Active Measurement of the Available Transfer Rate Used in an Algorithm for Generalized Assignment Problem

Iustin Alexandru Ivanciu, Andrei Bogdan Rus, Virgil Dobrota
Communications Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
{Iustin.Ivanciu,Bogdan.Rus, Virgil.Dobrota}@com.utcluj.ro

Jordi Domingo-Pascual
Computer Architecture Department
Universitat Politecnica de Catalunya
Barcelona, Spain
jordi.domingo@ac.upc.edu

***Abstract*— This paper proposes a solution for active measurement of the Available Transfer Rate needed in a multi-tunnel architecture with a smart mobile router offering uninterrupted services for its wireless customers. It simultaneously uses several wireless/mobile access networks to the Internet in order to reach the service continuity gateway located somewhere in a broadband network. This customer-oriented approach uses the measurements to select the tunnels between the router and the gateway. Furthermore an algorithm for solving the Generalized Assignment Problem minimizes the costs. The allocation decision must be taken once every second in order to obtain the minimum possible cost.**

Keywords—Available Transfer Rate, Generalized Assignment Problem.

I. INTRODUCTION

Nowadays, once access networks evolved, the terminals experiencing mobility have a variety of technologies available (Wi-Fi, WiMAX, GSM, 3G, 4G/LTE etc.). Their costs depend on the business plan of the operators (starting from free wireless access to very expensive 4G/LTE). From another perspective none of the technologies have 100% coverage of all the areas where the customer may travel. Emergency situations or public transportation may thus require seamless connectivity. It means that the customer should be able to preserve its connections to service providers, whilst the parameters of the access networks may vary significantly over time. Paper [1] investigates the advantages of simultaneously using multiple interfaces. Moreover, the authors propose a network layer architecture providing bandwidth aggregation, reliability support, resource sharing, data and control plane separation to the end users.

The solution is transparent to applications and it involves minimum changes in the infrastructure, namely the deployment of proxies and mobile clients. The architecture enables bandwidth aggregation for real-time applications thus increasing performance over single interface use. In [2] the authors focus on the best link selection for TCP connections based on the conditions over the end-to-end path, namely the RTT. An on-the-fly RTT probing mechanism is proposed. This mechanism duplicates the SYN packet during the three-way handshaking stage of a TCP connection and retrieves the end-to-end RTT through each access link. Next, the link from the lowest RTT path is selected. Results show that this probing mechanism is enough to select an ideal link even as it ignores the packet loss ratio and only one measurement of the RTT is taken into account when selecting a link. Also, this algorithm outperforms the traditional Round Robin link selection. The main disadvantage of this method is that it only works for TCP connections. In [3], the authors study statistical multiplexing for bursty individual connections which have transmission rates that vary greatly over time. Their model has two communicating nodes, connected by a set of parallel edges. The rate of each connection between the nodes is a random variable. Three related problems are studied: stochastic load balancing, stochastic bin-packing and stochastic knapsack. In the first problem, the number of links is given and the goal is to minimize the expected value of the maximum load. For stochastic bin-packing each connection needs to be assigned to a link using as few links as possible. The stochastic knapsack only allows one link which needs to accept as many connections as possible. The approximation algorithms use the notion of effective bandwidth – a means of associating a fixed demand with a bursty connection that replicates its distribution as closely as possible as well as a new definition introduced by the authors. A combination of the two measures is shown to provide good results when computing the bounds of the optimal solution.

None of the existing approaches investigated responded to the need of seamless connectivity in a transportation and/or emergency system involving the mobility of the end users. This paper uses a multi-tunnel architecture with a mobile router offering uninterrupted services for its wirelessly connected customers, proposed by the authors in [4]. The mobile router simultaneously employs several access networks (IEEE 802.11, 3G, 4G/LTE) to the Internet in order to reach a service continuity gateway SCG located somewhere in a broadband network. It is out of the scope of this paper to discuss how this tunnel-based solution was implemented. Herein we are focused on the intelligent selection of the

This paper is supported by the Sectorial Operational Programme Human Resources Development POSDRU/159/1.5/S/137516 financed from the European Social Fund and by the Romanian Government.

978-1-4799-7267-8/14/$31.00 ©2014 IEEE

tunnels depending on active measurements of the Available Transfer Rate (ATR) between the router and the gateway. This whole process is transparent to the infrastructure operators.

The rest of the paper is organized as follows. Section II describes the active measurement issues related to ATR. The next section explains how an algorithm for solving the well-known GAP (Generalized Assignment Problem) could be used by a mobile router with multi-tunnel access in order to minimize the costs of delivering the flows. The experimental results in Section IV prove that seamless connectivity without provider intervention is feasible for the proposed solution. Conclusions and future work end the paper.

II. Active Measurement of the Available Transfer Rate

This section presents the implementation of an estimation software tool called ATRAM (Available Transfer Rate Active Measurement), based on active probing and Kalman filtering. This tool was needed because there are no available tools to provide good results on both wired and wireless technologies. To the best of our knowledge, a tool called BART [5] might meet these requirements, but its commercial version is not available for the general public.

The tool allows the real-time estimation of the end-to-end Available Transfer Rate, consisting on two modules: the PTS (Probe Traffic Sender) and the PTR (Probe Traffic Receiver). The first one transmits sequences of probe frames through the network to the second one. Different sequences, depicted with black and white in Fig. 1 are transmitted at different rates. Here, the timestamp information and the probe frame rate are extracted and the average inter-frame strain is computed. Next, the Kalman filter uses this information to estimate the Available Transfer Rate.

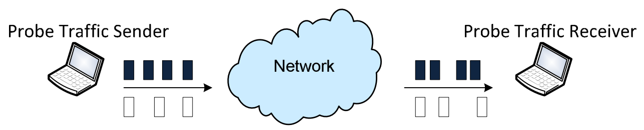

Fig. 1. ATRAM architecture

A. Probe Traffic Sender Module

The task of the Probe Traffic Sender module is to generate probe traffic. Probe frames are organized into sequences of frame pairs, transmitted at a certain rate *u*. Each frame contains information about the rate *u* and the inter-frame time gap corresponding to this rate. The block diagram of the PTS module is depicted in Fig. 2. The User Input component enables the selection of several parameters. These parameters refer to the network connection, e.g. the hostname of the PTR module and the port it is listening on. Other parameters are related to the range of values used for generating the rate *u*.

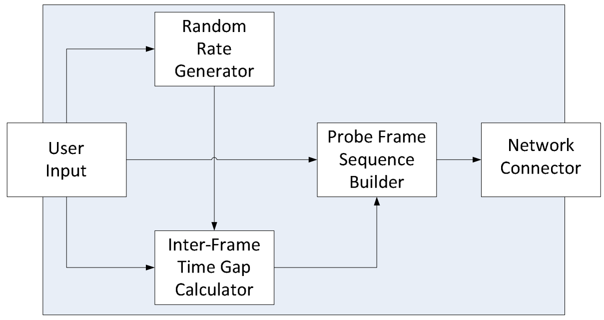

Fig. 2. Probe Traffic Sender module

Based on the information described above, the Random Rate Generator produces different rates for the probe traffic and forwards them to the Inter-Frame Time Gap Calculator. This component calculates the inter-frame time gaps that correspond to these rates. Once all the computations are performed, the Probe Frame Sequence Builder starts creating the probe frame sequences and sends them through the network path using the Network Connector. The Probe Traffic Sender module was implemented using the C programming language. The first step was designing the probe frame sequences. Authors in [5] show that the quality of estimation increases as the probe frame and probe sequence length increase. However, the main goal is to provide good quality estimations without introducing too much probing traffic in the network. The same authors state that a sequence of 17 frames is enough to obtain satisfying results. Therefore a default value of 17 frames was selected for the probing sequence. The probing sequences are generated once every second. The probe frames must be sent at different rates *u*. The range of values for these rates should be selected according to the type of connection, e.g. for Fast Ethernet a valid range would be 1-100 Mbps. In order to obtain accurate estimates and provide satisfactory tracking of the changes in the ATR, these rates should be generated randomly in order to cover the whole selected range. Once the rate *u* is obtained, the next step is to make sure that the probing frames are sent exactly at this intended rate. This is achieved by controlling the inter-frame time gap, depicted in Fig. 3.

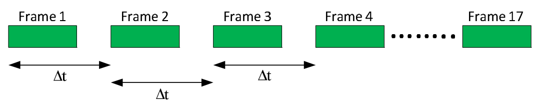

Fig. 3. The structure of a probing frame sequence

The inter-frame time gap is denoted with Δt and represents the time amount between the moment when the first byte of a probing frame is transmitted and the moment when the first byte of the next probing frame is transmitted. Considering the length of a frame *L* [bytes], and the desired probing rate *u* [Mbps], the formula to compute the inter-frame time gap is:

$$\Delta t = 8L/u [s] \quad (1)$$

B. Probe Traffic Receiver Module

The task of the Probe Traffic Receiver module is to capture the probe traffic and to extract the timestamp information of probe frames. This information, along with the rate at which the probe sequence was generated and the inter-frame time gap corresponding to this rate is then further processed in order to obtain an estimate of the Available Transfer Rate. The block diagram of the Probe Traffic Receiver module is depicted in Fig. 4. A description of these components and their interactions is summarized herein. Similar to the Probe Traffic Sender Module, the User Input component enables the selection of the port number on which the module is listening. Moreover, the user may change some parameters involved in the Kalman filtering process.

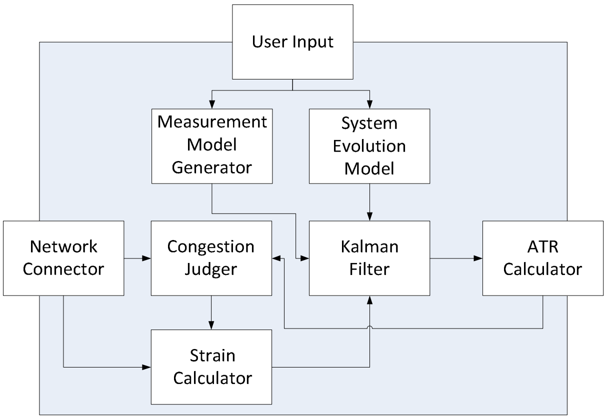

Fig. 4. Probe Traffic Receiver module

As previously mentioned, the Probe Traffic Receiver module extracts the timestamp information, the probe rate u and the time gap Δt used for the probing sequence. Next, these values are used by the Strain Calculator to compute the inter-frame strain for each pair of probe frames in the sequence. The same component also calculates the average inter-frame strain ε. The Strain Calculator then forwards the average inter-frame strain to the Kalman Filter, which performs the real-time estimation of the Available Transfer Rate. In order to work properly, the filter needs to know the expected evolution of the system between two consecutive measurements. This information is provided by the System Evolution Model component which has a static configuration. The Measurement Model Generator and the System Evolution Model have a high influence on the accuracy of estimations. This is because they control the filter parameters known as the covariance of the measurement noise and process noise, respectively, which in turn affect the system-state tracking characteristics of the Kalman filter. After all the computations are performed, the output is fed into an ATR calculator which updates the Available Transfer Rate. This value represents the output of ATRAM. The updated one is also sent to the Congestion Judger block, which makes an approximation of whether the used probe traffic rate u is high enough to overload the bottleneck link of the network path. In ATRAM, it is considered that congestion occurred if the used probe traffic rate is higher than the current estimate of the ATR. If this is the case, the Strain Calculator provides the Kalman filter with the average inter-frame strain. However, if the used probe traffic rate is lower than the current estimated value, the ATR will not be updated.

III. Algorithm for Generalized Assignment Problem

The Generalized Assignment Problem can be described using the terminology of knapsack problems. Given n items and m knapsacks with: p_{ij} = profit of item j assigned to knapsack i; w_{ij} = weight of item j if assigned to knapsack i; c_i = capacity of knapsack i, assign each item to exactly one knapsack so as to maximize the total profit, without assigning to any knapsack a total weight greater than its capacity:

maximize

$$z = \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} x_{ij}$$

subject to:

$$\sum_{i=1}^{m} w_{ij} x_{ij} \le c_{i,}\ i \in M = \{1, \dots, m\},$$

$$\sum_{i=1}^{m} x_{ij} = 1,\ j \in N = \{1, \dots, n\},$$

$$x_{ij} = 0\ or\ 1, i \in M, j \in N,$$

where

$$x_{ij} = \begin{cases} 1\ if\ item\ j\ is\ assigned\ to\ knapsack\ i, \\ 0\ otherwise. \end{cases} \tag{2}$$

The problem is frequently described in literature as that of optimally assigning n tasks to m processors, n jobs to m agents given the profit p_{ij}, the amount of resource w_{ij} corresponding to the assignment of task j to processor i and the total resource c_i available for each processor i. [6]

In our scenario, the items are represented by flows, while the knapsacks are represented by the different tunnels associated to the operators. The capacity of each knapsack is equal with the ATR of the tunnel while the weights of the items are equal with the rates of the flows. The problem is to assign each flow to a single tunnel in order to minimize the total cost of sending the flows. The problem was solved using Matlab function *bintprog*, a linear programming (LP)-based branch-and-bound algorithm to solve binary integer programming problems.

$$min_x f^t x, such\ that \begin{cases} A * x \le b, \\ Aeq * x = beq \\ x\ binary \end{cases} \tag{3}$$

Note that f, b, and beq are vectors, whilst A, and Aeq are matrices. The solution x has to be a binary integer vector, having as entries values of 0 or 1. In our case, each element of x represents a flow associated to a tunnel. If the flow is associated to the tunnel, the variable has the value 1, else the variable has the value 0. If there are n flows and m tunnels, the first *m* elements of *x* correspond to *flow 1* being assigned to *Tunnel 1*, *Tunnel2 ... Tunnel m*. The next *m* elements of *x* correspond to *flow 2* being assigned to *Tunnel 1*, *Tunnel 2 ... Tunnel m* etc. In total, *x* has *n x m* elements. In order to make sure tunnel capacity is not exceeded, the *m* elements of the column vector *b* must be set to the values of the ATR corresponding to each of the *m* tunnels. The matrix *A* is composed of *n m x m* diagonal matrices side by side. The elements on the main diagonal of matrix 1 represent the rates of *Flow 1*, for matrix 2 they represent the rates of *Flow 2*, etc. The other set of constraints requires that each flow is associated to exactly one tunnel. In other words, for each flow, the sum of the *x* values corresponding to that flow is exactly one. These linear constraints can be represented by building the appropriate matrices, as follows. *Aeq* is an *n x (n x m)* matrix. Line *i* of this matrix has the elements from *m x (i-1) + 1* to *m x* i in 1 and all other elements in 0. *beq* is a column vector of *n* values of 1. In order to minimize the cost of sending the flows we must set an appropriate column vector *f*. In our case, f has *n x m* elements. The first *m* elements of *f* correspond to the cost of sending *Flow 1* through *Tunnel 1*, *Tunnel 2 ... Tunnel m*. The next *m* elements of *x* correspond to *Flow 2* being sent through *Tunnel 1*, *Tunnel 2 ... Tunnel m etc.*

IV. Experimental Results

In order to demonstrate the feasibility of our solution we built a testbed with a smart mobile router SMR, performing vertical handover VHO and load balancing LB [4]. It had two wireless and one mobile links to the Internet, in order to reach a service continuity gateway SCG located somewhere in a broadband network. Suppose Wi-Fi1 (Tunnel 1) is charging 0.2 € cents/MB and Wi-Fi2 (Tunnel 2) is free. A 3G mobile operator (Tunnel 3) is charging twice than Wi-Fi1, i.e. 0.4 € cents/MB.

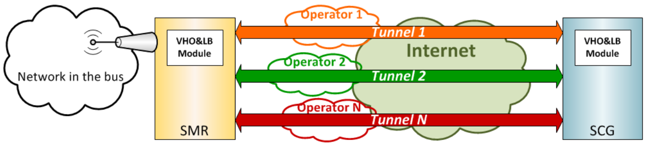

Fig. 5. Testbed scenario

Depending on how fast we move one flow from one tunnel to another, minor freezing of the service may occur. However this does not mean that the service is interrupted. For instance the TCP connections between SCG and the destination are kept, whilst just the access network from SMR to SGC is modified. We considered a realistic scenario, where 3G access is available anytime, anywhere within the testbed, whilst both Wi-Fi1 and Wi-Fi2 may not be always available. The experiments tried to answer to the question related to the frequency of applying algorithm for GAP. Suppose three flows have to be sent through these tunnels with the average rates: Flow 1: video (1517 kbps), Flow 2: Skype call (41 kbps), Flow 3: data (218 kbps). The results in Fig. 6 and Table I proved that the allocation decision must be taken once every second in order to obtain the minimum possible cost.

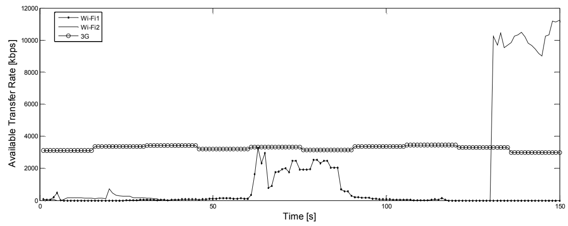

Fig. 6. Available Transfer Rate for each tunnel during the experiments

TABLE I. Cost of the tunnel-based solution vs. 3G-only based one

Flow	Time spent in Tunnel1 [s]	Time spent in Tunnel2 [s]	Time spent in Tunnel3 [s]	Data sent through Tunnel1 [MB]	Data sent through Tunnel2 [MB]	Data sent through Tunnel3 [MB]	Tunnel-based solution cost [€ cents]	3G-only based solution cost [€ cents]
1	21	20	109	3.22497	26.79500	20.70987	8.92894	10.85162
2	76	48	26	0.34153	2.14000	0.13602	0.12271	0.29306
3	30	27	93	0.77558	5.88800	2.43115	1.12758	1.56345
Total				4.34208	34.82300	23.27704	10.17923	12.70814

V. Conclusions and Future Work

The proposed algorithm for GAP minimizes the costs in a tunnel-based solution, making the customers to switch automatically from one operator to another, without service interruptions. Future work is focused on active measurements of end-to-end delay or energy consumption, as sensor networks are expected to be connected to the smart mobile router too.

References

[1] K.Chebrolu, and R.R. Rao, "Bandwidth aggregation for real-time applications in heterogeneous wireless networks". Mobile Computing, IEEE Transactions on, 5(4), 2006, pp. 388-403.

[2] Y.D. Lin, S.C. Tsao and U.P. Leong, "On-the-fly TCP path selection algorithm in access link load balancing". Computer communications, 30(2), 2007, pp. 351-357.

[3] J. Kleinberg, Y. Rabani and E. Tardos, "Allocating bandwidth for bursty connections". SIAM Journal on Computing, 30(1), 2000, pp. 191-217.

[4] FP7-UCONNECT 2012-2014, http://idi.gowex.com/uconnect/

[5] M. Li, M. Claypool and R. Kinicki, "WBest: a Bandwidth Estimation Tool for IEEE 802.11 Wireless Networks", Proc. 33rd IEEE LCN, Montreal, Canada, October 2008, pp.374-381.

[6] S. Martello and P. Toth, Knapsack Problems: Algorithms and Computer Implementations. John Wiley and Sons, 1990.