Computer Networks and Distributed Systems

IAM-02-004

November 2002

Computer Networks and Distributed Systems

Summer School of the
”Computer Networks and Distributed Systems” research group
Institute of Computer Science and Applied Mathematics
University of Berne

August 26-30
Vira-Gambarogno Ticino, Switzerland
http://www.iam.unibe.ch/~rvs/events/ss2002/

Abstract

The research group on ”Computer Networks and Distributed Systems” of the Institute of Computer Science and Applied Mathematics at the University of Berne led by Prof. Torsten Braun focuses the research activities in the areas of mobile and multimedia communications as well as on distance learning. In summer 2002, a seminar has been organized in Ticino with the goal to present and discuss intensively the state of the research work performed by the research group’s Ph.D. students. External international experts working in related research areas have been invited in order to contribute to the discussions and to present their current and future research areas. Each speaker had 90 minutes time for his presentation including discussion. The overall results have been very positive. In particular, the discussions have been very intensive and productive and should be valuable for the Ph.D. students future work. This reports summarizes the various talks from research group members and external experts.

CR Categories and Subject Descriptors: C2.1 [Computer-Communication Networks]: Network Architecture and Design; **C2.2 [Computer-Communication Networks]:** Network Protocols; **C2.5 [Computer-Communication Networks]:** Local and Wide-Area Networks; **C2.6 [Computer-Communication Networks]:** Internetworking.

General Terms: Algorithms, Design, Performance, security.

List of Presentations

1. Routing in Large Scale Mobile Ad Hoc Networks
   Marc Heissenbuettel, University of Berne
2. GSM Assisted Adhoc/P2P Networking and Service Provisioning
   Marc Danzeisen, University of Berne
3. Crypto-Based Identifiers and their applications
   Claude Castelluccia, INRIA Grenoble, France
4. Market Management of Peer-to-peer Services
   Burkhard Stiller, Univ. of Federal Armed Forces Munich, Germany and ETH Zurich
5. Charging and billing schemes for wireless networks
   Attila Weyland, University of Berne
6. eXperience Based Admission Control: Lowering the measurement overhead in MBAC
   Matthias Scheidegger, University of Berne
7. Signaling and Monitoring Resource Reservations in DiffServ Networks
   Florian Baumgartner, Pardue University
8. Self-organizing networks
   Torsten Braun, University of Berne
9. Active components for QoS and multicast setup
   Roland Balmer, University of Berne
10. Protocol Metabolism
    Christian Tschudin, Uppsala University, Sweden
11. Hierarchical Bandwidth Brokers Providing QoS-aware Handovers for Mobile IP
    Guenther Stattenberger, University of Berne
12. Mobile applications in ad-hoc networks: examples and improvements
    Stefan Fischer , Technical University Braunschweig, Germany
13. Approaches for improving TCP in wireless mobile ad hoc networks
    Ruy de Oliveira, University of Berne
14. Technical and didactical issues of a distance learning architecture
    Marc Steinemann, University of Berne

Aknowledgements: This event has been partly supported by Burgergemeinde Bern.

Ants-Based Routing in Large Scale Mobile Ad-Hoc Networks

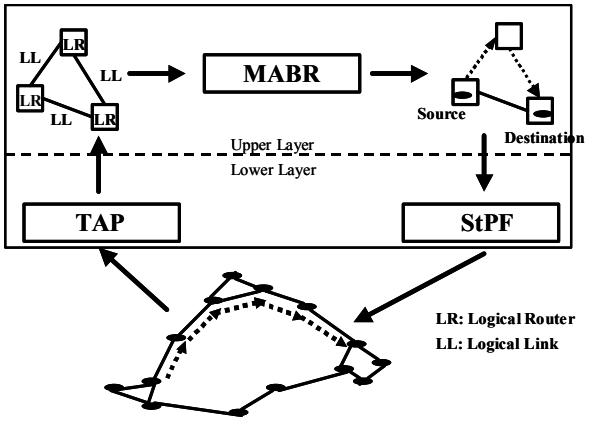
M. Heissenbüttel

The interest in MANETs has grown recently and many novel routing protocols were developed that deal with the special characteristics of such networks. Proactive routing protocols attempt to maintain at all times routing information from each node to every other node in the network (e.g. DSDV [1]), whereas reactive protocols only acquire routes on demand (e.g. AODV [5]). Obviously, both categories will not scale with large scale MANETs as considered in this paper. The signaling traffic constantly present in the network for proactive protocols is substantial because update messages are propagated throughout the network for any changes in the topology. Reactive protocols often employ a kind of flooding to acquire and sustain routes. The overhead induced becomes a serious limitation as well. Other approaches make use of location information (e.g. GPSR [9]). A packet is sent to a neighboring node, which reduces the distance to the destination. For the case that this greedy, simple method fails, different fallback mechanisms are proposed. These mechanisms could result in inefficiencies for scenarios where routing along the line-of-sight between source and destination nodes is often impossible. Further, those approaches do not consider routing around congested areas or where links suffer from poor quality, i.e. they normally only take only location information into account for routing decisions and not other criteria like, e.g., network load, congestion, available bandwidth.

Lately, a new kind of completely distributed routing protocols for fixed, wired communication networks evolved inspired from social insects (ants) behaviour (e.g. AntNet [15]). Ants were shown to find shortest paths through a process called "stigmergy", which could be described as indirect communication between individuals through the environment. Ants returning from a food source to the nest lay down pheromones (a chemical substance) behind them. Other ants are attracted by these pheromone trails and in turn reinforce them even more. As a result of this "auto-catalytic" effect, the shortest path will emerge rapidly. This process has been adapted for telecommunication networks. Current traffic conditions and link costs are measured by transmitting "artificial ants" (mobile routing agents) into the network, which not only update routing tables depending on the collected information, but as well mark the traveled path with an "artificial pheromone". These protocols showed very promising results and turned out to be highly adaptive in dynamic network environments.

The characteristics discussed above seem to make this family of protocols a priori well suited for MANETs. To our knowledge, this paper describes the first approach that tries to adapt ants-based routing for the scenario of MANETs. The proposed two-layered concept introduces different new protocols, of which each tackles a different problem. TAP (Topology Abstracting Protocol) will be used to supply in a transparent manner a simplified topology with fixed "logical routers" and fixed "logical links".

(We will use the term “logical” in general to indicate that we mean paths, links, etc. in the upper layer.) In t his context, the term logical router represents an aggregated collection of mobile hosts, which all together build and share among each other the same routing tables. A logical link represents a path along a roughly straight line to a distant logical router over possibly multiple hops. On top of this abstract topology the actual routing protocol MABR (Mobile Ants Based Routing) will be run. This ants-based protocol is responsible for updating the routing tables of logical routers and determining logical paths for routing packets over this abstract topology. Finally, the SPF (Straight Packet Forwarding) protocol is applied in order to transmit packets over a logical link. Therefore, it forwards packets along this logical link in a greedy manner. An overview of the architecture with the interactions between the protocols is depicted in the figure below.



The work in progress includes the elaboration of the protocols and the consideration of additional mechanisms to make the protocols even more suitable for MANETs. These include the aggregation of pheromone trails and other hierarchical concepts to obtain fewer long distance logical links. Different routing schemes for ants and data need to be studied in more detail. Furthermore, a first version is being implemented on a network simulator to verify the performance of this concept empirically.

References

[1] Perkins, C, et al.: Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. Proceedings of the SIGCOM '94 (1994)

[5] Perkins, C., Royer, E.: Ad-Hoc On-Demand Distance Vector Routing. Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications (1999)

[9] Karp, B., Kung, H.: GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom) (2000)

[15] Di Caro, G., Dorigo, M.: Antnet: Distributed stigmergetic control for communications networks. Journal of Artificial Intelligence Research. 9 (1998)

Adhoc Services with Cellular Billing Infrastructure

Marc Danzeisen

Today's mobile devices have Bluetooth hardware integrated. But there are only a few applications enabled like serial communication for sending information between the computer or PDA and the mobile phone to use its modem functionality or to synchronize the address book and agenda. Although the Bluetooth protocol stack was designed to be a general communication tool were future applications should be easily integrated in form of profiles, without the need of modification of the Bluetooth component, we are missing a base set of implementations of these profiles today. The basic components of Bluetooth hardware and protocols have been made to meet the requirements of an adhoc networking device. Mechanisms like inquiry and paging allow Bluetooth enabled devices to explore their environment and act in an autonomous way to establish communication channels to other devices in reach. In combination with service detection algorithms and service definitions implemented in the Service Detection Protocol even services can be detected without the need of user intervention. This behavior of environment exploration will influence the market for location based services like information distribution by sightseeing or timetable broadcasting at train stations, but also newspaper download at the kiosk. Adhoc services like these will occur wherever electronic information has to be transferred to a mobile device like a PDA.

One of the main problems in the domain of adhoc and peer-to-peer communication is the authentication of the mobile nodes. Without centralized authentication services like Certificate Authorities (CA) or Home Locator Register (HLR) it seems to be very hard to authenticate an opposite device or user which is not registered or does not have a shared secret in common. And without authentication paid adhoc services will never be available.

Today, the mobile operator is the only instance that can offer nearly every where access to its authentication servers. The mobile users accept the mobile operator as *trusted operator* [1] and so the operator should offer authentication services also for financial transactions.

This would not only allow the operator to enter a new market, but also harden the customer binding. Such an authentication service would allow the mobile users to make transactions of small amounts without the need of an additional device. Daily buys like mentioned above could be charged on the user's mobile phone bill.

First services like this are already available and have proven the technical feasibility. The possibility of paying snacks with the mobile phone, for example, is one of them. Bluetooth adhoc features will additionally allow to communicate with local servers in a secure and fast way.

The *Figure 1* shows an example how the current GSM system could be reused for securing and billing peer-2-peer transactions. Therefore the GSM authentication centre (AuC) of the home provider acts as a kind of certificate authority and signs the transaction data like the MSISDN numbers of the involved parties and the Bluetooth PIN used to encrypt the peer-2-peer communication. Normally, this PIN (or shared secret) has to be inserted manually into both devices during the so-called Bluetooth pairing. If this pairing can happen over the GSM network, it could be done in an automatic way. In the case of purchase of a mp3 music file, for example, the cashbox of the vendor, which is connected to the GSM network generates a random Bluetooth PIN and sends it with the buyer's MSISDN and the price to the home operator server of the buyer (*step 1*). This server gets the appropriate Ki of the buyer, based on the MSISDN, and signs the PIN with it. This signed PIN is then sent back to the seller (*step 2*). The buyer only needs the unsigned PIN from the server (*step 3*), signs it with its own Ki stored on the SIM card and transmits it over Bluetooth to the seller (*step 4*). To protect the PIN, only the hashed PIN is transferred, since the original PIN is proposed and therefore known by the seller. This way, the seller can compare both signed PINs and in the case of equivalence transfer the mp3 file over the secured Bluetooth channel. The buyer is accepting the price by sending the signed PIN over Bluetooth.

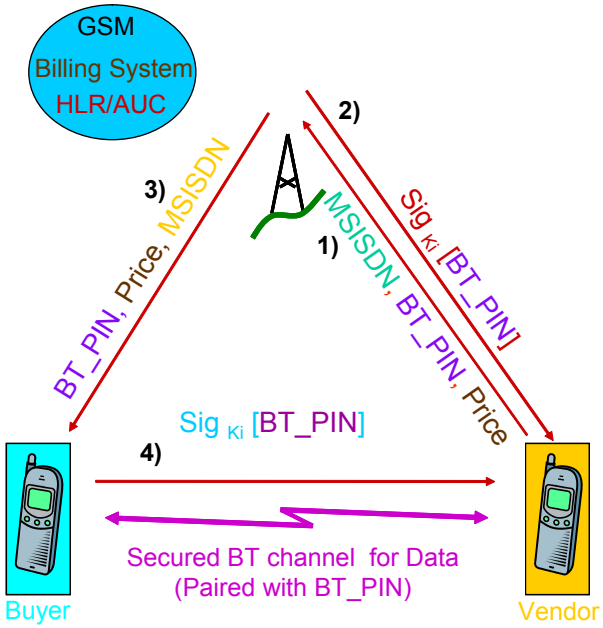


Figure 1: Trusted Operator

In combination with PDAs this secure adhoc channel will extend the usability of public infrastructure like the Ticket-Machine at the train station, where the user interface could then be realized on the screen of the PDA. So the bottleneck of all these services, namely the limited user interface, can be bypassed and the ticket can even be billed by the mobile operator *Figure 2*. An association of mobile operators called *Mobile Payment Forum* [1] was created to

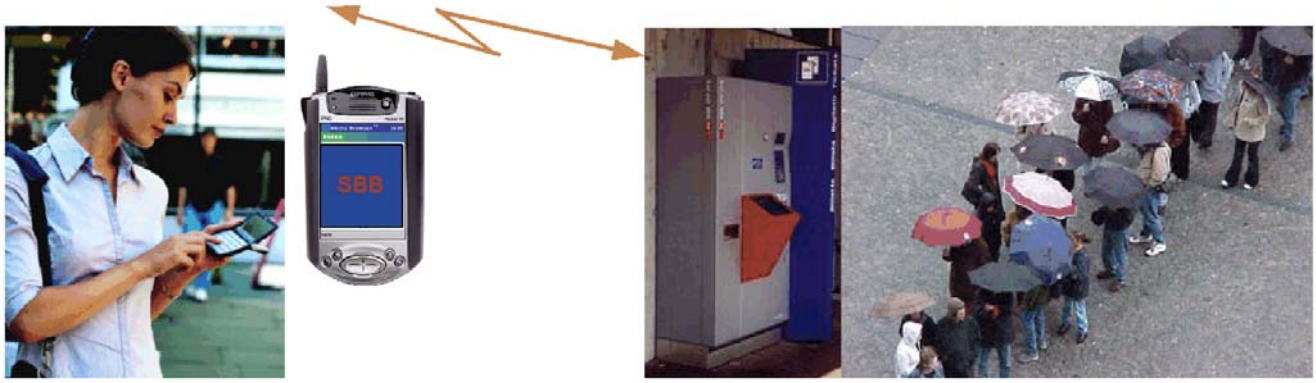


Figure 2: Adhoc Services enabled by Bluetooth

focus on such new business opportunities.

With future Bluetooth profiles like the Cordless Telephony Profile (CTP) the public phone will overcome its limitation to one user at the same time. Mobile users of other operators could access the PSTN with their own mobile device.

References

[1] Mobile Payment Forum, http://www.mobilepaymentforum.org/

Dynamic and Secure Group Membership in Ad Hoc and Peer-to-Peer Networks

Claude Castelluccia[a] ***Gabriel Montenegro***[b]
claude.castelluccia@inrialpes.fr *gab@sun.com*
[a]*INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 Montbonnot, France*
[b]*Sun Labs, Europe, 29, chemin du Vieux Chêne, 38240 Meylan, France*

1 Introduction

Ad hoc or peer-to-peer networks (called *impromptu* networks henceforth) pose many problems with respect to securing their highly dynamic structures. A naive approach assumes any given node can trust all other nodes in the impromptu network for any type of operation (for example, engaging in some cooperative and perhaps confidential activity). This paper improves on previous efforts to secure group authorization (including membership). We do so by employing crypto-based identifiers [6] for node and group identification, and then use these in authorization certificates. These allow groups (or nodes) to authorize nodes (or other groups) [3].

Our approach enables highly flexible and robust impromptu security services in an inherently distributed fashion. Previous work on securing impromptu networks has assumed the existence of a traditional PKI, of some web of trust or of some mechanism to distribute keys and shared secrets. We believe these assumptions are unrealistic in impromptu networks.

2 Secure Node Identity

First of all, we must start by defining an addressing model. Previous efforts for ad hoc networks conclude that since there is no aggregation to the degree possible with regular fixed networks, addressing can be more flexible. We propose to use pure identifiers with no topological meaning.

Our scheme improves upon these by having an implicit cryptographic binding between a node's identifier and its public key (or certificate). A node autoconfigures its (crypto-based) identifier (CBID) by doing the following:

- Create a pair of public and private keys (PK and SK).
- Create its CBID: $CBID = hash(PK)$.

Note that the hash function can be applied over more than just the public key (e.g. a salt or some other values) [6]. Given the secure correspondence between identity and public key, the latter can be communicated by the node itself. This simplifies key management, since no third parties need to be involved either in creating or distributing the public keys.

Provided the bit-length of the CBID's is large enough [6], these identifiers have two very important properties: (1) they are *statistically unique*, because of the collision-resistance property of the cryptographic hash function used to generate them, and (2) they are *securely bound to a given*

node: the node can prove ownership of the CBID by signing packets with the corresponding private key. Any other node can verify the signature without relying on any centralized security service such as a PKI or Key Distribution Center.

These characteristics (1) make CBID's a very scalable naming system, well adapted to ad hoc environments, and (2) provide an autoconfigurable and solid foundation for nodes to engage in verifiable exchanges with each other.

3 Dynamic and Secure Node Authentication

The first application of the above is to protect basic exchanges between two peers or nodes in a network from malicious intermediate hosts. For example, in on-demand ad hoc routing protocols (e.g. [8, 7]), nodes discover each other by exchanging "route request" and "route reply" messages. We have recently shown how CBID can protect this basic exchange from impersonation attacks [2]. Similar work is underway for the JXTA open-source peer-to-peer protocol [5].

4 Dynamic and Secure Group Membership

Subsequently, CBID's are used to express authorization via authorization certificates, similar to how they are used to *Secure Group Management for IPv6* [3].

Authorization certificates have the following form:

- $Cert = (group, node, delegation, tag, validity)$

In the above, *group* is a group CBID for the entire impromptu network, or for a subset of it. Appropriately, the certificate is signed with the private key that corresponds to it.

Here, *node* is the CBID of the beneficiary of this authorization, that is, the node that is authorized by the group to join it or perform certain services on its behalf.

delegation is a boolean (in either SPKI [4] or KeyNote2 [1]) that specifies whether or not the group has allowed the *node* to further delegate the permission expressed in the next field.

tag is the authorization to be a member of the signing group, or to perform certain services as authorized by the group.

This is an example of how a single node *A* with CBID of A_CBID could start an ad hoc network (really a group within a perhaps already physically existing ad hoc network) by following these steps:

- *A* creates the group group public and private key pair: G_PK and G_SK.
- *A* creates the group identifier: $G\_CBID = hash(G\_PK)$.
- *A* as the *group controller* issues a certificate to allow itself into the group: $(G\_CBID, A\_CBID, true, "groupMembership", someDuration)$.
- *A* as the *group controller* admits another node (e.g. *B*) into the group by issuing the corresponding certificate: $(G\_CBID, B\_CBID, false, "groupMembership", someDuration)$.

Now, either *A* or *B* can prove to other nodes that they are legitimate members of the group by sending a message which includes their certificate, and that is signed with their private key A_SK or B_SK, respectively.

References

1. M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. *The KeyNote Trust-Management System Version 2*. IETF, RFC2704, September 1999.
2. C. Castelluccia and G. Montenegro. Protecting AODVng against impersonation attacks. *ACM Mobile Computing and Communications Review*, July 2002.
3. C. Castelluccia and G. Montenegro. Securing group management in IPv6,. *INRIA Technical Report*, August 2002.
4. C. a. Ellison. *SPKI Certificate Theory*. IETF, RFC 2693, September 1999.
5. Project JXTA (www.jxta.org).
6. G. Montenegro and C. Castelluccia. Statistically Unique and Cryptographically Verifiable (SUCV) identifiers and addresses. In *NDSS'02*, February 2002.
7. C. Perkins, E. Belding-Royer, and S. Das. *Ad Hoc On Demand Distance Vector (AODV) Routing for IP version 6*. IETF, draft-perkins-manet-aodv6-01.txt, work in progress, November 2001.
8. C. Perkins, E. Belding-Royer, and S. Das. *Ad Hoc On Demand Distance Vector (AODV) Routing*. IETF, draft-ietf-manet-aodv-11.txt, work in progress, June 2002.

Charging and Pricing in wireless networks*

Attila Weyland

Todays available charging and pricing schemes were designed for fixed networks and do not take into account the specific attributes of wireless networks. Among them the reliable and balanced management of the scarce network resources has been identified as an important issue.

All existing charging models can be categorized into one of the following three generic charging schemes. The *Flat-Rate* corresponds to a lease agreement for a service. There is no consideration of the actual service usage. With *Usage-based* charging the level of usage is measured in units of time or size (i.e. packets, bytes). When using *Transaction-based* charging, the price of a service is independent of the time or size the service consumed. Instead, the price reflects the value perceived by the customer.

Table 1 summarizes the advantages and disadvantages of the three generic charging schemes. While the implementation and the predictability of the usage are easy for a Flat-Rate, it potentially causes waste and cross-subsidization [1]. Usage-based charging corresponds best to the cost of providing the service, but the final costs of the service usage are less predictable. Transaction-based charging provides the best percieved value by a customer.

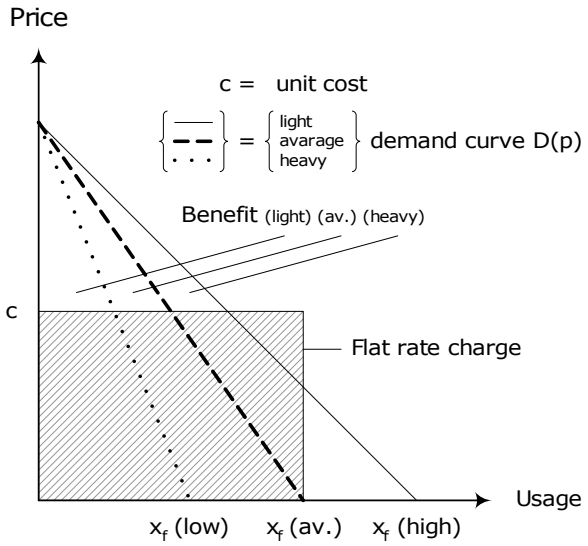
Table 1. Comparison of three generic charging schemes

	Flat-Rate	Usage based	Transaction based
Connection cost revenue	-	+	+/-
Perceived value by customer	+/-	+/-	+
Fairness	-	+	+
Predictability of service cost	+	-	+/-
Simplicity	+	+/-	+/-
Promotion of service usage	+	-	-

The cross-subsidization of the Flat-Rate scheme is explained in more detail in Figure 1. In this example, the Flat-Rate charge has been set to cover the cost of serving the average user, i.e. $c * x_f(av.) = c * D(0)$. This charge corresponds to the gray hatched

* An extended abstract of the presentation given during the RVS Summer School 2002 (August 26-30) held in Vira, Switzerland

Fig. 1. Different user types have different demand curves

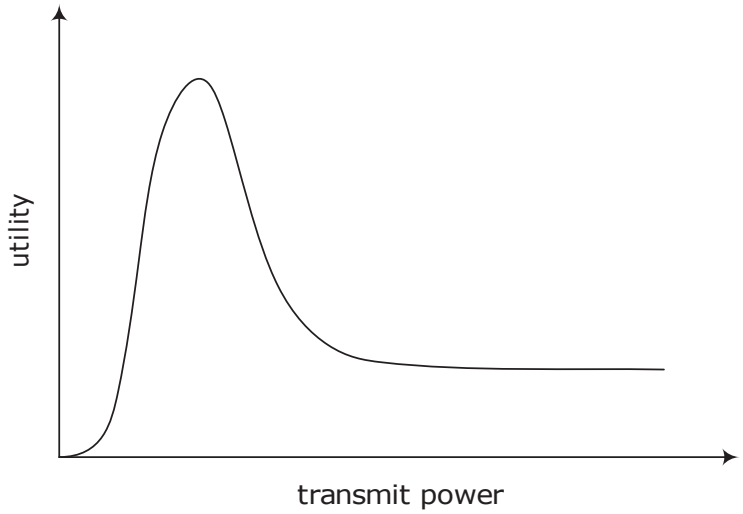


rectangle, which every subscriber has to pay. The benefit to the average user of consuming D(0) is the area under his demand curve, i.e. the middle triangular area. So the light users (the small triangle) subsidize the heavy users (the large triangle). The benefit for very light users can become even smaller than the Flat-Rate charge $c * D(0)$, so they will not subscribe. To retain these users, the ISP must set the charge below avarage cost (and incur an operating loss) or restrict usage.

One instrument to restrict usage is pricing. Especially discriminatory pricing is widely used for network services. Differenct prices for the same service can be set based on user, usage, time, congestion, quality or location.

Recent research investigates the application of economic models for resource control in wireless networks. [2] suggests a non-cooperative power control game with pricing. The utility is the number of information bits transmitted per unit of energy. The price of transmitting increases proportional to the required power, independet of the network load. Figure 2 shows the resulting utility function. In [4] a utility-based power control mechanisms is proposed. The price per power unit reflects the network congestion experienced by a user. In both papers the charge depends on the transmit power, which favoures nodes close to the base station as they don't need high transmit power. The authors of [3] perform the resource control based on congestion pricing. The transmission power is adjusted to achieve a certain signal quality (bit-energy-to-noise-density rate). The resource allocation takes place depending on the network load. The charge is independent of the transmit power and thereby also independent of the distance to the base station. These research activities show the growing need for good management concepts for the scarce resources (bandwidth, battery life).

Fig. 2. Utility in dependence of the transmit power



Additionally, the following requirements for a charging and pricing architecture for wireless networks have been identified. It should support heterogeneous networks by allowing different pricing models for different access technologies and the real-time adoption of the current pricing model. With scalable metering mechanisms the system is adoptable to a wider range of charging concepts. By considering the customer's location, her mobility behavior and the quality of the provided service, the service provider can charge fair prices.

Further investigations will be made on inter service provider charging schemes.

References

1. R. Edell and P. Varaiya. Providing Internet Access: What We Learn From INDEX. *IEEE Network*, 13(5), September 1999.
2. C. U. Saraydar, N. B. Mandayam, and D. J. Goodman. Efficient power control via pricing in wireless data networks. *IEEE Trans. Commun.*, 50(2):291–303, February 2001.
3. V. A. Siris, B. Briscoe, and D. Songhurst. Economic Models for Resource Control in Wireless Networks. In *Proc. of IEEE Int. Symp. Personal, Indoor, Mobile Radio Commun. (PIMRC)*, Lisbon, Portugal, Septmeber 2002.
4. M. Xiao, N. B. Shroff, and E. K. P. Chong. Utility-Based Power Control in Cellular Wireless Systems. In *Proc. of IEEE INFOCOM*, pages 412–421, Anchorage, Alaska, April 2001.

An Architecture for a Market Management of Peer-to-peer Services

Burkhard Stiller
University of Federal Armed Forces Munich, Germany and ETH Zrich, Switzerland

The Internet today combines mobile and high-bandwidth access as well as multiple possible services, participant roles, different classes of quality of service, and often distributed resource requirements. Emerging new peer-to-peer (P2P) systems offer potential solutions, but there are currently three areas of problems. (1) Different incentives for users to get them to provide their resources to P2P systems, (2) economically efficient resource allocation needs to be achieved in the absence of a central management system, and (3) scalable P2P communication mechanisms need to be in place covering the aspects of service provisioning and inter-operation.

The European Union MMAPPS project (Market Management of Peer-to-peer Services) will develop and trial an integrated solution to these problems that will allow P2P service creators and host, store, and transport providers to increase the benefit generated by the service. The talk will present the current status of this young project and proposes the architecture currently under discussion as well as their requirements.

eXperience Based Admission Control

Matthias Scheidegger

1 Introduction

In the field of QoS provisioning for IP networks the approach of Endpoint Admission Control recently attracted a lot of attention as a possible mechanism to scalably provide QoS guarantees to real-time flows. Unlike in other approaches to admission control there is no central entity controlling — or even knowing about — the resources used by the flows carried by the network. When a flow seeks admission to the network it first has to make a measurement and try to deduce from the results if the network can reasonably support the new flow. If not the flow fails to get admission and usually waits some random time until it tries again.

In the classical case the measurement is done by sending a stream of probing packets from the sender to the receiver. The bandwidth of this stream is the same as the peak bandwidth of the flow seeking admission to the network. After a couple of seconds (typically about 5), the receiver reports the packet loss rate back to the sender, which in turn stops the probing stream and evaluates the situation. By comparing thee packet loss ratio to a threshold one can limit the packet loss ratio to some low level λ with some high probability p. Typical values for these variables would be $\lambda = 0.01$ and $p = 0.95$. This results in a real-time service comparable to other, more centralized schemes.

There are many variants of this basic approach. [2] gives a good overview, while [1], [5] and [6], amongst others, describe single variants. Note that passive measurement based admission control as seen in [3] was not considered here.

However, there are a couple of problems with the EAC approach. Section 2 explains what they are. Sections 3 and 4 introduce an approach to solve these problems.

2 Problems and Properties of EAC

As mentioned above there are some problems with the Endpoint Admission Control approach. The most important of them is definitely the long setup delay common to all active approaches. Five seconds setup delay on the average — as required by the basic approach — are clearly unacceptable for most real-time applications. Even the more sophisticated schemes need around two seconds to set up a flow; still too much for applications like toll quality IP telephony.

Another problem with EAC is the rather high bandwidth waste caused by the probing packets. While this is a minor problem compared to the benefits of using EAC, reducing this overhead is still a worthy goal.

Before introducing the approach of eXperience Based Admission Control (XBAC) we will investigate a number of observations of fundamental EAC properties and the current Internet infrastructure who may help solving the problems mentioned above, and who constitute the fundament of the approach.

Many areas of the Internet topology look similar to the situation in Figure 1: A local or metropolitan area network is connected to the global Internet by means of a link with significantly smaller bandwidth than the links inside the local network. This situation frequently occurs in networks of medium- to large-sized organizations like universities, many enterprises or governmental institutions. The superior bandwidth inside the organization's network, as compared to the outside link, makes the nodes inside of such a network practically equivalent, with respect to EAC measurements to nodes in other parts of the Internet. That is, if any probing packets get dropped during a measurement period, it will very probably happen on the far side of the small bandwidth link. A single node inside this network could therefore perform the measurements for all others. Below, we will call subnetworks with these properties *equivalence zones*, or EZs for short.

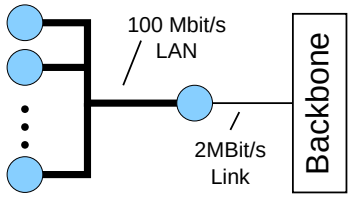


Fig. 1. Observed Topology

3 The XBAC concept

Endpoint admission controlled traffic from one equivalence zone often goes to other equivalence zones. When we look at the whole of the flows going from our equivalence zone A to some equivalence zone B, we observe something similar to Figure 2, i.e. a series of overlapping flows, each measuring then sending.

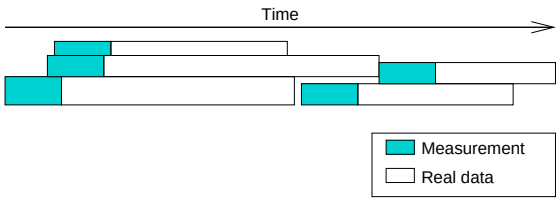


Fig. 2. Simplified view of flows between EZs

One inefficiency is rather obvious when we look at this Figure: Flows that enter the network shortly after another flow left still measure the network in any case, even if the bandwidth just freed is significantly higher than their own peak bandwidth. The fourth flow in Figure 2 is an example for this. The measurement phase of flow setup could probably be left out because of the network behavior experienced by earlier flows. Instead of going through a measurement procedure every time a flow enters the network we could thus draw on the experience gathered by earlier flows. This could lead to significantly shorter setup delays on the average and less bandwidth consumption for measurements on the whole. We call this approach *eXperience Based Admission Control* (XBAC).

To formalize the approach, assume a single node per equivalence zone to be the so called *Measuring Endpoint* (ME). All applications using the endpoint admission controlled service contact the ME to request admission and to report their experience once they leave the network. After a certain amount of time the ME has gathered enough experience to predict if a flow can enter the network without measurement or if a measurement to the destination is necessary. This prediction function is the most difficult element of the approach to find. For a destination EZ D the ME must provide an admission control function $\alpha_D(t, b, p)$. α shall return 1 if the packets of a flow with destination D entering the network at time t with peak bandwidth b will successfully arrive with a probability greater or equal than p, or 0 otherwise. Whether the admission control decision was based on stored experience or on active measurement remains transparent.

4 Identifying Equivalence Zones

In order to be able to implement the approach from Section 3 — besides specifying the α function — the solution to a further problem is required.

Since experience is stored separately for every destination EZ, we need a way to determine the "far end" equivalence zone of a flow. A simple approach would be to define a protocol, which allows for endpoints to return an equivalence zone ID (e.g. the IP address of its ME) when queried. However, such a service may prove too difficult to deploy on all endpoint devices. Alternatively, topology discovery may be used. Traditional tools like `traceroute` may help here, but there are alternatives: Calculating the correlations of measurements and flow experiences for different destinations can help detecting EZs. Moreover, differences in measurements to different destinations can be useful in topology detection. The latter approach is related to network tomography and is described in [4].

References

1. G. Bianchi, A. Capone, and C. Petrioli. Throughput analysis of end-to-end measurement based admission control in ip. In *IEEE INFOCOM, Tel Aviv*, 2000.
2. L. Breslau, E. Knightly, S. Shenker, I. Stoica, and H. Zhang. Endpoint admission control: Architectural issues and performance. In *SIGCOMM 2000, Stockholm*, 2000.

3. C. Cetinkaya and E. Knightly. Egress admission control. In *IEEE INFOCOM, Tel Aviv*, 2000.
4. Mark Coates, Alfred Hero, Robert Nowak, and Bin Yu. Internet tomography, 2002.
5. V. Elek, G. Karlsson, and R. Rönngren. Admission control based on end-to-end measurements. In *IEEE INFOCOM, Tel Aviv*, 2000.
6. F. Kelly, P. Key, and S. Zachary. Distributed admission control. In *IEEE JSAC*, 2000.

Signaling and Monitoring Quality of Service in Differentiated Services Networks

A Token Based Reservation Scheme for Differentiated Services Networks

Florian Baumgartner

Department of Computer Sciences, Purdue University
baumgart@cs.purdue.edu

1 Introduction

The core routers of a Differentiated Services domain only forward packets, while the ingress and egress routers are responsible for the policing and shaping of flows. While this concept guarantees scalability within the backbone it also complicates the network provisioning and the detection of possible bottlenecks and service violations. Following some signaling process to negotiate Service Level Agreements (SLAs) between a customer (e.g. an upstream ISP), the ingress routers will set up appropriate policing mechanisms.
Within the network the core routers are not concerned with the negotiation of SLAs, and a core router has no information about the current reservation situation within the network. Of course it can easily measure the current load, but not the resources, which are allocated at the border routers. This lack of information is a crucial disadvantage if it comes to network planing, the avoidance of service degradation or the detection of intrusions like the theft of QoS.
An approach to cope with that problem is proposed by Habib [1]. By injection probe packets at the ingress routers along with the usual traffic, the current Quality of Service parameters along a certain path between a pair of ingress/egress routers can be measured. By mapping the network topology to a tree the status of each dedicated link within the domain can be calculated (see also Duffield [2])
Since the signaling process to negotiate an SLA is an open issue also, we propose a scheme which provides:

- an automatic negotiation of Service Level Agreements for a specific service class on a per flow level
- admission control using Differentiated Service traffic conditioning mechanisms.
- a distribution of information regarding the actual reservation load on specific routers within the network.

The basic mechanism of the proposed scheme is simple. To provide a signaling of SLAs on a per flow level, Reservation Tokens (RTs) are sent parallel to the data packets to

signal a resource requests to the routers. In contrast to RSVP [3] this procession in the core routers is done state less and therefore scalable.

Figure 1 shows the flow of RTs from different end systems or access networks though a bottleneck link. Since RTs are simply forwarded with the flows, they are allocating resources for, an aggregation of flows automatically leads to an aggregation of tokens. In contrast to SRP [4], which provided no solution for the policing of flows, RTs have to be policed similar to incoming DS traffic. Since the tokens contain information about the bandwidth and have a limited time to live they also can be used at the ingress routers to set up the policing mechanisms, and to provide information about the actual allocated resources at each point within the network.

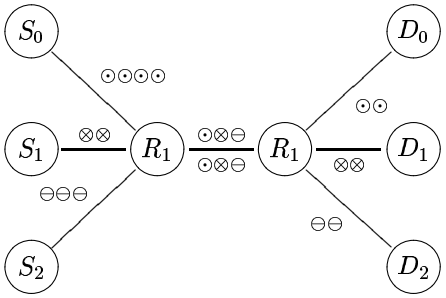


Fig. 1. Flow of Reservation Tokens through networks. The ingress router applies policing rules ($\odot$ and $\ominus$ tokens).

2 A simple Reservation Protocol

Based on the idea to send tokens along flows to signal a resource request, a simple protocol has been developed. Especially important is to minimize the required modifications to the involved network devices.

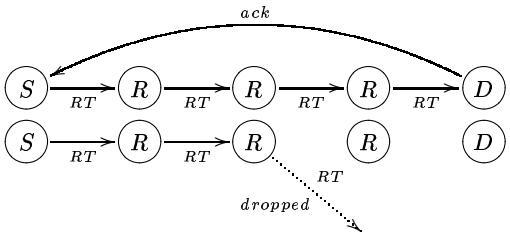


Fig. 2. Simple Reservation Token Protocol. If the reservation request can be granted, the first token ($seq = 0$) is acknowledged with an echo reply. If the request would exceed the available resources, the token is dropped and no echo reply is received by the sender of the token.

The protocol is sender driven and requires an adaption of the traffic sources only. Since, as will be explained in the next paragraphs, the protocol is rather simple, it can be added to existing services (video streaming, web and proxy servers) or can be provided by some kind of daemon, which injects RTs for the applications. Figure 2 shows the message exchange during the establishment of a reservation. If a reservation has to be set up, the traffic source sends an RT along the destination. If the destination receives the token, it simply acknowledges the reception to the source. Once the source has received that acknowledgement, further RTs can be send and the data transmission can start.

If a router on the path of the RT will not be able to handle the request, the RT is simply discarded. Therefore the sender has, after sending the initial RT, to wait for a short time, whether he receives an echo reply or the RT was discarded. Since the purpose of this scheme is to keep the network congestion free, this timeout can be very short, since no queueing delays can occur. To eliminate the need for an adaption of the end system, we propose to modify ICMP echo reply (a.k.a. ping) packets to carry the RT information. Since every IP host has to provide the echo reply service, the system will work with any type of receiver.

Content of Reservation Tokens

To minimize the overload, RTs are very small. Table 1 shows the minimal set of required fields and the proposed number of octets per field. Besides the requested bandwidth in bytes per second, a sequence number starting at 0, being incremented which each RT sent, and the time to live, an RT might also contain additional information like authorization codes. In the next paragraphs mechanisms the token processing mechanisms for the core and the ingress routers are presented.

Description	Octets
Bandwidth	4
Sequence No.	2
Time to Live	1

Table 1. Data Fields within a Reservation Token.

3 Token Processing in Border and Core Routers

A component, each router has to maintain, is a time table as shown in Table 2. This time table is used to aggregate the bandwidth values signaled by the RTs for a specific service. The time table contains a set of n time slots, each covering a time period of the length g. Even if the actual value for g is up to the single router the overall forecast period ng must at least cover the maximum allowed value of an RT's time to live field, resulting in a table of 255 columns for a maximum allowed time to live of 255 seconds and a time period length of one second.

Service	0	1	2	...	n
S_0					
S_1					

Table 2. Router Time Table.

To check whether a resource request, represented by an incoming RT with the service class s can be fulfilled, the core has to check whether any field of the time table $T_{i,s}$ with $ig \leq ttl_{RT}$ for that particular service S_s exceeds a certain threshold th. Since no reservations in advance are allowed, and therefore $T_{i,s} \leq T_{i+1,s}$ is true, it is sufficient to check the $T_{0,s}$ field of the according service s. If the resource can be granted, the bandwidth signaled within the RT, is added to each field $T_{i,s}$ with $ig \leq ttl_{RT}$ and the token is forwarded. Since each router has only maintain one table with only $\approx$ 1 kb of memory is required per service class, the memory consumption is negligible.

Applying this simple scheme it is possible for a core router to keep track of allocated resources, and even estimate, within certain limits, future demands. Early simulations revealed two main problems of that a simple scheme:

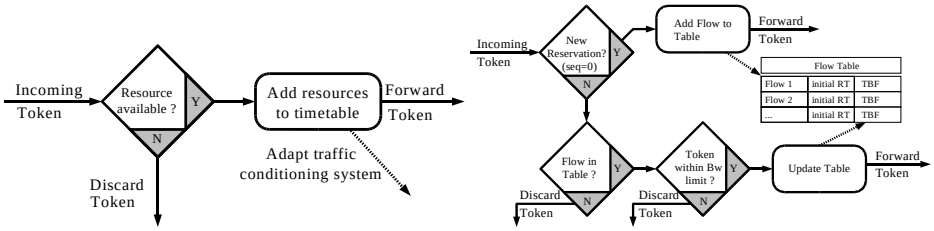


Fig. 3. Processing of Reservation Tokens in Core and Ingress Routers.

- If resources get scarce, new reservations should be rejected while existing reservations are kept alive. Instead of rejecting RTs equally, tokens of existing reservations should be accepted with a higher priority.
- If a downstream router, rejects an RT and discards it, resources in the upstream routers are already allocated for ttl_{RT} seconds. This causes a mismatch between the reservation status represented by the timetable and the real situation.
- Misbehaving end systems, may spoil the system.

To keep existing reservations alive, while new reservations are already rejected, two thresholds are used instead of a single one. A threshold value $th_{s=0}$ is used for initial tokens with a sequence number of 0 and a larger one $th_{s>0}$ for all other tokens. It is important for $th_{s>0} - th_{s=0}$ to be larger than the maximum amount of bandwidth which can be allocated for a single flow.
The second problem of unnecessary reserved resources in upstream routers is reduced by a small change to the protocol. The ttl of the first RT to be sent, is set very small. If accepted, it only has to reserve bandwidth as long as it takes to receive the echo reply packet from the receiver. The following RTs then can cover longer time periods. On the other hand, if rejected, this first token only causes a wrong reservation for a very short time.

4 Evaluation

The protocol presented in the previous section was evaluated by simulations using a flow based model. The rejection rates and the signaled and allocated bandwidths within the network were evaluated. Four access networks (A_i) are connected to a set of nodes acting as ingress routers ($N_0, ..., N_3$). Further nodes in the tree like topology cause further aggregation with a complete aggregation of all flows in N_0. Due to the aggregation not all reservations can be granted. During the simulation each for each flow a constant bandwidth of 64 kbps is tried to be allocated. The start time, the duration of the sender and the frequency the tokens are sent with are chosen randomly for each sender.
Figure 4 shows the number of rejected tokens during the simulation period. Since the senders start randomly during the simulation and are active for random time, there is only moderate load at the beginning of the simulation. Once the load exceeds the available bandwidth ($th_{s=0}$), tokens for new reservations are dropped, while tokens for al-

Fig. 4. The overall token rate, and the rates of rejected tokens during the simulation.

ready existing reservations are still accepted. Therefore, even if no new reservation is accepted, all already existing flows get the requested bandwidth.

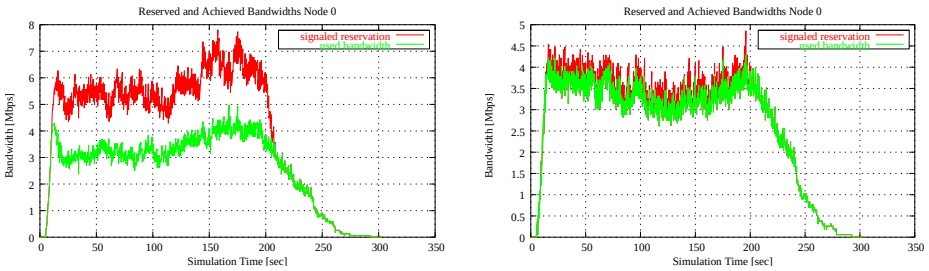


Fig. 5. Allocated and consumed bandwidth with (right hand graph) and without (left hand graph) short first interval mechanisms.

Since there is no exact signaling of a rejected reservation, there is a certain difference between the resources a node has to reserve and the resources actually consumed by the clients. The left hand graph in Figure 5 shows this difference. A router, which accepts a reservation, has to allocate appropriate resources for the time signaled by the token, even if the reservation is rejected further downstream. Of course the tree like scenario with N_0 as bottleneck is a worst case scenario for that case. This waste of bandwidth, can be reduced by minimizing the time to live of the first token sent. This way, resources are allocated only for a short time. The right hand graph on Figure 5 shows the same scenario, but with the "short first interval" algorithm. The mechanisms reduces the overhead significantly. Of course the overhead depends also on how many routers have accepted the token until it is rejected. The more hops are involved and the more downstream the token is rejected, the worse is the overhead.

5 Conclusion

Even if the allocation of resources using traffic aggregates using DS works well, there is a need for concepts to signal reservation requests and to provide information about the

reservation status to core router also. The scheme using RTs to signal resource requests provides a scalable and light-weight mechanism to negotiate SLAs, to distribute information to the core routers and to enable mechanisms to control service violations. The protocol does not require a modification of the receivers and also supports core routers, which perform no RT processing.

References

1. A.Habib, M.Khan, B.Bhargava: *Overlay-based Distributed Network Monitoring*, June 2002.
2. N.G.Duffield, F.Lo Presti, V.Paxson, D.Towsley: *Inferring Link Loss Using Striped Unicast Probes*, 1998.
3. B.Braden, L.Zhang, S.Berson, S.Herzog and S.Jamin: *Resource Reservation Protocol (RSVP) - Version 1 Functional Specification*, RFC 2205, September 1997.
4. W.Almesberger, T.Ferrari and J.Le Boudec: *SRP: a Scalable Resource Reservation Protocol for the Internet*, http://lrcwww.epfl.ch/srp, March 1998.

Self-Organizing Networks

Torsten Braun

Self-organizing networks organize themselves without any centralized control, i.e. in a fully distributed manner. Such concepts can be found in both peer-to-peer (P2P) networks [1], [2], [3] and mobile ad-hoc networks [4].

Peer-to-Peer (P2P) Networks

P2P networks have usually the task to map application identifiers (e.g., file identifiers) to end system identifiers (e.g., IP addresses). P2P networks can be classified into networks with centralized index (e.g., Napster) and distributed networks, in which the index information is distributed among the various peers. The distributed P2P networks can further be divided into unstructured and structured P2P networks. Unstructured P2P networks are characterized by flat network structures, where peers can connect arbitrarily to each other and where index information is rather randomly stored on peer nodes. Examples for those are Gnutella and Freenet. However, in particular the flooding approach (for searches and neighbour detection) of Gnutella may result in severe performance problems. In contrast, structured approaches establish network structures among peers such as rings (e.g., Chord) or hypercubes (e.g., CAN). Keys are mapped to a certain location within such a structure. This can be used to redirect search queries to a location that can be derived deterministically from the key. An additional approach is to introduce a secondary overlay structure such as SuperNodes that have special functions within a peer network. In particular, they may store which nodes are belonging to the subset of peers they are responsible for.

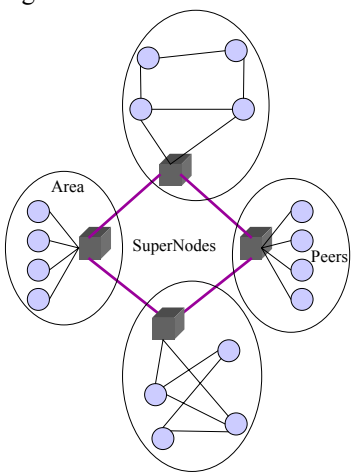
Mobile Ad-Hoc Networks

A similar problem as in P2P networks must be solved in mobile ad-hoc networks, in particular in approaches based on geographic positions [5]: A logical identifier (e.g., an host name) must be mapped to some network specific address such as a geographic position. This is supported by so-called location services. Examples are DREAM, quorum-based location services (QLS), GRID, and the virtual home region (VHR) approach. However, these approaches are not appropriate for P2P networks due to several different reasons. DREAM does not scale well, QLS might require wide are connections, GRID performs a kind of flooding, and VHR does not cover the complete hash space. On the other hand, P2P mechanisms are not appropriate for mobile ad-hoc networks, since structured approaches might suffer from peer mobility and unstructured approaches might easily get scalability problems.

Peer-to-Peer over Mobile Ad-Hoc Networks

One could imagine to run P2P networks over mobile ad-hoc networks.For example, audio files or software updates existing at a mobile device could be made available to mo-

bile peer devices. Integrating P2P and mobile ad-hoc network concepts requires to map keys to IP addresses in a first step and IP addresses to geographic location information in a second step. One could think about mapping keys to location information directly, but due to frequent end system mobility, this approach might probably not be feasible. Therefore, we propose to keep the two-stage mapping supported by an architecture that makes use of both P2P and mobile ad-hoc network mechanisms. We propose SuperNodes for each geographic area that are responsible for certain intervals of the total key space (cf. Figure below). Each geographic area has now similar tasks as a single CAN node, i.e. it should be able to resolve IP addresses from a given key out of a certain key interval. If a peer has to resolve a key, a request is forwarded into the geographic area derived deterministically from the key. Within that area an (un)structured P2P approach can be used to resolve the key and return an IP address of the peer with the desired information associated with the key. If a peer leaves its area, it has to deliver the mapping data it stores to other peers remaining in the area. A peer entering an area should take over some mapping information (key, IP address) from other peers in that area. Due to reliability, keys should be stored on more than a single peer within an area. Optionally, the SuperNodes (probably several redundant ones exist for an area) may act as location servers. Peers entering an area register with one or more SuperNodes in that area. SuperNodes set up another P2P network and store mapping information (IP address, location). OceanStore with SuperNodes in the home area as root nodes might be an appropriate approach for this P2P network. Another node searching a location of a peer (identified by an IP address) forwards a request towards the SuperNode network. The SuperNodes then resolve the given IP address onto the area of the server peer.



[1] Andy Oram (ed.): Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology, O'Reilly & Associates, 2001

[2] Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), 7-8 March 2002, Cambridge, USA, www.cs.rice.edu/Conferences/IPTPS02/

[3] Theme Feature "Peer to Peer Networking", Internet Computing, January/February 2002

[4] C. Perkins: Ad-Hoc Networking, Addison-Wesley, 2001

[5] M. Mauve, J. Widmer, H. Hartenstein: A Survey on Position-Based Routing in Mobile Ad Hoc Networks, IEEE Network, November 2001

Active Components for QoS and Multicast Setup

Roland Balmer

Introduction

Over the years, the Internet has become so popular that its size has doubled several times, and to some extent it has been successful in satisfying the needs of most of its users. These users deal typically with non-demanding services such as file downloads and emails exchanges, where usually the amount of data is small and the involved transfer duration is short. To such services, the Internet might not impose any bandwidth constraint as it is usually faster than the conventional access networks (e.g., dial up, ADSL) used by most users.

On the other hand, in recent years emerging demanding applications have been posing very tough requirements to this global network. An example of such a service is broadcasting of video streams, in which both the amount of data and the transmission duration are usually very expressive. Typical values for this traffic are bandwidth of up to 15 Mbps for about 2 hours of transmission. Considering that this video traffic needs to be transmitted in real-time, then the data must arrive in time in order to guarantee the required quality of service.

Therefore, to make sure that this sort of traffic will receive privileged treatment, a multicast reservation scheme must be in place. In this way, possible problems with intermediate low bandwidth links are avoided. In this report, we outline the main requirements involved in this reservation scheme and present our early idea, through an example, for improvements in such a scenario.

Requirements

We classify the requirements as Video Service Provider (or simply Service Provider) and Internet Service Provider (ISP) requirements. The Service Provider has no special requirements. It only wants that the video stream be transported over the Internet without any reduction of the quality. In other words, the transported data must arrive in time (low delay and low jitter) and no loss (or very little) are allowed. So, since the Service Provider cannot change anything in terms of setup, on the Internet, it requests the local ISP to set up the protection of its data.

Regarding the ISP, its requirements mean to use the resources in an effective way in order to maximize its profit. Thus, it has to keep control over the entire network and only allow requests from trusted neighbors, via a service level agreement (SLA). To do this, the ISP needs at the edge of the network an instance that is capable of checking and handling the incoming requests, which can be done by the so called Policy Enforcement Points (PEP). Depending on the power of the PEP and on the amount of traffic it has to handle, it might need to have a superior authority to take the not simple decisions. This, so called Policy Server authority (policy decision point) must keep an overview of the network of the ISP. The extent of this overview should be defined by the mechanism used to set up the core routers of the network. The control of the core routers can be handled in two different ways:

First, the routers are controlled from a central instance, called Bandwidth Broker (BB) [GSB02]. In this case the BB controls all routers directly and manages the information of every router and link in the network.

Second, the decisions are taken directly in the core routers. The problem here is that the core routers are typically optimized to forward and reroute the transported packets. Because of that, in most cases, the operating system of the core routers are not designed to handle

any possible management requests. To be able to support requests, the core routers need a mechanism to load the needed modules that will handle the new task. Provided that the core routers support requests, then the code can be transported with the setup request over the network [BCZ98]. Because the setup is done on each router by itself, there is no central instance which knows the status of the whole internal network. With the help of the PEP, the status of the border routers can be determined. These information are enough to check if the SLAs are being respected.

The advantage of the second solution is that the intelligence is only in the network and the central instance is only needed to supervise the network. Thus, we outline below an example of this second approach tailored to our needs.

An example of how to manage the ”requests”

To explain the steps of the example, Fig. 1 is used. In this example the resources are reserved for a service that uses multicast, such as native or explicit multicast [BFI+01]. The multicast-tree (solid and thick line) is already established. There are two links connecting authorized clients to the Multicast-tree. One client is connected directly to the ISP, and one or more clients are connected to another network. The multicast-tree has also a member which is not allowed to reserve/receive any resource. The server identifies the clients by the information inside the request messages.

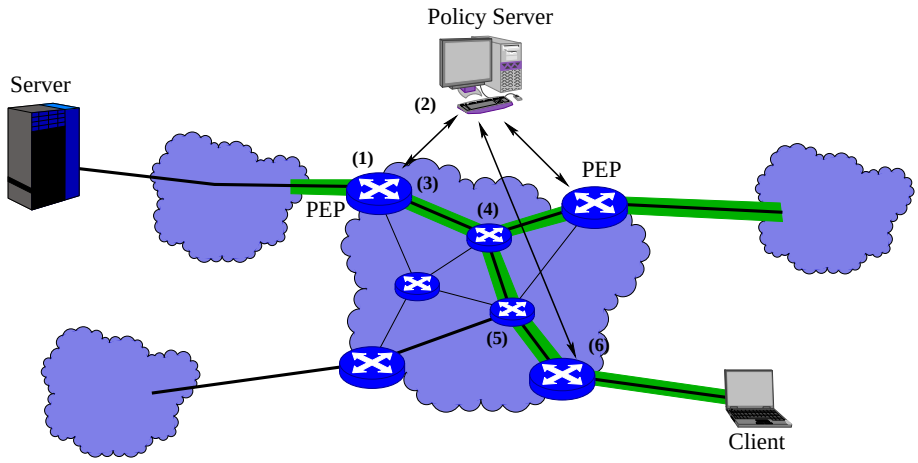


Fig. 1. Szenario used for the example.

(1) The PEP receives the request from the last ISP. The PEP verifies if it is a valid request and of what kind it is. The request asks for new resources and this can only be granted by the Policy Server. (2) The request is forwarded to the Policy Server. The Policy Server checks if the SLA is not violated and if the resources are available on the forwarding links. If it is all okay, then a confirmation is sent back to the PEP, including the needed program-modules and parameters to set up the routers. (3) The PEP adds these active components to the request and further processes it. As the PEP is not a split point the resources on the outgoing link are reserved and the packet is sent to the next router. (4) The next router checks for the validity of the packet, and processes it if it is a valid one. Since the router is a split point, the two outgoing links receive the resource reservation. So, the packet is duplicated and sent to the two next routers. (5) On the next router the packet is checked in the same way, and as only on one of its links a client exists, the resources are only reserved for the that link and the request is forwarded. (6) At the outgoing PEP the request is checked again. The resources are reserved and the request is forwarded to the next ISP

or to the client. Additionally, the active components could be removed before forwarding. If requested by the Policy Server, the PEP can inform it about the successful setup of the request.

References

[BCZ98] Samrat Bhattacharjee, Kenneth L. Calvert, and Ellen W. Zegura. Active Networking and the End-to-End Argument. Georgia Institut of Technologie, Atlanta, 1998.

[BFI+01] R. Boivie, N. Feldman, Y. Imai, W. Livens, D. Ooms, and O. Paridaens. Explicit Multicast (Xcast) Basic Specification. Internet Draft: draft-ooms-xcast-basic-spec-02, October 2001.

[GSB02] Günther Stattenberger and Torsten Braun. Design and Performance of a Bandwodth Broker. submitted for publication, July 2002.

Protocol Metabolism

Christian Tschudin
Uppsala University, Sweden

In most cases, communication protocols are considered as communicating finite state machines. Static protocol implementations provide a communciation service by exchanging passive messages. Mobile programs provide a new way to implement communication protocols and allow a more flexible and dynamic execution. However, only a few results are available from theoretical analysis of such active protocols. The talk presented a new protocol model based on the exchange of fraglets. Fraglets are small fragments that include both code and data. This allows that fraglets can operate on fraglets and therefore the difference between passive data and active code disappears. Based on concrete examples, it has been shown how protocols can be implemented.

Hierarchical Bandwidth Brokers to provide QoS to Mobile Users

Günther Stattenberger

1 Introduction

In the last few years an astonishing propagation of small, wearable electronic devices like laptops, personal digital assistants (PDAs), and mobile phones could be observed. While in the beginning of this trend, the former could not conveniently used for communication, the latter had not enough computational power to solve difficult tasks. However, recently a affiliation of those initially separated branches is noticeable: mobile phones are equipped with larger displays and powerful CPUs, whereas PDAs and laptops contain a broad variety of communication equipment, such as Wireless LAN (IEEE 802.11), and Bluetooth (IEEE 802.15). Due to this fact new applications that were beforehand restricted to fixed terminals are now getting available to mobile users. The transfer of applications from fixed to wireless access methods implies nevertheless requires additional thought: The Quality of Service (QoS) wireless networks offer varies from the QoS users are used to in wired networks:

- the bandwidth of wireless networks is much smaller than the bandwidth of wired networks
- the delay in a wireless network is higher,
- wireless media usually suffer from an bit error rate several orders of magnitude higher than wired media (10^{-5} compared to 10^{-13}), they are subject to interference and shadowing effects
- handovers between different base stations in wireless access networks cause a break in the transmission
- a wireless network uses a shared medium, i.e. the QoS delivery depends on the presence of other nodes in the network

Therefore, Quality of Service (QoS) issues have a much higher impact on applications running on a mobile device. Several modern applications, such as multimedia streaming, audio/video conferencing, or distance education, rely on continuous QoS provisioning of the underlying network. Adaptive applications may be able to decrease the bandwidth requirements of themselves if they detect QoS degradation but this always results in a worse quality for the user and is therefore not desirable. In this thesis I want to discuss the enhancements that are to apply to the current internet architecture in order to provide continuous Quality of Service to mobile users.

Nowadays a mobile user meets several problems if he has negotiated a certain QoS level in its home network and is now willing to roam to a new (foreign) access network. We want to present an architecture that is able to support mobile users to sustain a QoS level even during their movements.

2 Bandwidth Broker

2.1 Architecture

In order to support mobile users and their QoS requirements we propose a novel Bandwidth Broker architecture plus a QoS signaling protocol [6, 5, 4] that provides enough functionality to solve the problems mentioned above and has a flexible programming interface to add arbitrary extensions in the future. Our BB architecture can be split into two different

parts: a management part and a configuration part. Several other architectures use this separation, too [7, 2]. The novelty in our approach lies in the interconnection of those two layers: We propose to use an object-oriented virtual representation of the underlying network. The real-world network is thereby mapped to a collection of `Router` objects. Those `Router` objects present a generic interface for configuration to the management layer. Via derivation (polymorphism) we can ensure, that each instance can correctly configure the real router it represents. Therefore we can greatly simplify the management layer since we can configure a virtually homogeneous network.

The configuration layer consists of hardware - dependent configuration daemons running one at each router. The communication between the configuration layer and the virtual network is unspecified. Virtually any means of configuration protocol can be used (e.g. SNMP, CLI). In our implementation we use a TCP/IP socket based protocol offering a command line interface to the Linux `traffic control` configuration API. The variety of this communication is again hidden from the management layer by the polymorphic virtual network.

The management layer of the BB has to fulfill several requirements in order to be able to configure the network:

- it has to know the topology
- it has to keep track of the resources that are currently allocated
- it has to perform call admission control
- it has to communicate with adjacent bandwidth brokers
- it has to communicate with users to negotiate resource reservations

Several building blocks are thus needed to form the management layer of the BB:

- topology database
- reservation and utilization tables
- policy database
- flow table
- user and broker communication interfaces

The topology database naturally forms the interface between the management layer and the virtual network. It is automatically built during the startup phase and contains all IP routing information about the network. This information is held in routing tables that are part of the `Router` objects of the virtual network.

Both communication interfaces — broker-broker and user-broker are built on top of a TCP/IP server-client model that offers a functionality similar the the SUN RPC model. The server offers a certain set of functions (e.g. `add_flow`, `del_flow`) the client can call; the client can pass parameters to the functions and receives a result. Broadcast functionality is included, too.

The bandwidth management part consists of the reservation and utilization tables, together with the policy database. Those components form the decision base about admission or rejection of a flow: a flow can only be admitted if the resulting bandwidth utilization on any link is lower than the limit configured in the policy database. The policy DB contains such limits for each ingress link and each connected foreign subnet according to the contracts negotiated with neighboring BBs.

2.2 Implementation and Performance

The architecture presented here has been implemented in C++ and evaluated to show which size of network can be managed [3]. Another topic was the call admission rate (i.e. the time a user has to wait until it receives a response from the BB) that can be achieved with this architecture. Our performance evaluation showed a very good call admission rate (about 4 ms per flow) even at large network sizes (800 nodes). As far as we know, no other BB implementation has been tested with such a large network.

2.3 Hierarchical Bandwidth Brokers

Even if the performance evaluation showed a good performance there may be situations where there is a need for a higher CAR. This might be the case in a environment consisting of lots of small wireless cells and thus lots of handovers (e.g. crowded areas like airports or populated highways). In such an environment, distributing the load between several bandwidth brokers can provide better performance. A hierarchy of bandwidth brokers can be applied, where the leaf brokers manage a small part of the network independently, allocate large chunks of bandwidth at the root broker and distribute those resources on their own responsibility. The distribution of the leaf brokers is crucial and needs further investigation individually for each environment: the location of hot spots and the detection of common paths is important to form subsets of access points that are managed by one bandwidth broker in order to minimize the inter-broker handoffs.

3 Quality of Service for Mobile Users

The support of handovers in a wireless environment is the critical topic of QoS provisioning for mobile users, since handovers result in a variety of inevitable consequences affecting QoS:

- connectivity disruption
- packet loss due to misrouted packets
- registration at the home agent ($\Rightarrow$ delay)
- new reservation setup ($\Rightarrow$ delay)
- no bandwidth available at the new cell?

Several improvements are currently discussed in the IETF namely fast and smooth handover techniques [1], but those are mainly focusing on the second and third item. We want to propose a QoS-aware handover procedure that helps to solve the 4th and 5th item. Only very limited work is done on this field so far.

The main idea of our *user-initiated* handover is to delegate the decision about handover time and new access network to user-configurable daemon. This daemon monitors the signal quality, alerts affected applications and scans for available new wireless cells. Since this is made while staying connected via the old cell the daemon can additionally make reservations in the new wireless cell and explicitely perform the handover at a specified time. Our bandwidth broker supports the daemon by providing additional information about the base stations, such as available bandwidth and location.

4 Conclusion

Our BB implementation supports QoS in a mobile and dynamic environment. Mobile users can achieve quasi-continuaous QoS with minimal interruption by negotiating resource reservations with the new access network previous to the handover. In a heavy-loaded environments with lots of handovers a hierarchical structure of bandwidth brokers provides a good performance.

References

1. G. Dommety (Ed.), A. Yegin, C. Perkins, G. Tsirtis, K. El-Malki, and M. Khalil. Fast Handovers for Mobile IPv6. Internet Draft, March 2002. work in progress.
2. G. Politis, P. Sampatakos, and I. Venieris. Design of a Multi-Layer Bandwidth Broker Architecture. In Sathya Rao and Kaare Ingar Sletta, editors, *Next Generation Networks — Networks and Services for the Information Society*, volume 1938 of *Lecture Notes in Computer Science*, pages 316 – 325. October 2000.

3. G. Stattenberger and T. Braun. Performance of a Bandwidth Broker for DiffServ Networks. In *KiVS 2003*. submitted for publication.
4. G. Stattenberger and T. Braun. Providing Differentiated Services to Mobile IP Users. In *Proceedings of the 26th Annual IEEE Conference on Local Computer Networks*, November 2001.
5. G. Stattenberger and T. Braun. QoS Provisioning for Mobile IP Users. In H. Afifi and D. Zeghlache, editors, *Conference on Applications and Services in Wireless Networks, ASW 2001*, Paris, July 2001.
6. G. Stattenberger, T. Braun, and M. Brunner. A Platform - Independent API for Quality of Service Management. In *Proceedings of the IEEE Workshop on High Performance Switching and Routing*, May 2001.
7. Zhi-Li Zhang, Z. Duan, and Y. T. Hou. On Scalable Network Resource Management Using Bandwidth Brokers. In *Proceedings of the Network Operations and Management Symposium*, April 2002.

Mobile Applications in Ad-Hoc Networks – Examples and Improvements

Extended Abstract of a Talk given at the
RVS Summer School 2002
Vira, Switzerland, August 2002

Stefan Fischer
TU Braunschweig
17 September 2002

1 An Example for an Ad-Hoc Network Environment: DeepMap

Mobile Ad-Hoc networks are networks that are built spontaneously from mobile devices communicating via wireless networks and currently being available in a certain geographical region. The idea is that some of the devices act as relay stations such that two stations, which are not in direct range, can communicate with each other over this new multi-hop connection.

A typical environment where setting up ad-hoc networks is possible is provided by the DeepMap project. The goal of DeepMap is more or less implementing a mobile virtual tourist guide for the city of Heidelberg. Tourists are equipped with a handheld device which they can use to access information systems being distributed all over the city. Therefore, these devices are equipped with mobile network card, in our case with Wireless LAN. For many applications, the devices can connect directly to each other and can make use of the proximity of other devices, for relaying purposes. An important question, however, is, when it is worth trying to set up an ad-hoc network. Ad-hoc networks are not very stable and it might happy relatively often that an established link breaks shortly thereafter. Therefore, we are interested in investigating the connectivity in such networks.

2 Checking Connectivity in Ad-Hoc Networks: the ANSim Simulator

For that purpose, we developed our own ad-hoc network simulator which we called ANSim. ANSim uses a simple model for the transmission characteristics of mobile nodes, basically assuming huge free spaces where the nodes are located. The user can freely choose to modify a relatively huge set of parameters, for instance the size of the overall area, the transmission range, or the number of mobile nodes. ANSim has been implemented in Java and is therefore easily portable to many platforms.

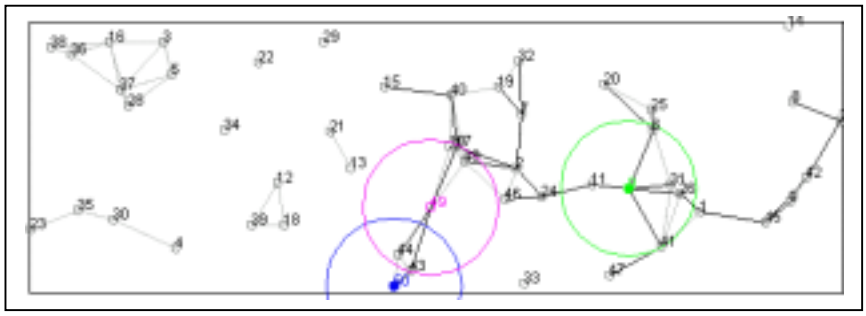
Using ANSim, we found out that many nodes are needed to really establish a relatively stable network, in which reasonable connectivity can be provided. In a standard scenario, each node would need between 5 and 15 directly reachable neighbors for a connection probability of 95%. This is a lot, especially taking into account the interference that kicks in when a few nodes are within reach.

3 Improving Ad-Hoc Network Connectivity using Mobile Robots

As a consequence of these rather discouraging results with respect to the usage of ad-hoc networks, we are now introducing a mechanism to improve the connectivity in such environ-

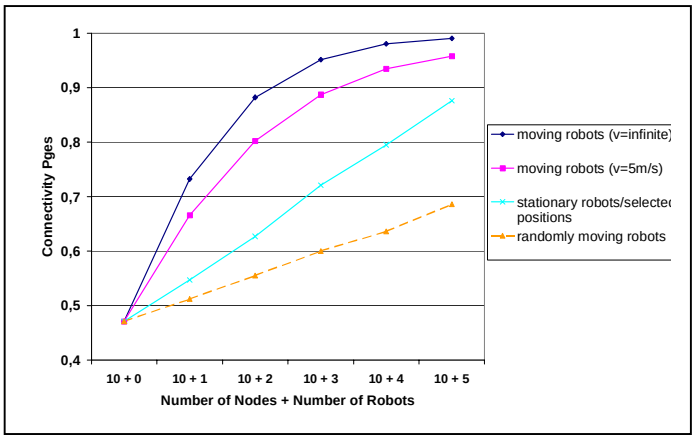
ments. One problem is that all participating nodes have a certain application job to fulfil, i.e., they cannot be controlled in order to keep or improve the current network structure. Thus, the idea is to introduce a number of mobile robots which have the sole task of maintaining the structure. There movements can be controlled in order to fulfil the task

The following picture on the right shows such a situation. Node 49 is in a position to connect node 50 to the larger subnetwork to the right. Node 49 has moved to this position since it has been instructed to do so.



There are a lot of open questions with respect to how to control the robots. It is, for instance, unclear how much knowledge the robots have, and where they get it from. There might also be conflicts for which position to choose. We are currently investigating a benfit and cost function which should help in making optimal or at least very good decisions.

Our first simulation results with mobile robots are shown in the following graph (ANSim has been extended for this purpose). We have four different measurements with randomly moving robots (same situation as with application nodes), stationary robots, robots that move with a certain speed and robots that can move infinitely quickly to the optimal position.



The graph shows the expected results. With infinitely quick robots, we can achieve major improvements over the normal situation with respect to the probability of being able to set up a connection. Even though this is an unlikely scenario, it gives us an upper limit of what can be achieved. The much more realistic scenario with robots at a certain speed shows not much worse performance. It seems that the mobile robots approach is worth following further.

Approaches for improving TCP in Wireless Mobile Ad-Hoc Networks

Ruy de Oliveira

In ad-hoc networks, where changes can happen somewhat quickly and unpredictably, the Transmission Control Protocol (TCP) has to deal with new tough challenges caused by mobility, such as high probability of both network partition and route failures. In order to adapt TCP to this complex scenario, some improvements have been proposed. Nevertheless, most of them present limited enhancements as they do not address important related issues. In this short report, we present a brief introduction to this subject and point out potential approaches for enhancements in this environment. Our early proposal is outlined as well.

TCP was designed to work in wired networks where packet losses can safely be associated to network congestion. So, TCP relies on the assumption that any packet loss is result of congestion inside the network, and appropriately slows down its transmission rate when that happens. In wireless mobile ad-hoc scenario, however, such losses may occur not only by congestion but also due to transmission errors and mobility. Transmission errors induce TCP to mistakenly reduces its sending rate, while mobility induced losses (link breakage) may lead TCP to incredibly long periods of inactivity due to its exponential backoff mechanism. This mechanism imposes an exponential increasing delay at every unsuccessful and consecutive attempt of retransmission accomplished by TCP sender. In situations when a large delay triggering is immediately followed by a link restoration (due to mobility), the exponential backoff mechanism will prevent TCP from retransmitting for a relevant period of time which may be over ”one minute” long. Therefore, TCP needs to be improved in order to detect the actual cause of any packet loss and takes the most appropriated action for each case. Additionally, this environment is also prone to packet reordering and path asymmetry. These issues are well known sources of further problems for TCP, and so work on them is also encouraged.

Concerning the existing proposed solutions for the mentioned problems, it is noticeable that not many appealing proposals exist. Besides, most of them make use of feedback from the network to distinguish the cause of the loss. It is done either by adjusting the routing protocol or using existing protocols such as ICMP and ECN. Although feedback messages from the network provide an accurate information about the network congestion state, in such an environment they may face constraints to reach the end-nodes. The dependence on specific network protocols may be a drawback as well. Additionally, cooperation from intermediate nodes can be hard to be achieved, since it demands changes on those nodes. Because of that, we believe that end-to-end approaches, in which only the end-nodes need to be changed, are more viable in terms of widespread deployment.

The key idea of end-to-end congestion control mechanisms is to provide the TCP sender with updated information about the congestion state of the network. In this way, the sender can combine such information with other elements (e.g., length of last interval without any packet receipt) and appropriately acts. Congestion measuring may be performed by the sender alone or in cooperation with the receiver. RTT (round trip time) monitoring and Relative delay are approaches that require only sender adaptation, while Equation-based congestion control and Interpacket delay approaches rely on sender-receiver cooperation. We restrict our following discussion to the first case, as it is our desired approach.

RTT monitoring refers to the idea used by TCP vegas version, in which the sender monitors round trip times to infer congestion, and by comparison between actual and ideal throughputs it increases or decreases its sending rate. The actual throughput is computed using the RTT of the current received packet and the ideal throughput relates to the smallest RTT seen so far. Two parameters, alfa and Beta, bound the accepted RTT variations. The main advantage of this approach is the simplicity, but pure RTT estimation may be quite

inaccurate in ad-hoc scenario where path asymmetry seems to be a relevant inherent issue. To keep a historical of the last n RTTs may be an efficient improvement to this approach.

Relative delay provides isolation for the forward thoughput from any congestion that might be present along the reverse path. It is done by measuring the delay that packets experience with respect to each other as they propagate through the network. For that, timestamps information are needed in all packets so as to the sender can compute the relative delay by using the arrival time at the receiver, which is carried by the correspondent ACK packet. In addition, the sender needs to store for every packet both the transmission time and the receiving time reported by the receiver. By using the relative delay, the sender can infer whether the congestion is increasing or decreasing in either directions of the connection. This approach seems to be quite attractive for ad-hoc networks, because it is relatively simple and still addresses path asymmetry related problems.

As stated above, packet reordering may be a serious problem in ad-hoc environments, since mobility as well as multipath routing may be in place. Such a reordering will, in most cases, leads the TCP sender to unnecessary retansmissions and reductions in its sending rate. So, alternative ways of dealing with reordering are indeed desired in this scenario. Some existing proposals are outlined in the following.

TCP SACK version along with DSACK extension represent one of the possible options for dealing with reordering. It is important to note that this approach is standardized, and so higher probability in its acceptance is expected. TCP SACK as originally proposed is robust in handling several packet losses within a single transmission window, since it uses selective acknowledgements to trigger retransmissions instead of relying only on duplicate packets or timeout induced retransmissions. On the other hand, it does not address conditions in which a duplicate packet is detected by the receiver, thereby wasting a possibility of informing the sender about such a fact by which it could react faster. Thus, DSACK extension has been added to SACK for solving the mentioned problem. Using DSACK information the sender can take any action it is adjusted for. Alternatively, TCP timestamp option as well as a second sequence number for any segment sent (data or ack), have also been proposed for aiding to detect out of order packets. Regardless of the method used for detecting reordering, the response is always to recover from its effect (throughput drop) by leading the sender to the state prior to the fast retransmit or timeout events (previous higher throughput).

In terms of path asymmetry, some proposals that have been proposed for wired network and may be used here include: TCP header compression, ack filtering, ack congestion control (RED) and ack-first, sender adaptation and decoupling of cwnd growing from the number of acks received (timestamp-based). These schemes attempt to avoid lack of acks at the sender, which is caused by asymmetry and can induce severe throughput degradation. Further analysis on theses approaches toward ad-hoc scenario is needed.

Based on the explanation given above, we outline our early proposal for enhancing TCP performance in ad-hoc scenario. Reordering and path asymmetry treatments will be considered as optional optimizations that may be implemented in a future stage. The main point of our approach is to monitor the congestion state of the network using one of the methods cited above, or derived ones, and act appropriately under losses, as outlined in the pseudo code below.

```
(3 dacks)                          ;ack loss
if(congestion)
   invoke CC                       ;loss due to cong.
else    simply retransmit          ;medium loss
if(timeout)                        ;pck/ack losses or disconn.
   if(cong. & recent pck received) ;(recent pck rec.= % of RTO)
      invoke CC                    ;loss due to congestion
   else
      put sender into "probe mode" ;disconn. under cong. or not
```

The pseudo code above summarizes the sender action when losses are detected by either receiving 3 duplicate packets (3 dacks) or timeout. In case of 3 dacks, the sender checks the network state for congestion. If it gets a positive response, then the normal congestion control (CC) mechanism is invoked once such a condition indicates loss due to congestion. Otherwise, a random loss caused by the medium is inferred, and it justifies a simple retransmission without invoking the CC, since it would force a needless decrease on the sending rate.

TCP sender is led into timeout by losses due to congestion or disconnection (mobility). In order to distinguish between timeout by congestion and timeout by disconnection, being the network facing congestion or not, the sender monitors the instant the last packet for the specific session is received. This is represented by the ”recent pck received” variable in the pseudo code. Using such information, the sender may determine whether a recent packet has been received, which indicates that there is connectivity. If so, the loss is associated with congestion, and then the CC mechanism is called. Otherwise, a disconnection is detected and the sender is put into ”probe mode”, in which the sender will probe the network until it receives a reply from the receiver.

Therefore, using this algorithm, the sender will be prevented from invoking the CC when it is not the best choice, which implies in avoiding the mentioned problem with the exponential backoff mechanism and in improving the end-to-end throughput. The main challenge of our approach regards the mechanism for detecting congestion inside the network accurately, and also the simulation scenario adaptation. These are the next steps to be tackled in our work.

Distance Education and Architecture

Marc-Alain Steinemann

Introduction

This extended abstract intends to give an overview about the architecture designing work done in the context of a distance education project in the field of computer networks.

Distance Education

The group RVS has started to develop Internet-based courses in 2001. An important part of the work is being performed within the project called Virtual Internet and Telecommunications Laboratory of Switzerland [VITELS], which is one of several projects within the Swiss Virtual Campus [SVC] program funded by the Swiss ministry of education and science. Each partner of the VITELS project - four universities (Bern, Fribourg, Genève, Neuchâtel) and one engineering school (Fribourg) - is currently developing modules based on the own competence and equipment. The seven modules focus on Linux System Installation and Configuration, IP Network Simulation, Configuration and Performance Evaluation of a Real IP Network, Client/Server Programming, Protocol Analysis, IP Security, and Firewall Management.
Each participating university develops and maintains its modules within its own laboratory environment, but allows remote students to access and use the laboratory infrastructure via Internet technology. The entire course must appear to the user as being homogeneous, although it is distributed over several locations in Switzerland. A web-learning environment called [WebCT] is used to lead through the course modules.

VITELS placed its partners for the problem of connecting network hardware to the network itself and at the same time giving the students access to the hardware as they would be the owners. As a consequence of this and because there were no already available solutions around, an architecture for connecting network material and almost any type of hardware that can be connected to a computer was designed and implemented.

The architecture allows students to book course modules and to reserve time slots on-line. This procedure was necessary as network hardware is expensive and sometimes only one exemplar is available.

The architecture was designed to be adapted to other authentication systems if needed. Switch started a project called authentication and authorization infrastructure (AAI) for Switzerland in 2000. VITELS is a pilot project partner and tries to adapt the VITELS architecture to the new AAI. For this reason, we are currently developing an AAI portal that allows connecting non-AAI enabled resources to the AAI. Non-AAI-enabled resources are software solutions such as course platforms that were not designed for connecting to an AAI and lack the necessary interfaces.

Figure 1 shows the portal under development and its surroundings. On the upper left side, the home organizations are shown. Home organizations are for example universities. They maintain their user databases and authenticate their users. On the lower left side, users are depicted. Users want to attend courses and connect to resource provider sites, shown on the lower right side. In the case of non-AAI-enabled resources, they get to the AAI-Portal. The portal manages the handling of user data between the AAI and the resources. It allows collecting additional user data and administrational people to interact with the user data.

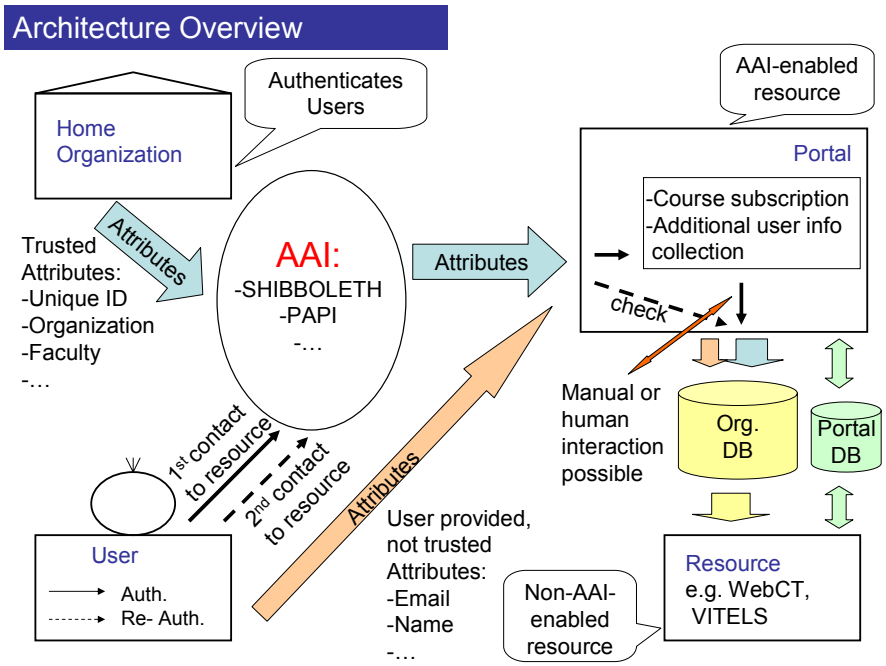


Figure1: AAI-Portal and its surroundings

Conclusions

The VITELS architecture is running and the undergoing extension and design adaptation to the new Swiss-wide AAI can make our developments very attractive for other distance learning projects.

References

[SVC] Swiss Virtual Campus, http://www.virtualcampus.ch/
[WebCT] WebCT, http://www.webct.com
[VITELS] Virtual Internet and Telecommunications Laboratory of Switzerland, http://www.vitels.ch