image reconstructed given the availability of all four subsampled versions.

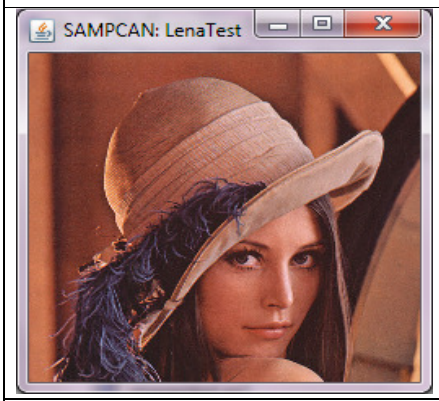
Whenever images were reconstructed, missing pixel information was considered as white. An improvement is sought with a block-based predictive algorithm to predict the missing pixel information using the values of immediate neighbours.

Despite the overall downsampled data sent is relatively larger than the original, there is still a reduction the risk of network partitioning and an increase the lifetime of the network as the data comes from (1) various parts of the network, and (2) proxies that are closer than the actual source would be.

It should also be noted that this novel caching technique can be applied to any other client-server model with multi-hop paths.

Lenasubsample a – Size: 109 Kb, resolution: 256 * 256

Lenasubsample c – Size: 109 Kb, resolution: 256 * 256



Lenasubsample b – Size: 109 Kb, resolution: 256 * 256

Lenasubsample d – Size: 109 Kb, resolution: 256 * 256

Figure 6. Subsampling of Lena image with υ=4

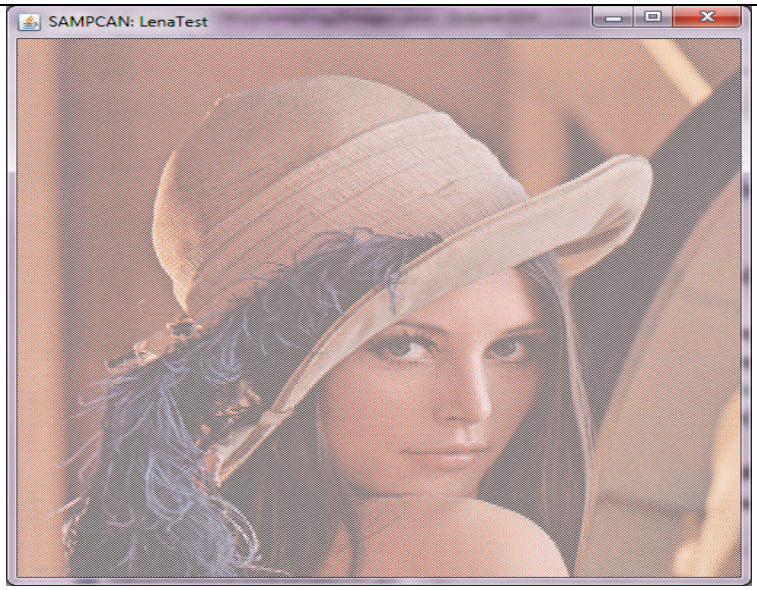


Figure 7. Reconstructed image – given sample a & b, resolution: 512 * 512

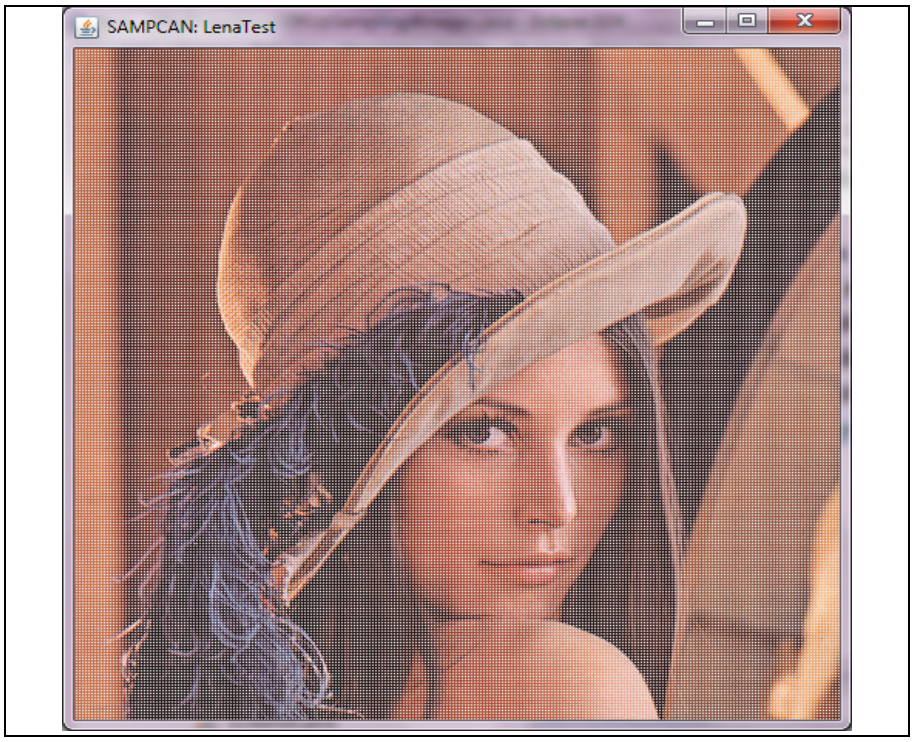


Figure 8. Reconstructed image – given sample a, b & c, resolution: 512 * 512

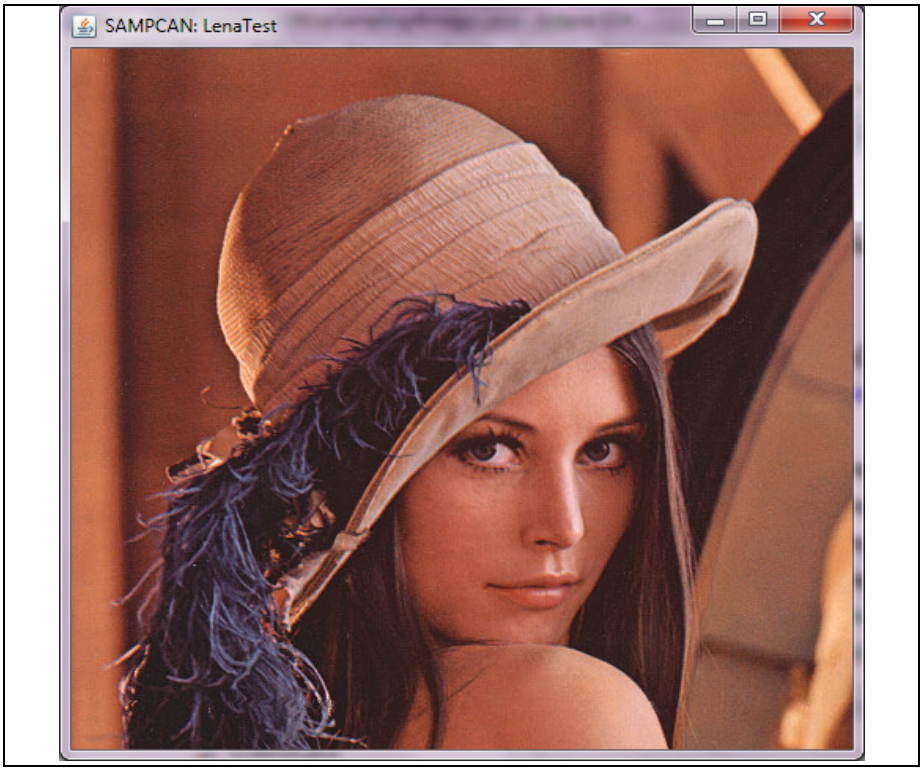


Figure 9. Reconstructed image – given all four downsampled versions, resolution: 512 * 512

5. FUTURE WORKS

A context-aware cache admission control is required for to dynamically update the threshold of the following parameters which can be used to optimise the replacement of a cached object c_x at node n_i: (1) distance between n_i and the data source x, (2) object time-to-live as timestamped by the data source, (3) Number of similar requests, (4) time of last request, and (5) whether subsample of the object exists already. The optimum number of tracks and sectors with respect to context of use needs to be evaluated before applying SAMPCAN.

6. CONCLUSION

One of the mentioned features for the scalability of routing protocols is the locality of communication as the ad hoc network grows. This mode of interaction was restricted to nodes in the process of maintaining communication paths or gathering local context information. In the same line, SAMPCAN promotes localised communication in the interactions of client and server dimension. SAMPCAN introduces a novel caching technique that optimised the caching storage using subsampling that can support mainly two data types, images and XML document. Cached elements are accessible via a local proxy by neighbouring nodes requesting similar data. The quality of objects required can be augmented using subsampled cached versions from several nodes. Apart from extending the availability of the local cache to neighbouring nodes, the caching method optimises storage with at least maintained cache hit ratio as existing objects are not swapped with new elements but downsampled.

7. REFERENCES

[1] Pedro García López, Raúl Gracia Tinedo and Josep M. Banús Alsina**,** Moving routing protocols to the user space in MANET middleware, Journal of Network and Computer Applications, vol. 33, issue 5, 2010.

[2] Yi Y., Gerla M. and Obraczka K., "Scalable team multicast in wireless ad hoc networks exploiting coordinated motion", Ad Hoc Networks Journal, August 2003.

[3] Madruga E. L. and Garcia-Luna-Aceves J. J, "Scalable multicasting: The core-assisted mesh protocol", ACM/Baltzer Mobile Networks and Applications, Special Issue on Management of Mobility, 6(2):151--165, 2001.

[4] Corson M. S. And Batsell S. G., "A reservation-based multicast (RBM) routing protocol for mobile networks: Initial route construction phase", ACM/Baltzer Wireless Networks, vol. 1, no. 4, pp. 427-450, Dec 1995.

[5] Ji L. and Corson M. S., "A lightweight adaptive multicast algorithm", Proc. of Globecom'98, 1998.

[6] Mauve M., Füβler H., Widmer J. and Lang T., "MobiHoc Poster: Position-based multicast routing for mobile ad hoc networks", Mobile Computing and Communication Review, Vol. 7, No. 3, pp. 53 – 55, 2003.

[7] Iwata A., Chiang C., Pei G., Gerla M. and Chen T., "Scalable routing strategies for ad hoc wireless networks", IEEE Journal on Selected Areas in Communications SAC, 1999, Vol. 17, pp. 1369-1379.

[8] Pei G., Gerla M. and Chen T., "Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks", IEEE International Conference on Communications, 2000, Vol 1, pp. 70-74.

[9] An B. and Papavassiliou S., "A mobility-based hybrid multicast routing in mobile ad hoc wireless networks", Proc. of MILCOM 2001, Vienna, VA, Oct 2001.

[10] Appavoo P. and Khedo K., "SENCAST: A Scalable Protocol for Unicasting and Multicasting in a Large Ad hoc Emergency Network", International Journal of Computer Science and Network Security, Vol.8 No.2, February 2008.

[11] Acer U.G., Kalyanaraman S. and Abouzeid A.A., Weak State_Routing_for Large-Scale Dynamic Networks, IEEE/ACM Transactions on Networking, vol. 18, issue 5, 2010.

[12] Grossglauser M. and Tse D. N. C., "Mobility Increases the Capacity of Ad hoc Wireless Networks", IEEE/ACM Transaction on Networking, vol. 10. no. 4,Aug. 2002.

[13] Chen K., Shah S. H. and Nahrstedt K., "Cross-Layer Design for Data Accessibilityin Mobile Ad Hoc Networks", Wireless Personal Communications 21: 49–76, 2002.

[14] Conti M., Maselli G., Turi G. and Giordano S., "Cross-Layering in Mobile Ad Hoc Network Design", Computer IEEE Computer Society, pp 48, 2004.

[15] Al Amri H., Abolhasan M. And Wysock T., Scalability of MANET routing protocols for heterogeneous and homogenous networks, Signal Processing and Communication System, Elsevier, vol. 36, issue 4, 2010.

[16] Nitnaware D. and Verma A, Energy constraint Node cache based routing protocol for Adhoc Network, International Journal of Wireless & Mobile Networks (IJWMN), Vol.2, No.1, 2010.

[17] Abrams M., Standridge C., Abdulla G., Williams S. and Fox E., "Caching Proxies: Limitations and Potentials," Proc. Fourth Int'l World Wide Web Conf., Boston, 1995.

[18] Aggarwal C., Wolf L. and Yu P., "Caching on the World Wide Web", IEEE Transactions on Knowledge and Data Engineering, vol. 11, no. 1, 1999

[19] Williams S., Abrams M., Standridge C. R., Abdulla G., and Fox E. A., "Removal Policies in Network Caches for World Wide Web Documents," Proc. ACM SIGCOMM, pp. 293-304, 1996.

[20] Cao G., Yin L. and Das C. R., "Cooperative Cache-Based Data Access in Ad Hoc Networks", Computer – IEEE Computer Society, 2004, VOL 37; PART 2, pages 32-39.

[21] Gonzalez-Canete F. J., Casilari E, and Trivino-Cabrera A., Proposal and evaluation of a caching scheme for ad hoc networks, ADHOC-NOW '09 Proceedings of the 8th International Conference on Ad-Hoc, Mobile and Wireless. 2009

[22] Pekhteryev G., Sahinoglu Z., Orlik P. and Bhatti G., "Image Transmission over IEEE 802.15.4 and ZigBee Networks", Mitsubishi Electric Research Laboratories, 2005.

[23] Deshpande S., Zeng W., "Scalable Streaming of JPEG2000 Images using Hypertext Transfer Protocol", 2001.

[24] Athanassios S., Charilaos C., and Touradj E., The JPEG 2000 Still Image Compression Standard, IEEE Signal Processing Magazine, 2001.

[25] Carlson C. G. and D. E. Clay D. E., "The Earth Model – Calculating Field Size and Distances between Points using GPS Coordinates", Site-Specific Management Guidelines series-11, Potash & Phosphate Institute (PPI), 1999.

[26] W3C XML Query (XQuery), http://www.w3.org/XML/Query/, accessed 1st February, 2011.

[27] Oracle Corporation, Java Advanced Imaging (JAI) API, http://java.sun.com/javase/technologies/desktop/media/jai/, accessed 13th November, 2010.