ℙ periodica polytechnica

Electrical Engineering
55/1-2 (2011) 45–52
doi: 10.3311/pp.ee.2011-1-2.05
web: http://www.pp.bme.hu/ee

RESEARCH ARTICLE

An anycast based feedback aggregation scheme for efficient network transparency in cross-layer design

Zoltán Kanizsai / *László* Bokor / *Gábor* Jeney
Received 2010-08-10

Abstract

To ensure Quality of Service for multimedia data sessions in next generation mobile telecommunication systems, jointly-optimized cross-layer architectures were introduced recently. Such shemes usually require an adaptive media source which is able to modify the main parameters of ongoing connections by transferring control and feedback information via the network and through different protocol layers from application layer to physical layer and vice versa, according to the actual state of the path between peer nodes. This concept of transmitting cross-layer information is referred as network transparency *in the literature, meaning that the underlying infrastructure is almost invisible to all the entities involved in joint optimization due to the continuous conveyance of cross-layer feedbacks. In this paper we introduce and evaluate a possible solution for reducing the network overhead caused by this volume of information exchange. Our soulution is based on the anycasting communication paradigm and creates a hierarchical data aggregation scheme allowing to adapt each entity of the multimedia transmission chain based on frequent feedbacks and even so in a low-bandwitdh manner.*

Keywords

network transparency · aggregation scheme

Acknowledgement

This work is supported by the OPTIMIX project [17] *which is partly funded by the 7th Framework Programme (FP7) of the European Union's Information and Communication Technologies (ICT). The authors would like to thank all participants and contributors who take part in the studies.*

Zoltán Kanizsai
László Bokor
Gábor Jeney
Department of Telecommunications, BME, H-1117 Budapest, Magyar Tudósok körútja 2., Hungary

1 Introduction

According to the latest trends in telecommunication and mobile devices, multimedia contents have become more popular than ever. To satisfy the needs of users and subscribers of multimedia applications, service providers try to do their best to keep the *Quality of Service* (QoS) at an acceptable level. In a tipical multimedia scenario the significant traffic belongs to the downlink – from a remote media source (website, multimedia server, application server, etc.) to the user's terminal. The terminal sends information uplink only when it announces its pretension to start or to stop a service (usually in a User Datagram Protocol (UDP) [27] based communication), or according to the type of the content when it acknowledges the received data packets (Transport Control Protocol (TCP) [28] like communication). In this basic type of scenario the media server provides a constant value of quality and the QoS on the terminal's side is affected by the network segment between the source and the receiver.

To ensure that the terminal-side QoS is not exposed by the intermediate network segment and to keep it approximately on a constant level, adaptive media sources have appeared which are able to change the outgoing quality parameters according to the continuous feedback information from the terminals. Usually, user's terminals collect information about the actual quality of the received service, and in a predefined amount of time, they periodically send this information back to the media source in feedback messages. However, the collected information could be very different according to the actual model of network. The classic ISO-OSI network model [19] is based on a communication stack which includes seven layers creating a modular framework, where layers are allowed to exchange information only with the direct upper and lower layers. The simplified and most widespreaded version of the ISO-OSI model is the TCP/IP network model [11] which represents four layers, but their communication are also limited to their neighbours. In these cases information only can be collected from the scope of the application layer.

Nowadays a new network model becomes more accepted which says the communication of the different layers in the stack should not be limited to its neighbours, because more efficient communication management is achievable if applications could directly exchange information with lower layers. This new model is called *jointly-optimized cross-layer architecture* [21]. Thanks to the free interoperability between layers a cross-layer optimized multimedia application is able to hive system and network parameters from the lower layers and send them back to the adaptive media source. This technique makes possible to continuously observe network conditions between the server and terminal, therefore the source can reduce or increase the bandwith needs of a specific service according to the circumstances in almost real-time. The design of transmitting cross-layer information is called *network transparency*, because it almost hides the whole underlying infrastructure from the network nodes.

In the recent years the cross-layer architecture design became a rapidly developed area of network and protocol engineering, it is simply because this kind of underlying technology can bring significant performance improvement in many transmission scenarios. Some examples from the literature:

In the European Community's IST-PHOENIX project [20] the cross-layer design is used to develop a scalable video coding (SVC) and transmission environment in wireless next generation networks [16]. The child project of PHOENIX is called ICT-OPTIMIX [17] and it also studies video transmission over cross-layered network architecture, however here multicast scenarios are investigated instead of unicast. The OPTIMIX network design is described in [26], which uses Media Independent Handover (MIH, IEEE 802.21) as a basis for a triggering framework that supports adaptive multimedia transmission in multicast scenarios. The feedback aggregation scheme presented in this paper is also based on the OPTIMIX network architecure. The authors of [1] introduce the same MIH framework but they used it for collecting layer-aware information to support mobility management in 4$^{\text{th}}$ generation (4G) environment. One more paper worth mentioning [4], because it summarize well the challenges of multimedia transmission over wireless cross-layer architecture and gives a list of parameters and constraints that should be taken into consideration when an adaptive media transmission technique is developed.

The main disadvantage of the joint optimized design is that feedback messages also require bandwith on the network, thereby further reducing the available resources. This paper describes the feedback traffic generated in a multicast, cross-layer communication enabled network, where the number of clients are large (e.g. cross-layer optimized wired or mobile IP-TV). We introduce our IPv6 anycast [5, 25] based solution, to reduce the amount of feedback messages. We show how this method affects the number of maximum servable peers.

The paper is organised as follows. The next section provides background information about the used technology in our proposal, then in Section 3 we introduce the Anycast based feedback aggregation scheme. In Section 4 a mathematical evaulation of our method is presented, finally Section 5 concludes the paper and shows some possible future work.

2 Background

This section is about to give a little overview and background information about data aggregation in generally and to introduce the IPv6 anycasting paradigm [25] in order to help the understanding of our proposal.

2.1 Data aggregation in nutshell

In general, data aggregation is responsible to compose, recompose, transform, and analyze data originated from wide variety of sources such as real-time sensor nodes, simulators, mobile terminals, etc. There are three main issues to be dealt with while using such aggregation schemes.

First we have to answer questions regarding the ways of data access, for example how the aggregated data can be routed to a particular node so that the data can be processed and merged. Secondly data sources may produce output with different syntax and semantics, so it should be decided what data is being actually collected. To measure the effectiveness of a data aggregation scheme for a complete system with one single numeric value, the size of the original source data can be divided by the size of the aggregated data and this rate number is called aggregation ratio. The third main issue brings forward the effect of timing in aggregation schemes where data is sent periodically. The question is how long should an aggregation fork node wait and collect data from its sources before sending aggregated data to the sink. If it waits too short the aggregation ratio could be low resulting a not so effective scheme, but if it waits too long the collected data might interfere with the information collected in the previous periods. In other words this is a trade-off constraint between energy and bandwith consumption of the sources and data accuracy at the sink node. Of course, the problems caused by this timing issue depends on the nature of the aggregation data, for example how much the source data is sensitive to delays introduced by the intermediate network nodes and the aggregation system itself.

A data aggregation scheme is based on a set of adaptive methods which can merge and aggregate information from wide scale of possible data sources and data types into well organized and uniformized datagrams. Data aggregation can be grouped by two main aspects. *Routing-centric* aggregation mechanisms mainly cover routing problems, for example when and physically where two or more information pieces can meet each other in order to be aggregated. *Data-centric* aggregation schemes mainly include coding, calculation, and compression of aggregatable data coming from multiple sources, using mathematical functions (e.g. MAX, MIN, AVERAGE, etc.) as aggregation functions. Further in this paper we only deal with routing-centric aggregation.

The authors of [9] consider the packet forwarding mechanisms of data aggregation schemes and discuss three different type of aggregation. The first one is *structured aggregation* where a fixed forwarding structure called forwarding tree is set up in advance, and then, packets can be aggregated at the tree forks. Fixed-structure data aggregation methods can meet the requirements of simple and fast queries, but they receive too much communication overhead while constructing and maintaining these tree structures, resulting in that such schemes are not quite applicable to all-wireless dynamic environments like ad-hoc and sensor networks. *Structureless aggregation*, the second type, does not maintain fixed forwarding structure during the aggregation procedures. The information is collected, aggregated, and rebroadcasted by each node in a periodical and stochastic manner. The main drawback of such a scheme is that periodical broadcasts dramatically increase communication overhead, potentially deteriorating the operation of other applications. The last type, *semistructured aggregation* is intended to combine the benefits of both of the previous aggregation methods, but often introduce more complicated aggregation structures, which are not suitable for dynamic, all wireless environments.

According to the applications on the data sources two types can be considered: *event-based* and *data-gathering* applications [10]. Event-based applications are continuously observing some predefined parameters and they only send data on the network when some kind of predetermined circustances trigger them. The data-gathering approach differs from the previous one that it constantly measures one or more parameters and the values of them are sent to the sink periodically. One of the most important scope of data aggregation with data-gathering applications is real-time monitoring [22, 23, 30], where the main issue is to maintain a relatively accurate current "view" of the network by a control node (sink). This type of monitoring is the basis of the latest multimedia networks, since the high, periodical sampling rates of parameters in low delay feedback messages provide real-time information about the network's and the terminals' conditions, which is highly optimal for proper multimedia data transmission. According to the drawbacks of large amount feedback messages the main problem is how to achieve effective aggregation of feedbacks with minimal forwarding delay.

As stated in [29] the communication cost is several orders of magnitude higher than the computation cost, the best solution, in order to reduce the energy consumption, bandwith and overhead, is to minify the data volume locally for the long distance delivery. This method is referred as *early aggregation* in the literature. The procedures of such a system can be performed in-network so that communication overhead can be reduced soon after the (often redundant) information is produced [18]. So the benefits of data aggregation can be maximized if aggregation is performed on location-related nodes with semantic-related data.

In this paper we show how to reach a good feedback aggregation ratio, in a jointly-optimized, multicasting, and cross-layer communication enabled network, with *routing-centric* data aggregation and *data-gathering* applications by using IPv6 anycasting.

2.2 Overview of anycasting

Today's communication possesses at least four different kind of delivery modes. The most widespread is the unicast (one-to-one) method, however it is not the only scheme in use: other delivery possibilities, such as broadcast (one-to-all), multicast (one-to-many) and anycast (one-to-one-of-many) are also available. Here we focus on anycasting, which is a group communication scheme originally introduced in RFC 1546 [25]. The basic idea behind the anycast communication paradigm is to separate the service identifier from the physical host, enabling the service to act as a logical entity of the network. This idea of anycasting can be achieved in different layers (e.g. network and application layers) and they have both strengths and weaknesses as well. We focus on network-layer anycasting in this article, where a node sends a packet to an anycast address and the network will deliver the packet to at least one, and preferably only one of the competent hosts.

RFC 1546 introduced an experimental anycast address for IPv4 but in this case the anycast addresses were distinguishable from unicast addresses therefore resulting in difficulties of deployment. In the next generation IP version (IPv6) [5], the anycasting paradigm was adopted as a basic and implicit service. When an IPv6 node sends a packet to an anycast address, the network (based on underlying routing algorithms) will deliver the packet to one host of the anycast group thus establishing one-to-one-of-many communication. In this matter IPv6 anycasting is considered as a group communication scheme, where the group of nodes is represented by an anycast address and anycast routing algorithms are dedicated always to find the most appropriate destination for an anycast packet. The "appropriateness" is measured by the metric of the routing protocol. In IPv6 the anycast addresses can not be distinguished from the unicast addresses, they share the same address space. Therefore the beginning part of IPv6 anycast addresses is the network prefix: the longest P prefix identifies the topological region in which the anycast group membership must be handled as a separate host entry of the routing system. Outside this region anycast addresses of that membership can be aggregated. Recent drafts categorize IPv6 anycast based on the length of P [12]. On one hand Global Anycasting should be taken into consideration, where the value of the P prefix is zero, making aggregation impossible and leading to serious scalability problems: individually stored anycast entries easily could cause explosion of routing tables if anycasting gets widely used. On the other hand, Subnet Anycasting should be considered when anycast packets can reach the last hop router by normal unicast routing, and the current Anycast Responder is determined by the last hop router (e.g. based on Neighbor Discovery). Regional Scoped Anycasting [3] is a natural outgrowth of Subnet Anycasting: the anycast subnet may contain not only one router (i.e. the last hop router) but more, creating a controlled anycast subnet (or region) by restricting the advertisement of anycast routing information.

Anycast routing protocols working in the subnet (i.e. scope-controlled region) should take care of managing the anycast membership and exchanging the anycast routing information. Although the most important element of network anycasting is the underlying routing protocol, current IPv6 standards do not define anycast routing. Beyond the lack of standards, there is quite small amount of literature about practical IPv6 anycasting. However the existing drafts are quite prosperous [7,24,31], there are still challenges to be solved. The problems and possible solutions regarded the current state of researches, and an anycast routing architecture (based on seed nodes, gradual deployment and the similarities to multicasting) are summarized in [6]. The area of secure and reliable anycast group membership management protocol is also being investigated (e.g. [32]), as well as the application problems coming from the stateless nature of anycasting (i.e., an anycast destination is determined on a packet-by-packet basis by the routers) and some possible solutions to it [6]. Due to promising achievements in this area, the restrictions introduced in the first IPv6 standard [13] are now removed [14] in order to ease research, development and deployment of IPv6 anycasting.

Several promising practical application can be imagined based on the above. The most popularly known application of the anycast technology is helping the communicating nodes in selection of service providing servers. In this approach the client host can choose one of many functionally identical servers. As a result, load distribution and balancing can be achieved between the multiple servers when we use a feasible anycast routing protocol, where anycast requests are fairly forwarded. An excellent survey of the IPv6 anycast characteristics and applications can be found in [2, 6, ?appipv6any], where authors describe many advantages and possible applications of anycasting and also address deployment and operational issues of distributed services using anycast for both IPv4 and IPv6 networks.

Just two papers where anycasting is used for efficient data aggregation and for proper sink selection: The first article [15] presents a method how to find the closest sink in a wireless sensor (node) field where multiple mobile sinks exist. The proposed solution is to create Reverse Forwarding Trees (RFT) for every single source node, and when a sink appears it must send a tree connection (join) message to the neighbour sources. The join message is then forwarded to the other sources and by this way every source node can build up an RFT (which is a Shortest Path Tree, SPT) where the sinks represent the leaves. In other words this tree is the graf representation of the anycast routing table. The second paper is [9] which introduces Data-aware Anycasting (DAA) at the MAC layer and Randomized Waiting for event-based applications. As maintaining a forwarding structure requires notable overhead, thus bandwidth and energy, the aim was to create an efficient data forwarding method in a structre-free environment. Data-aware Anycast here represents a small knowledge base which describes who has aggregated data among the neighbours and who is closer to the sink than the actual source. With this distributed information a highly efficient data aggregation is possibly while data constantly moving towards the sink.

In this article we apply network-layer anycasting for an efficient feedback aggregation scheme, where individual feedback messages of a stateless communication model are sent to an anycast address and the network will deliver these packet to at least one, and preferably the most appropriate one of feedback aggregation servers for further processing.

3 Anycast based feedback aggregation scheme

After receiving feedback data from individual mobile terminals, the designated network entities – special nodes called Feedback Aggregation Servers (FAS) – will further aggregate the information and relay this newly composed aggregated data towards the adaptive media source. Feedback Aggregation Servers are supposed to aggregate individual reports originated by mobile terminals, and also to produce final reports containing terminal identificators, time-stamps, and of course actual feedback values.

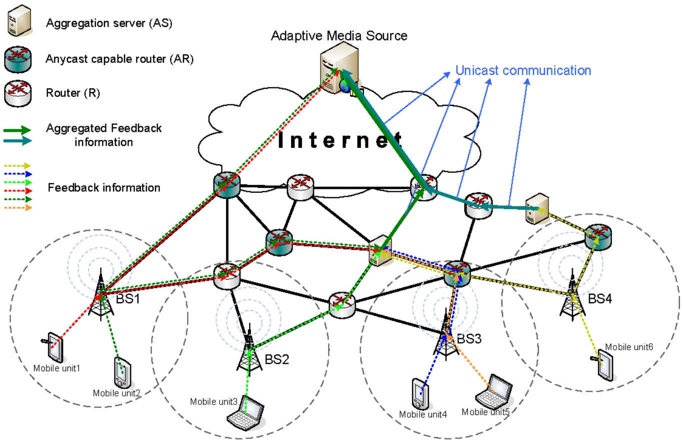

Fig. 1. Anycast based feedback aggregation architecture

The media source and the feedback aggregation servers are in the same anycast group, addresseable with the same anycast address, which is one of the unicast addresses of the media source [14]. This addressing architecture ensures that the packet is delivered to the proper destination even if it meets only with unicast capable routers on its path back to the source. IPv6 anycasting helps to reach the aggregation servers in an optimal way (Fig. 1): a terminal addresses the feedback packets to the anycast address of the aggregation servers, thus packets are delivered to the "closest" aggregation server (or directly to the media source if it is the closest member of the anycast group) through the Base Stations (BS) using anycast routing protocol (AOSPF, ARIP [8], etc.) which is implemented in the intermediate anycast capable routers (AR). Note, that it is not necessary that all of the routers are anycast capable: however, in this case, only near-optimal transmission of feedback data is achievable. Also note that in this network scenario the stateless property of anycast communication does not raise any problem, since the terminals send individual feedback packets and it makes no difference which aggregation server they are delivered to. Aggregation servers supported by anycast communication provide Network-level (or System-level) aggregation by collecting feedback information fragments from the mobile terminals in a near-optimal way and by aggregating and sending the collected feedback data to the media source. In accordance to the literature, the aggregation ratio in this level is determined by the tracking history length and the MTU on the aggregation servers uplink. On average an aggregation ratio between 2:1 and 10:1 can be achieved.

4 Evaluation

4.1 System Model

The following parameters are used in the analysis.

- The data link layer is regarded slotted. The width of the timeslot is given as τ_s. Extension to the continuous case is not necessary, since the system performance is very poor also in a perfectly slotted network, as shown later. For the sake of simplicity, let us assume that τ_s equals the signalling period.
- Parameter l_0 denotes the header length of the feedback packet. As long as we use IPv6 and UDP for the feedback messages, l_0 equals 384 bit (40 + 8 byte).
- Assuming that we have plenty of feedback categories,
  - l_i denotes the length of the *i*th feedback information, *i* starts from one.
  - $q_i - 1$ is the number of free timeslots between subsequent feedback messages. q_i is proportional to the frequency of the *i*th feedback information. $q_i = 1$ states that in every timeslot the *i*th feedback is sent from the clients, $q_i = 2$ states that every second timeslot is used for delivery, etc. Non-integer values are also allowed, but $q_i \geq 1$ (feedback cannot be more frequent than one per timeslot).
  - Both l_i and q_i are assumed to be constant (extension to the random case is not included here).
- There are N_c clients in the system. All these clients are identical and independent, generating feedback information as an i.i.d. process.
- There is only one Master Application Controller (stream server).
- Feedbacks are aggregated in special nodes which are called Feedback Aggregation Servers (FAS). There are N_{FAS} of them.

Let we investigate a particular timeslot in the network. The feedback traffic at the *k*th node (T^k) is given as

$$T^k = \left(\sum_i l_i \psi_{ik}\right) + l_0, \tag{4.1}$$

where ψ_{ik} is an indicator parameter: it equals one, if the *k*th client sends the *i*th feedback information in the investigated timeslot, and zero otherwise.

Now, the total traffic in the network is described. First, we assume that there is no feedback aggregation server in the network, so every client sends the feedback directly to the application controller. The traffic in the application controllers' network (T_{AC}) is given as

$$\begin{aligned} T_{AC} = \sum_k T^k &= \sum_k \left(\sum_i l_i \psi_{ik} + l_0\right) \\ &= \sum_i \left(l_i \sum_k \psi_{ik}\right) + N_c l_0. \end{aligned} \tag{4.2}$$

Note the exchange of sums in the above equation. Introducing a new variable

$$\eta_i = \sum_k \psi_{ik}, \tag{4.3}$$

we get

$$T_{AC} = N_c l_0 + \sum_i l_i \eta_i. \tag{4.4}$$

Now let us see the properties of η_i. It describes the total number of *i*th feedback messages in the network per timeslot. Since the clients are identical and independent, it is a binomial random variable with mean N_c/q_i and variance $N_c/q_i(1 - 1/q_i)$, and the distribution is

$$\Pr\{\eta_i = n\} = \binom{N_c}{n} \left(\frac{1}{q_i}\right)^n \left(1 - \frac{1}{q_i}\right)^{N_c - n}.$$

If N_c is large, η_i can be estimated by a normal random variable.

$$\eta_i \sim \mathcal{N}\left(N_c/q_i, N_c/q_i(1 - 1/q_i)\right). \tag{4.5}$$

Substituting (4.5) into (4.4), it turns out that the traffic at the application controller is a sum of normal random variables. That is, the traffic also follows a Gaussian distribution, with a mean that equals the sum of means, and variance which is equal to the sum of variances:

$$\mathbb{E}\{T_{AC}\} = N_c \left(l_0 + \sum_i \frac{l_i}{q_i}\right),$$

$$\mathbb{E}\left\{T_{AC}^2\right\} - \mathbb{E}\{T_{AC}\}^2 = N_c \sum_i \frac{l_i^2}{q_i}\left(1 - \frac{1}{q_i}\right).$$

$$T_{AC} \sim \mathcal{N}\left(N_c \left(l_0 + \sum_i \frac{l_i}{q_i}\right), N_c \sum_i \frac{l_i^2}{q_i}\left(1 - \frac{1}{q_i}\right)\right). \tag{4.6}$$

The bandwidth of the network is denoted by *B*. That is, in each slot, $B \cdot \tau_s$ bits can be pushed through the network. Now we will cover two basic cases.

4.1.1 The network router with infinite buffer

Here, the network's router has infinite buffer. It means that although more information arrives at the router, the excessive information above the bandwidth is buffered, so there is no packet loss. However, the delay of the packets gets longer and longer if the demand is higher than the available bandwidth. If the mean of the traffic is higher than the bandwidth, the average traffic will be also higher. This provides a natural limit for the number of clients. Substituting the mean and drawing strict upper bound with $B\tau_s$, one gets

$$N_c < \frac{B\tau_s}{l_0 + \sum_i \frac{l_i}{q_i}}. \tag{4.7}$$

This is an inequivality that must be fulfilled to avoid infinite delay of feedback information. This formula can be evaluated given the parameters of the system.

4.1.2 The network router with zero buffer

First, the network router has zero buffer length. It means that if more packets arrive than the maximum mass of packets the network can handle, the packets will be automatically lost. In other words, if the traffic is above the bandwidth of the network, packets will be lost. So, the probability of the packet loss can be evaluated by taking the tail of the Gaussian distribution, as

$$\Pr\{\text{feedback lost}\} = \frac{1}{2}\,\mathrm{erfc}\left(\frac{B\tau_s - N_c\left(l_0 + \sum_i \frac{l_i}{q_i}\right)}{\sqrt{2}N_c \sum_i \frac{l_i^2}{q_i}\left(1 - \frac{1}{q_i}\right)}\right), \tag{4.8}$$

which is never zero. Knowing the parameters of the system and the acceptable level of feedback information lost ratio (χ), one can find the maximum number of clients, N_c, which can be served in this environment as

$$N_c < \frac{B\tau_s}{\sqrt{2}\,\mathrm{erfc}^{-1}(2\chi) \sum_i \frac{l_i^2}{q_i}\left(1 - \frac{1}{q_i}\right) + l_0 + \sum_i \frac{l_i}{q_i}}. \tag{4.9}$$

Please note that if the q_i parameters equal one (there is no random effect in the system, every slot is used for feedbacks), the first term in the nominator becomes zero, thus we get back (4.7), the two cases are identical. However, if at least one q_i parameter differs from one, the random effect appears, the first term of the nominator will be different from zero, consequently (4.7) and (4.9) yield different bounds on N_c.

4.1.3 A numerical example

If the bandwidth of the application controller's network equals $B = 10\,\text{Gbps}$, and all the clients send three different feedback information every $\tau_s = 10\,\text{ms}$ ($q_1 = q_2 = q_3 = 1$), all of them together amount 31 bytes ($l_1 + l_2 + l_3 = 248\,\text{bits}$). Substituting these numbers into (4.7), it turns out that N_c must be lower than 158,228. That is, there could be approximately 150 thousand users in this system. This is indeed a very low value.

If the above parameters are changed a bit, $q_1 = q_2 = 1, q_3 = 2$, $l_1 = l_2 = 80\,\text{bits}$ and $l_3 = 176\,\text{bits}$, the (4.7) leads the same bound as before. However, (4.9) is now different: taking $\chi = 0.1$, which means that every tenth feedback packet can be lost (here we need $\mathrm{erfc}^{-1}(0.2) = 0.9062$), we get $N_c < 9473$. Less, than ten thousand users can be served in this case.

Now let us see what happens if the aggregation servers are switched on.

4.2 The aggregation servers switched on

The effect of the aggregation servers are twofold. First, the number of (IP+UDP) headers are significantly lowered due to the fact that many clients send their feedback to the aggregation servers, instead of the application controller. Then, aggregation servers send only one packet compared to the many they receive. Secondly, the feedback information can be compressed, so not all the feedback information must be sent back, probably some statistics (e.g. mean, variance, lowest, etc.) are sufficient for the application controller.

Aggregation servers could be arbitrary many in the network. Following our notations, the number of aggregation servers equals N_{FAS}.

For sure, aggregation should not introduce too much delay in the network, which yields that q_i parameters are the same after and before aggregation. (4.1) holds, but the input traffic of the jth feedback aggregation server is given as

$$T_{FAS}^j = \sum_{k\in U_j} T^k = \sum_{k\in U_j}\left(\sum_i l_i\psi_{ik} + l_0\right) = \sum_i \left(l_i \sum_{k\in U_j}\psi_{ik}\right) + |U_j|l_0, \tag{4.10}$$

where U_j is the set of users under the jth aggregation server: these are the users, whose traffic is “catched” by the jth aggregation server. As before, we introduce a new variable

$$\eta_i^j = \sum_{k\in U_j}\psi_{ik}, \tag{4.11}$$

which describes the total number of ith feedback messages at the input of the jth aggregation server. As before, since the clients are identical and independent, it is a binomial random variable with mean $|U_j|/q_i$ and variance $|U_j|/q_i(1 - 1/q_i)$, and if U_j is a large set, η_i^j can be estimated by a Gaussian random variable

$$\eta_i^j \sim \mathcal{N}\left(|U_j|/q_i, |U_j|/q_i(1 - 1/q_i)\right). \tag{4.12}$$

This random variable simplifies (4.10) as

$$T_{FAS}^j = |U_j|l_0 + \sum_i l_i\eta_i^j. \tag{4.13}$$

It turns out that the input traffic at the jth feedback aggregation server is a sum of normal random variables. That is, the input traffic also follows a Gaussian distribution

$$T_{FAS}^j \sim \mathcal{N}\left(|U_j|\left(l_0 + \sum_i \frac{l_i}{q_i}\right), |U_j|\sum_i \frac{l_i^2}{q_i}\left(1 - \frac{1}{q_i}\right)\right). \tag{4.14}$$

Up to this point we have the same equations as before, (4.14) looks the same as (4.6), however, here, the number of users can be set according to the positioning of aggregation servers. That is, the aggregation servers should be installed in such points of the network, that the total traffic described in (4.14) should not exceed a given value. Taking the numerical example again, we can say that every 150 thousand users (or in the second case every 10 thousand users) should have at least one feedback aggregation server.

The output of the feedback aggregation servers can be written as

$$T^{out}_{FAS} = l_0 + \sum_i c_i l_i / q_i, \tag{4.15}$$

where c_i is the compression constant, it defines how many feedback values are sent instead of all the values they receive to lower the network load. For instance, the minimum, the average, the deviation, and the maximum values (here $c_i = 4$) can be sent from the feedback aggregation server. Every feedback category should have its own c_i parameter. Please note that (4.15) is both deterministic and independent (it does not depend on the actual aggregation server).

Now let us see what happens at the input of the application controller. To make distinction from the case without aggregation server, we will denote this traffic as T'_{AC}. The traffic arriving at the application controller consists of the output traffic of the feedback aggregation servers, and the feedback traffic of those users which were not aggregated.

$$\begin{aligned} T'_{AC} &= \sum_{k\in U_0} T^k + N_{FAS} T^{out}_{FAS} \\ &= \sum_{k\in U_0} \left(\sum_i l_i \psi_{ik} + l_0 \right) \\ &\quad + N_{FAS} \left(l_0 + \sum_i c_i l_i / q_i \right) \\ &= \sum_i l_i \left(\sum_{k\in U_0} \psi_{ik} + N_{FAS} \frac{c_i}{q_i} \right) \\ &\quad + (|U_0| + N_{FAS})\, l_0, \end{aligned} \tag{4.16}$$

where U_0 is the set of the users whose traffic is not aggregated. It makes the traffic random (Gaussian) as detailed before.

Taking the numerical example of the previous section, and assuming the easiest case, where all the users can find one aggregation server, $c_1 = c_2 = c_3 = 4$, one can see that the maximum number aggregation servers can be $N_{FAS} = 72{,}674$. With this setup, 11.5 billion users can be served, if the network routers have infinite buffers. This is an acceptable number. Even for zero buffer network devices, and assuming 10 % acceptable packet loss ratio, the maximum number of users, which can be served with the help of the aggregation servers, equals 688 millions.

5 Conclusions and future work

The research presented in this paper mainly concerned the questions and challenges of feedback aggregation in jointly-optimized, cross-layer communication enabled networks. We introduced how feedback messages affect the Quality of Service on the receiver's side and what are the disadvantages of this feedback framework. Then we gave a short overview of data aggregation and IPv6 anycasting, just before we presented our solution for an efficient feedback aggregation method. Then we evaulated our solution with mathematical analysis and we have seen that without feedback aggregation, the number of users which can be served is very limited and unacceptable. Installing a few aggregation servers, the maximum number of clients that can be served goes up high to acceptable levels. Thus, feedback aggregation is a must, not just a possibility.

As a part of our future work we are planning to run simulations in different scenarios to confirm the results of the analysis presented in this paper. We are also planning to refine this feedback aggregation model for more optimized operation.

References

1 **Abdelatif M A, Kalebaila G K, Chan H A**, *A Cross-layer Mobility Management Framework Based on IEEE802.21*, Proceeding of The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07), posted on Sep, 2007, DOI 10.1109/PIMRC.2007.4394146, (to appear in print).

2 **Abley J, Lindqvist K**, *Operation of Anycast Services*, Dec, 2006. RFC 4786.

3 **Ata S, Kitamura H, Murata M**, *Possible Deployment Scenarios for IPv6 Anycasting*, Oct, 2004. IETF Internet Draft, draft-ata-anycast-deploy-scenario-01.txt.

4 **Bouras C, Gkamas A, Kioumourtzis G**, *Challenges in Cross Layer Design for Multimedia Transmission over Wireless Networks*, WWRF - 21st Meeting WG3 - Future Architecture (Oct, 2008).

5 **Deering S, Hinden R**, *Internet Protocol, Version 6 (IPv6) Specification*, Dec, 1998. RFC 2460.

6 **Doi S, Ata S, Kitamura H, Murata M**, *IPv6 Anycast for Simple and Effective Service-Oriented Communications*, IEEE Comm. Mag **42** (2004), 163-171, DOI 10.1109/MCOM.2004.1299362.

7 **Doi S, Ata S, Kitamura H, Murata M**, *Design, Implementation and Evaluation of Routing Protocols for IPv6 Anycast Communication*, Proceedings of the 19th International Conference on Advanced Information Networking and Applications - Volume 2, AINA '05 (March, 2005), 833 – 838.

8 **Doi S, Ata S, Kitamura H, Murata M**, *Design, implementation and evaluation of routing protocols for IPv6 anycast communication*, In Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05) (Mar., 2005), 833 – 838.

9 **Fan K W, Liu S, Sinha P**, *Structure-Free Data Aggregation in Sensor Networks*, IEEE Transactions on Mobile Computing **6** (2007), 929–942, DOI 10.1109/TMC.2007.1011.

10 **Fan K W, Liu S, Sinha P**, *Dynamic Forwarding over Tree-on-DAG for Scalable Data Aggregation in Sensor Networks*, IEEE Transactions on Mobile Computing **6** (2008), 1271–1284.

11 **Internet Engineering Task Force: Network Working Group**, *Requirements for Internet Hosts - Communication Layers*, Oct 1989. RFC 1122.

12 **Hashimoto M, Ata S, Kitamura H, Murata M**, *IPv6 Anycast Terminolgy Definition*, Jan, 2006. IETF Internet Draft, draft-doi-ipv6-anycast-func-term-05.txt.

13 **Hinden R, Deering S**, *IP Version 6 Addressing Architecture*, Dec, 1995. RFC 1884.

14 **Hinden R, Deering S**, *IP Version 6 Addressing Architecture*, Feb, 2006. RFC 4291.

15 **Hu W, Bulusu N, Jha S, Senac P**, *An Anycast Service for Hybrid Sensor/Actuator Networks*, UNSW-CSE-TR-0331, UNSW Computer Science and Engineering, Nov,2003.

16 **Huusko J, Vehkapera J, Amon P, Lamy-Bergot C, Panza G, Peltola J, Martini M G**, *Cross-layer architecture for scalable video transmission in wireless network*, Signal Processing: Image Communication **22** (2007), 317–330, DOI 10.1016/j.image.2006.12.011.

17 **ICT-OPTIMIX**, *ICT-OPTIMIX – Optimisation of Multimedia over wireless IP links via X-layer design*, Mar 2008. http://www.ict-optimix.eu.

18 **Intanagonwiwat C, Govindan R, Estrin D**, *Directed diffusion: A scalable and robust communication paradigm for sensor networks*, In Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 2000) (Aug 2000).

19 **ISO/IEC**, *Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model*, Nov, 1994. ISO/IEC 7498-1.

20 **IST-PHOENIX**, *IST-PHOENIX – Jointly optimising multimedia transmissions in IP based wireless networks*, Jan 2004. http://www.ist-phoenix.org.

21 **Lamy-Bergot C, Huusko J, Martini M G, Amon P, Bergeron C, Hammes P, Jeney G, Ng S X, Panza G, Peltola J, Sidoti F**, *Joint optimization of multimedia transmission over an IP wired/wireless link*, Proceedings of EuMob Symposium (September, 2006).

22 **Madden S, Franklin M J, Hellerstein J M, Hong W**, *TAG: a tiny aggregation service for ad-hoc sensor networks*, SIGOPS Oper. Syst. **36** (2002), 131–146.

23 **Madden S, Szewczyk R, Franklin M J, Culler D**, *Supporting aggregate queries over ad-hoc wireless sensor networks*, Proceedings of WMCSA ’02, posted on June, 2002, 49 – 58, DOI 10.1109/MCSA.2002.1017485, (to appear in print).

24 **Matsunaga S, Ata S, Kitamura H, Murata M**, *Design and Implementation of IPv6 Anycast Routing Protocol: PIA-SM*, Proceedings of the 19th International Conference on Advanced Information Networking and Applications - Volume 2, AINA ’05, posted on March, 2005, 839 – 844, DOI 10.1109/AINA.2005.151, (to appear in print).

25 **Partridge C, Mendez T, Milliken W**, *Host Anycasting Service*, Nov,1993. RFC 1546.

26 **Piri E, Sutinen T, Vehkapera J**, *Cross-layer Architecture for Adaptive Real-time Multimedia in Heterogeneous Network Environment*, Proceeding of European Wireless ’09 conference, posted on May, 2009, DOI 10.1109/EW.2009.5357979, (to appear in print).

27 **Postel J**, *User Datagram Protocol*, Aug,1980. RFC 768.

28 **Postel J**, *Transmission Control Protocol*, Sep,1981. RFC 793.

29 **Pottie G J, Kaiser W J**, *Embedding the internet: wireless integrated network sensors*, Communications of the ACM **43** (2000), 51-58.

30 **Solis I, K. Obraczka K**, *The impact of timing in data aggregation for sensor networks*, Proceedings of ICC ’04, posted on June, 2004, 3640 – 3645, DOI 10.1109/ICC.2004.1313222, (to appear in print).

31 **Yue Wang, Li Zhang, Zhinan Han, Wei Yan**, *Anycast Extensions to OSPFv3*, Proceedings of the 11th International Conference on Parallel and Distributed Systems - Volume 01 (July, 2005), 223 – 229.

32 **Zhiguo Zhou, Gaochao Xu, Jinxin He, Jianhua Jiang, Chunyan Deng**, *Research of Secure Anycast Group Management*, Proceedings of Fourth International Conference on Networked Computing and Advanced Information Management, posted on Sept, 2008, 604 – 608, DOI 10.1109/NCM.2008.33, (to appear in print).