Automated Negotiation for On-Demand Inter-Domain Performance Monitoring

Lidia Yamamoto

Hitachi Europe, Sophia Antipolis Laboratory
1503 Route des Dolines, F-06560 Valbonne, France
Tel.: +33 4 89874170, Fax: +33 4 89874198
E-mail: Lidia.Yamamoto@hitachi-eu.com

Abstract

Inter-domain network monitoring is vital to provide reliable and high quality services. Nevertheless today many obstacles persist against flexible, responsive, and completely automated monitoring across administrative boundaries. Within the IST 6QM project on IPv6 QoS measurements, one of our goals is to facilitate inter-domain performance measurements, which will become increasingly critical as the Internet will have to deal with multiple protocols families. This paper presents an overview of current 6QM activities in this area. We start with a description of ongoing standardization efforts, and then focus on our most recent research activity, which is still work in progress, on the usage of automated negotiation techniques for inter-domain monitoring.

1. Introduction

The Internet is made possible through the cooperation among multiple independent administrative domains. Inter-domain network monitoring is crucial to provide reliable and high quality services. Monitoring is necessary for several important tasks such as fault diagnosis and repair (troubleshooting), verification of conformance to Service Level Agreement (SLA) in networks with Quality of Service (QoS) support, accounting, detection of attacks, accumulation of statistics for network planning and engineering, etc. Today most of these tasks are not yet automated, and require human intervention to establish agreements between peer domains, to launch new monitoring tasks, or to visually inspect the results.

There are many obstacles against fully automated, global-scale performance monitoring. Security and privacy are the stronger ones. End users do not wish their traffic to be eavesdropped along the path, and network providers do not wish to reveal the internal configuration of their networks to untrusted parties. Moreover, the amount of data corresponding to monitoring results might be overwhelming, and handling such mass of information is a potential source of performance degradation. As a consequence, monitoring an arbitrary end-to-end path today is difficult and restricted, and the obtained information is very limited and inaccurate.

Since no central authority controls all domains, inter-domain monitoring is inherently distributed and decentralized. Cooperation among domains cannot be taken for granted, and pre-configured measurement tasks might not suit the need for fast response time required in applications such as troubleshooting and detection of attacks. It is necessary to foster cooperation between providers for the execution of measurement tasks and the provision of the corresponding results.

The first step toward this goal is to obtain secure and standardized means to exchange monitoring requests and results. A number of standardization efforts are in progress to address this step [6, 25, 30, 37]. However this is not sufficient to provide responsive on-demand monitoring services that take changing network conditions into account.

As the next step, we would like to have an automated way to dynamically agree on which parameters may be monitored across domains, depending on the resources available, the current network conditions, the trust levels among providers, and other policies and constraints. We propose to apply automated negotiation techniques [27, 28] in this context. Automated negotiation mimics human negotiation processes to reach agreements on one or more issues. This could be used to agree on parameters for the set-up of measurement tasks across domains and upon demand. We have identified the potential protocols and strategies that could be applied, and mapped monitoring parameters to them.

This paper is structured as follows: Section 2 provides the basic conceptual background and an overview of the current state of the art in inter-domain monitoring techniques, including on-going contributions to standardization by members of the 6QM team. After that it presents an introduction to the research field of automated negotiation in multi-agent systems. Section 3 proposes a generic

Proceedings of 2nd International Workshop on Inter-Domain Performance and Simulation,
Budapest, Hungary, March 2004.

proxy-based architecture for measurement set-up and export. Within this architecture, an agent-based automated negotiation mechanism is responsible for the set-up process. The protocol and strategy to achieve this are described in Section 4. This is still work in progress, so our conclusions are preliminary. We present them in Section 5, together with a discussion on the open issues and next steps.

2 Background and related work

In this section we briefly explain the basic conceptual background concerning inter-domain monitoring, and present an overview of the current state of the art in the area, with focus on performance measurement mechanisms. We describe on-going contributions to standardization by members of the 6QM team, and provide an introduction to automated negotiation techniques.

Network monitoring is one of the basic roles of a network management system (NMS). One may monitor several network parameters at several layers of the protocol stack, including configuration parameters, state of various hardware elements, accounting information, etc. Performance monitoring is the branch of network monitoring oriented toward the measurement of network performance parameters such as delay, loss, jitter, etc. Performance measurement mechanisms may be passive or active. Passive mechanisms capture existing packets, while active mechanisms inject test packets into the network. The IETF IP Performance Metrics (IPPM) working group [26] has already standardized several performance metrics and accompanying active measurement methodologies.

Inter-domain QoS measurement can be divided into three phases:

- *Measurement set-up*: This phase comprises the initial agreement between providers involved in a given measurement task, and the corresponding configuration of the elements involved in the requested measurement.

- *Measurement task execution*: In the case of passive measurements, passive meters located at strategic positions in the network, or meter components within routers, are activated to run a specified measurement task. In the case of active measurements, an important part of task execution is the recognition and treatment of standard test packets by active probes [34].

- *Measurement result exportation*: After a measurement task is executed, standard formats are needed for the exchange of measurement results so that they can be unambiguously interpreted in different domains.

Currently none of these three phases is completely automated nor sufficiently reliable. Measurement set-up and exchange of measurement results across domains is still relatively rare in practice, falling far short of what is needed for reliable troubleshooting, QoS-based services, and other applications.

Simple and popular active measurement tools such as *ping*, *traceroute*, and *pathchar/pchar* [8] work only if the domains involved allow such traffic in their networks. However many network managers decide to disable this traffic for security reasons. Moreover these tools do not provide results in standardized format.

In Europe, RIPE NCC [32] offers the TTM Service (Test Traffic Measurements) [33] that collects measurement data from sites in order to enable a proactive monitoring of the network. This service requires a test box that is installed at each measured site and managed by RIPE in a centralized fashion. A RIPE test box is an active measurement probe able to measure one-way delay, one-way packet loss, traceroutes, and bandwidth capacity. The delay and loss measurements comply with IPPM [26] standards. IPv6 is supported since 2003. The measurement results are supplied to RIPE participants via a password-protected web site access to a centralized database and analysis server. Fully meshed measurements are performed, which clearly presents a scalability problem. The RIPE TTM service allows inter-domain measurements, but they are managed in a static fashion and are centrally controlled. This is not always acceptable by the authorities of a domain. The service does not support reactive applications such as troubleshooting, attack detection, or quick response to new service demands.

In many cases of commercial deployment, the reality is even cruder, relying on phone calls among network engineers in order to monitor the network and pinpoint eventual problems.

2.1 6QM Activities

The 6QM project [1] is working on IPv6 QoS measurements, with focus on the development of a passive measurement platform. Within this project, the OpenIMP platform has been designed and implemented, and is going to be released soon [29]. A overview of its design is available on 6QM deliverable D3.1 [7]. This platform was developed during the first phase of the project which focused on intra-domain measurements. Inter-domain measurements, among other subjects, are being incorporated during the second phase of the project (in progress).

Several members of the 6QM team are actively involved in standardization efforts related with QoS measurements, with emphasis on IPv6 support. Most of the contributions are also important in the inter-domain context. In this section we summarize these contributions.

In [36] a Management Information Base (MIB) registry for the IPPM metrics is being defined. It assigns a MIB Object Identifier to each currently standardized metric, and defines rules to add future metrics. The IPPM Reporting MIB [37] is a draft document in which a MIB architecture and corresponding objects are being specified for managing and reporting results of IPPM-compliant measurements. The MIB architecture extends the RMON model [39]. The RMON MIB specified in [39] supports a single point of measure, while in [37] multiple points of measurement must be supported, recognizing common time references and measure identification. Members of 6QM are working on a proposal to extend RMON with IPv6 protocol identifiers [38].

Three main measurement architectures are proposed in [37]:

- *Proxy architecture*: In this architecture, the IPPM Reporting MIB agent acts as intermediary between the NMSs requiring measurement data and the subsystems that actually collect such data (the points of measure). The Reporting MIB agent is responsible for access control from the NMSs to the data they are allowed to receive. The communication between the MIB agent and the NMSs is done via standard SNMP using the objects defined in the draft [37]. On the other hand, the communication between the MIB agent and the points may be done using proprietary protocols to allow a lightweight implementation of measurement points.

- *Reporting architecture*: In this architecture SNMP is used directly between the points of measure and the NMSs. In this case the points of measure must implement the IPPM Reporting MIB agent.

- *Gateway architecture*: This architecture combines elements of the proxy and reporting architectures. An IPPM Reporting MIB gateway registers the queries from the NMSs, but the allowed NMSs can directly consult the results from the point of measure using the MIB via SNMP.

The Reporting MIB proposal [37] extends the existing MIB model to IPPM metrics, and uses standard SNMP to access results. While this is conceptually easy, it is not so clear if this will be an ideal model in practice. The MIB defined is quite complex, even in the current state where active measurements are the main focus. Therefore it is likely to become even more complex when passive measurements and other new metrics are added.

This approach is proactive, and does not support on-demand monitoring. The risk is that it might produce too much information which is not always used, or on the other hand might also produce too little information for certain applications. Using current MIB data it is difficult to obtain end-to-end performance on arbitrary paths.

The proposal [37] is not specific for inter-domain measurements, but it recommends the use of “view based access control” VACM for inter-domain. VACM rules specify the access privileges of each user registered in the MIB. For example, the network administrator may be given write privilege to most parts, while other users may have read-only access to certain parts and denied access to others. While access control is essential for any inter-domain interaction, it is not enough for full automation of these interactions, since the dynamic aspects are not taken into account.

Provider policies currently either accept or deny a query within their domains, with deny being the default policy and accept only in exceptional cases. Only a few privileged domains may be authorized to obtain results. This leads to lack of flexibility, incomplete information, and reduced usefulness of the few painfully obtained measurement results.

Besides that, the proxy-based access authorization scheme grants access to predefined parts of the MIB and predefined actions on the results (e.g. aggregation). What is allowed or not allowed is pre-configured manually by the network manager. Access control is therefore only qualitative, not quantitative. It does not take dynamic network conditions into account. A domain might thus unwillingly saturate a link due to too many exported results, since there is no quantitative access control.

The IETF IP Flow Information Export (IPFIX) Working Group [25] is defining a protocol and data format for exportation of measurement results. Within IPFIX, 6QM members have co-authored draft submissions on its applicability [41] and requirements. In principle IPFIX allows results of passive measurements to be exported in a standard format. However, IPFIX specification today is focused on global statistics such as first and last timestamp for the flow. It is thus designed for coarse-grain monitoring, with accounting as the main application, rather than fine-grain packet-level traces that could enable automated diagnosis and reconfiguration. Within 6QM, Pohl et al. [30] propose an extension of the IPFIX protocol that allows to export packet information, thus enabling full export of passive measurement results.

Measurement results can represent an overwhelming amount of data. An article on the Sprint passive monitoring infrastructure [23] reports more than one Terabyte of collected data per day, and about 12 hours to transfer this volume to a repository. Therefore exporting measurement data to other domains raises obvious concerns regarding resource usage: it should not negatively interfere with user traffic sharing the transmission link, and should not overload the available storage capacity.

There are several ways to reduce the amount of information exported. First of all, on-demand monitoring will ensure that only the data that is needed is collected, in contrast with current proactive monitoring which attempts to collect every possible parameter. Additionally, sampling and filtering techniques can be used. The PSAMP IETF Working Group [31] is defining standards for these techniques. The current PSAMP draft by Zseby et al., [40] whose first author is a 6QM member, gives an overview of sampling and filtering techniques, and also describes how they can be combined. However it does not define how to adjust filtering and sampling parameters according to dynamic network conditions. To further reduce the amount of data, aggregation can be used to produce coarser-grain statistics.

Another important contribution that can have an impact on inter-domain measurements is the definition of spatial metrics in [35], which allows to obtain end-to-end performance by aggregating a sequence of measures corresponding to each segment of a path. It relies on a standard test packet signature, which is being proposed in [34].

2.2 Other approaches to inter-domain monitoring

Within the IST INTERMON project [24] a Service Level Indication (SLI) document format is being defined [6]. It uses XML syntax to exchange QoS monitoring information across domains. Monitoring information related to an SLA is retrieved from within the domain and provided to a user application in another domain. A CADENUS Resource Mediator is responsible for managing the available resources within the domain, including node configuration. The Resource Mediator also converts raw monitoring information into an SLI document. This seems more concentrated on offering information to the customer instead of pure inter-domain peer exchanges. Besides that, it is still not clear what protocol will be used to request monitoring information to another domain, or the protocol used to make the SLI documents available to the user.

With active measurements, test packets follow the routes computed by the underlying inter- and intra-domain routing systems. Therefore route changes automatically divert test packets to the new route. The same does not happen with passive measurements. Given the packet capturing rules for the desired measurement (source/destination address/prefix) each domain must find out which passive measurement elements must be activated.

One way to do this is to inspect the dynamic routing tables, and track any route changes. Route changes must then trigger reconfigurations, which can be complex in the inter-domain case: measurement tasks that had been already configured in previous domains might now need to be migrated to other domains.

Another solution is to use in-band signaling for measurement set-up [2, 5]: signaling packets set up measurement tasks within the measurement points along the path. The advantage of the latter approach is that signaling packets follow the routing path, thus naturally accommodating route changes without any special procedure. This approach has the same advantage as active measurements, but also the same drawback of injecting extra packets into the measured path, with the risk on interfering with the measurement results. Another shortcoming of in-band signaling is that it does not support measurement of traffic aggregates specified by a network prefix. Moreover, in case of failure of the measured data path, and necessary diagnosis via the monitoring system, relying on signaling messages on the same (perhaps faulty) path might be not a good idea.

2.3 Overview of Automated Negotiation Techniques

Automated negotiation is an active research topic in the agents field, and has been applied to several areas. A good introduction and survey can be found in [27], and an overview of more realistic and complex scenarios can be found in [28]. There are several applications in the telecommunications and computer networking areas. For example, a number of agent-based systems to enable provider selection and inter-domain interactions have been proposed [3, 4, 9, 11, 12].

The use of Agent Communication Languages enables rich and flexible interactions, which can be made interoperable through standardized specifications provided by the FIPA consortium [22]. Several FIPA standard protocols and languages are available which could be applied to inter-domain interactions: Contract negotiation [15, 18], brokering [14], proposals [20], auctions [16, 17], QoS [21], network management [19]. As an example, Faratin et al. [9] present a FIPA-compliant multi-agent system for automated negotiation applied to inter-domain VPN provisioning.

Cascade negotiation toward an end-to-end service is only partially supported in existing approaches. In the case of [4] the first domain agent (in the source domain) communicates directly with all the other domain agents on the path to a given destination. For instance, if a path is made up of domains A, B, C and D in sequence, with A as source domain and D as destination domain, the agent in domain A sends negotiation messages to B, C, and D. It would be more transparent if A would negotiate with B, B with C, and C with D following the path sequence. This is important for inter-domain measurements: due to authorization issues, it is not realistic to maintain peer agreements with all possible domains worldwide.

In practice negotiations cannot last forever: they must usually be completed before a deadline, therefore they are time-constrained [13]. Agents have a utility function, defined as the benefit of receiving a certain bundle of negotiated goods or services. An agent is said to be patient when it gains utility as the deadline approaches, and impatient when the utility decreases.

In single-issue negotiation, only one parameter (typically the price of a demanded good or service) is negotiated. Multi-issue negotiation involves several parameters, typically the prices of different goods, but also other parameters such as delivery time, quality of service, payment methods, etc. There are two approaches to multi-issue negotiations: the easiest one is to discuss the issues one by one, in sequence: after an agreement on the first issue is reached, the second issue is discussed, and so forth. Another approach is to discuss all the issues simultaneously. In this case the trade-off among issues can be exploited [10]. For example, an agent might be willing to accept a later delivery date if the quality is higher or the price lower. This makes the negotiation more complex but also more realistic. Fatima et al. [13] have proposed a model for multi-issue negotiation under time constraints, in which agents have incomplete information about each other. Although their protocols support simultaneous discussion over multiple issues, they adopt a sequential approach: issues that have already been agreed upon are removed from message exchange and the trade-off among issues is not taken into account. Faratin et al. [10] devise a heuristic strategy to handle issue trade-off, and show that it increases the social welfare of the system.

3 Architecture for inter-domain negotiations

We proposed an architecture for generic inter-domain negotiations based on proxy agents that negotiate agreements on behalf of their domains, and perform the corresponding configuration tasks within their respective domains once agreements are reached.

A sketch of the architecture is presented in Figure 1. Each domain is represented by a Proxy Agent that is in charge of all the inter-domain negotiations, and of retrieving the associated results.

The agents communicate with each other using a standardized language and transport protocol. An agent dynamically obtains information from other elements within the domain, about the domain policies and current network conditions, and uses this information to make decisions.

Figure1 also depicts the fact that domains might share a single transmission pipe between themselves, stressing the importance of controlling the agent traffic such that it does not interfere with the traffic from the real users.

As in [6], we would like to view inter-domain network monitoring as a service provided from a server domain to a client domain. As such, it might even have a price, so price can be one of the parameters to be negotiated. Indeed charging for monitoring services could act as an extra incentive for providers to offer them more widely.

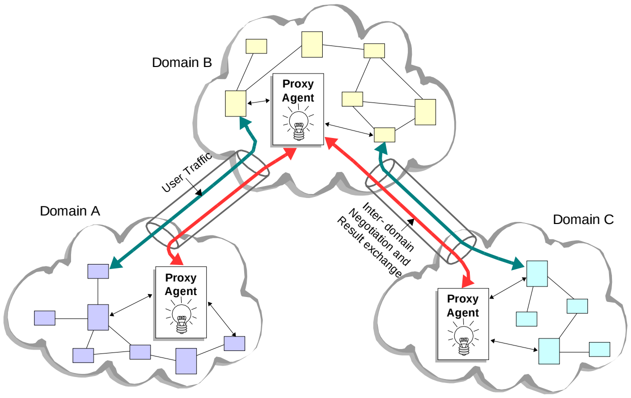

Figure 1. Inter-domain architecture

Although we focus on monitoring services, the architecture proposed could be used for inter-domain negotiation of any other service such as VPN provisioning as in [9], provider selection as in [11, 12], outsourcing of resources, etc.

For the performance measurement service, the domain's proxy agent negotiates on the measurement parameters, accuracy, amount of data to be exported, such as to respect local policies and current network conditions, mainly to avoid overloading the network with measurement data. After an agreement is reached, the agent issues commands within its domain to set up the corresponding measurement tasks.

As oppose to the Proxy MIB architecture [37] discussed in Section 2.1, the architecture presented in this section does not rely on SNMP and the MIB model, although it could use them when needed. Moreover, for the remainder of this paper we concentrate on passive measurements, which are currently not the focus of [37].

3.1 Intra-domain architecture

Figure 2 shows a more detailed view of the architecture, inside a domain. For inter-domain negotiations, the Proxy Agent may receive new requests from the network manager, requiring a given service from another domain; or from other domains requesting a given service.

In the context of this paper, the service is always a network performance measurement task, but the same idea applies to any service, provided that: *(i)* the agent has been programmed with strategies to deal with each kind of service, *(ii)* the necessary interfaces are available to the agent

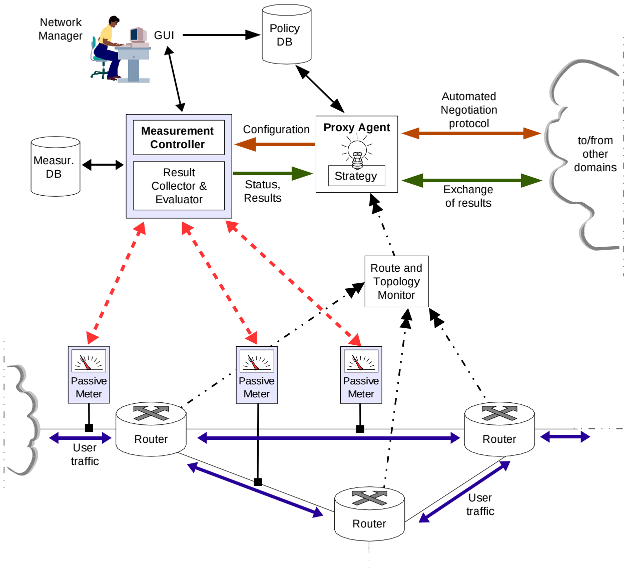

Figure 2. Intra-domain architecture

so that it can obtain information about network conditions and configure network elements within the domain.

The measurement controller, collector/evaluator, and passive meters are part of the 6QM measurement architecture defined in [7] and implemented in the OpenIMP platform [29]. For intra-domain measurements, the network manager specifies the desired measurements via a web-based user interface, which communicates with the measurement controller. To activate new measurement tasks, the controller issues measurement commands to the passive meters. During task execution, at specified intervals, the meters send their measurement data to the collector/evaluator, which then calculates the corresponding metrics and stores the results in the measurement database. The network manager may then visualize the results via the user interface.

After receiving a negotiation request for an inter-domain measurement task, the agent must determine whether the request should be granted, denied, or if a counter-proposal should be generated. This is part of the agent's negotiation strategy. In order to make such a decision, the agent needs information about the current state of the domain in terms of policies and dynamic network conditions. It obtains policy information from the policy database, and meter information via the measurement database. The topology and route monitor provides information on network conditions to the agent in a transparent way, using an interface that is independent on routing or network management protocols.

If the internal policies determine that the request must be denied, the agent sends a denial message to the requesting entity and goes no further. Otherwise, using the obtained meter and routing information, the agent is able to determine which measurement points should be activated for a given measurement task. It then uses meter information again to check whether the concerned points have enough resources to perform the task. Based on the requested parameters, the agent may be able to estimate the amount of data that will be exported, and evaluate whether this amount can be supported with current resources. When available this information can be of great assistance in the decision process. After a decision is made, the agent generates and issues the corresponding commands to the measurement controller, which processes them as if they had come directly from the user interface to the network manager. This helps automating the process of measurement set-up across domains. We now describe the automated negotiation mechanism in more detail.

4 Automated negotiation mechanism

The mechanism is divided into three parts:

- *Negotiable parameters*: within the set of all parameters requested for a given service, only a few might be negotiable.
- *Negotiation protocol*: defines the semantics of the messages to be exchanged, how they are encoded and transported over the network.
- *Negotiation strategy*: specifies the agent's internal decision algorithms used to obtain the desired negotiation outcomes.

4.1 Parameters of the monitoring service

The INTERMON project [24] is defining a document format for the Specification of Monitoring Service (SMS), which should contain all the necessary parameters for inter-domain QoS monitoring. We are working to keep our list of parameters essentially compatible with the INTERMON SMS format, however we wish to adapt it to the specific case of automated on-demand measurements, in which the recipients of the measurement results might not be human beings directly but software agents intended to interpret the results in order to perform diagnosis or other tasks.

We do not attempt to provide an exhaustive list of parameters. It is also important that the specified format be open enough to accommodate new parameters that might be incorporated in the future.

An agent requesting monitoring service from another domain must specify at least the following parameters in the negotiation request message:

- *Flow description*: describes the flow to be monitored in terms of rules to be applied to the monitored packets to identify the flow, such as source address, destination address, port numbers, protocol, and other packet fields. It is important to be able to measure traffic aggregates, and not only single flows: this is crucial in inter-domain measurements where a huge amount of flows traverse a transit domains.

- *Time schedule*: start and end of monitoring task.
- *Metrics*: the performance parameters to be measured, e.g. one-way delay, loss, jitter, throughput, average packet or bit rate over a specified interval, etc. Each metric may have the following associated attributes:
  - *Notification threshold*: value that triggers a notification to the client domain when exceeded.
  - *Report schedule*: interval for sending periodic reports to the client domain.
- *Report format*: format in which measurement results should be sent to the requesting domain. A suitable standard format, or set of standards according to each metric, must still be agreed upon. Starting points are for instance [6, 30, 37].

The INTERMON SMS format contains other information not treated here:

- *Scope*: ingress and egress points of the traffic flow. In our case, the proxy agents determine these points from the source/destination address or prefix specified in the Flow description, together with route information provided by the Route Monitor.
- *Report destination address* (e-mail, postal, fax,...): this is oriented toward delivery of results to a human customer. In our case, the results are delivered to the agent that requested the service.
- *Security parameters* (authentication data and encryption service): In our case we assume that the Proxy Agent runs over a secure transport connection, such that the domain identification of the peer agent can be assumed to have already been properly authenticated. Moreover the communication over the secure connection is assumed to be encrypted for privacy when needed.

We must now define which parameters are negotiable, among the previously selected ones (flow description, time schedule, metrics, notification threshold, report schedule, and report format). A non-negotiable parameter must be accepted as is, otherwise the measurement task becomes infeasible. On the other hand, a negotiable parameter admits some flexibility within a range of values, in which the measurement task remains feasible but with different accuracy or resolution.

In principle, the flow description and metrics cannot be negotiated. One could imagine that for a flow described in terms of a network prefix, the prefix length could be negotiated: a longer prefix would mean that less packets are captured, and depending on the purpose of the measurement task this could be sufficient. However this is difficult to quantify in practice. For simplicity we will not consider this possibility.

The other parameters (time schedule, notification threshold, report schedule, report format) are all negotiable in general: A shorter time schedule, or a shift in time schedule, can make a measurement task acceptable for a server domain, while still useful for the client domain. The notification threshold can be adjusted in order to raise less alarms. The report schedule interval can be increased in order to reduce that amount of exported data. The report format can be chosen such as to generate an acceptable amount of data.

All the negotiable parameters go in the direction of saving resources by reducing the amount of information exported. Other parameters should be added to this list. The most important one is accuracy information: the domains must agree on the exact precision of the results in order to be able to interpret them in an unambiguous manner. The precision obviously also has an impact on the amount of information exported, since higher precision values require larger fields to hold them. Sampling and filtering parameters can also be added to control the trade-off between the amount of information obtained and the resources needed.

Besides the parameters of the monitoring service, there are also parameters related to the negotiation itself. The most important parameter is the deadline of the negotiation (timeout).

4.2 Negotiation Protocol

The proposed negotiation protocol is essentially an instance of the FIPA Iterated ContractNet Interaction Protocol [18]. The choice of an existing protocol has the advantage of dispensing the network community from a potentially long standardization process.

FIPA Iterated ContracNet defines the exchange of messages between an Initiator Agent and one or more Participant Agents. The Initiator issues a Call For Proposals (`cfp` act) to every Participant. Within a given deadline, each Participant may refuse the `cfp` (`refuse` message) or reply with a proposal (`propose` message). The Initiator evaluates all the received proposals and responds with `reject-proposal`, `accept-proposal`, or a new, revised `cfp`. In the latter case, a new iteration takes place, with new proposals being evaluated, and so on, until an agreement is reached (i.e. at least one of the proposals is accepted), or the Initiator decides to stop (i.e. reject all proposals, either because they are not satisfactory or because a deadline is reached).

The Iterated ContractNet protocol is very generic and does not specify details of the negotiated parameters. In the case of measurement services it is not necessary to issue multiple calls for multiple agents (that would be the case in a service provisioning request, for example, in which the client domain would issue several concurrent calls to competing domains, in order to choose the most interesting service offer). Based on this, we have refined the contents of the messages to be exchanged as:

- *Request*: This is the first message sent from the domain that requests the monitoring service (client domain) to the domain that is expected to provide the service (server domain). It is equivalent to a `cfp`, but specialized for this service. It contains the selected monitoring service parameters described in Section 4.1. The format is:

  request(seqno,service)

  where: *seqno* is a sequence number that uniquely identifies the current request within the negotiation; *service* contains the list of <*variable,value*> pairs that describe the desired characteristics of the requested service, in terms of the selected parameters of Section 4.1: flow description, starting time, finish time, list of metrics with corresponding notification threshold and report schedule if any, and report format.

- *Propose*: This message is issued by the server domain to say that it is willing to offer the requested service. However it may suggest changes in one or more of the negotiable parameters of a previous *request* message. The format is:

  propose(seqno,proposal)

  where *seqno* is the sequence number of the corresponding *request* message, and *proposal* is a list (possibly empty) of <*variable,value*> pairs containing the new proposed values for a number of parameters. If the list is empty it means that all requested parameters have been accepted. The client agent may or may not accept the proposal. If accepted, it issues an *Accept* message for *seqno*, otherwise it may issue a *Reject* message or a new *Request* message with revised parameters (and a new *seqno*).

- *Accept*: This message indicates that the previous proposal has been accepted. Format:

  accept(seqno)

  where *seqno* is the identifier of the corresponding *Propose* message. This message successfully terminates the negotiation. The outcome is the *Request* whose *seqno* is mentioned in the proposal, modified with the new parameter values proposed in corresponding *propose* message.

- *Refuse*: This message is issued by the server domain in order to categorically refuse a previously issued *Request* message. This message causes the negotiation to abort. Format:

  refuse(seqno)

  where *seqno* is the sequence number of the corresponding *Request* message.

- *Reject*: This message is issued by the client domain in order to say to the server domain that it rejects its previous proposal. Format:

  reject(seqno)

  where *seqno* is the corresponding proposal identifier. This message causes the negotiation to abort unsuccessfully.

In addition to the abovementioned parameters, all messages contain a *negid* (Negotiation Identifier) parameter (not shown) to uniquely identify a given negotiation between the two agents involved. It can be formed, for instance, by concatenating the requesting agent's Autonomous System (AS) number with a locally generated sequence number. This allows for multiple negotiations to be handled in parallel.

An example of a typical negotiation interaction is shown in Figure 3. It involves two domains, A (the initiator or client) and B (the participant or server). Domain A requests a service with two negotiable parameters p_1 and p_2. It sends the negotiation request with the desired values for each parameter (i.e. optimum from A's point of view). The agent from B evaluates the request and makes a new proposal, more advantageous from B's point of view (changing the value of p_1 and accepting p_2 as is). After evaluating B's proposal, the agent from A decides to send a new request with some modified parameters, hoping to achieve a better deal. B then proposes a compromise solution ($p_1 = 4$) which is finally accepted. After that the agent from B issues the necessary commands within its domain such that measurement set-up can take place according to the negotiated parameters.

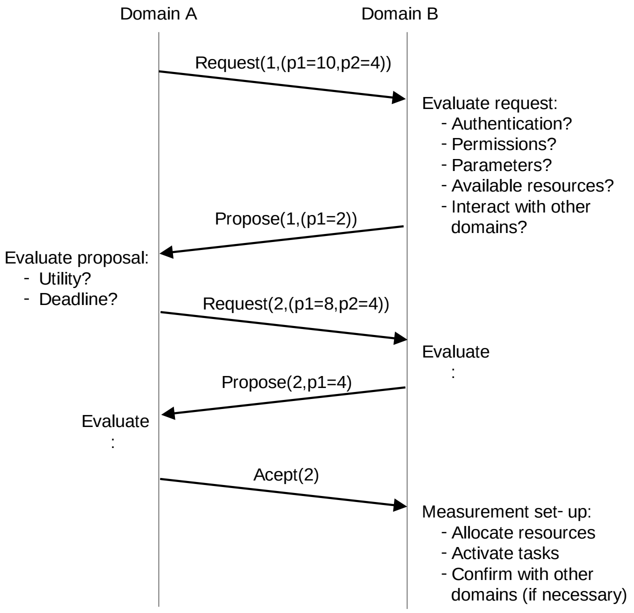

Figure 3. Typical inter-domain negotiation

4.3 Agent Strategy

The agent strategy is not part of the negotiation protocol, therefore can be kept secret. It is indeed in the best interest of each domain to do so, since the agent that has a good negotiation strategy can win competitive advantage by negotiating agreements that are highly beneficial for the domain's owner. Moreover, an agent should not reveal its negotiation deadline, since its opponent could exploit this knowledge to push its own selfish interests (for instance, by offering a very high price to an agent with a short deadline, hoping that the agent accepts the offer because it is in a hurry to reach an agreement).

Extensive studies on negotiation strategies are available from literature [9, 10, 13, 27, 28]. We have selected a few deemed suitable for the case of a network performance measurement service. First of all, since the agents have deadlines, we restrict ourselves to those strategies especially designed for time-constrained agents. The strategies described in [10, 13] seem very suitable, also because they are able to deal with multiple issues. We recall from Section 2.3 that while the strategy in [13] evaluates each issue independently, strategy [10] considers the trade-off among different issues.

We are currently mapping the parameters of the service as listed in Section 4.1 to the selected strategies [10, 13]. We should soon have a proof-of-concept prototype in order to evaluate these strategies experimentally over a running IPv6 network.

5 Conclusions and Future Work

With the evolution in network services and applications, networks will tend to become more and more interdisciplinary. Much can be learned from other areas that can be applied to networks to make them more effective and easy to manage. The present paper is an attempt to apply lessons from agent technology to improve cooperation between different administrative domains in the important task of network monitoring. We believe that a well-designed automated negotiation mechanism could enable on-demand agreements for the dynamic set-up of measurement tasks across domains, similar to the way goods can be purchased in electronic markets. This could act as an incentive for cooperation, as providers that cooperate to offer monitoring results would be in a better position to offer higher quality services appreciated by customers, and to promptly react to customers' requests.

To the author's best knowledge, this is the first proposal to apply automated negotiation to inter-domain measurement set-up for network monitoring. Existing related work concentrates on access selection [11, 12], and service provisioning [3, 4, 9].

A proof-of-concept prototype will be implemented soon, in order to validate the proposed approach in a quantitative way, to refine the specification of the message exchange standard languages and protocols between domains, and to test different negotiation strategies in practice.

Acknowledgments

This work has been performed within the IST project 6QM [1], which is partially funded by the European Commission.

References

[1] *"6QM - IPv6 QoS Measurement"*, 2002. IST Project IST-2001-37611, `http://www.6qm.org/`.

[2] D. Agarwal, J. M. Gonzalez, G. Jin, and B. Tierney. *"An Infrastructure for Passive Network Monitoring of Application Data Streams"*. In Proceedings of Passive and Active Measurement Workshop (PAM 2003), La Jolla, California, USA, April 2003.

[3] M. Calisti and B. Faltings. *"Agent-Based Negotiations for Multi-Provider Interactions"*. In Proceedings of 2nd International Symposium on Agent Systems and Applications (ASA 2000), Zurich, Switzerland, September 2000.

[4] M. Calisti and B. Faltings. *"Distributed constrained agents for allocating service demands in multi-provider networks"*. Journal of the Italian Operational Research Society, Special Issue on Constraint-Based Problem Solving, XXIX(91):199–215, 2000.

[5] A. Couturier. *“Signaling for QoS Measurement”*. IETF Internet Draft (work in progress), `draft-couturier-nsis-measure-00.txt`, May 2003. expires November 2003.

[6] S. D’Antonio, M. Esposito, M. Gargiulo, S. Romano, and G. Ventre. *“A Component-based Approach to SLA Monitoring in Premium IP Networks”*. In First international workshop on Inter-domain performance and simulation (IPS 2003), Salzburg, Austria, February 2003.

[7] D. Diep (Editor). *“IPv6 QoS Measurement Specification”*. 6QM Deliverable D3.1, `http://www.6qm.org/files/6qm_pu_d3_1_v6_4.pdf`, April 2003.

[8] A. B. Downey. *“Using pathchar to Estimate Internet Link Characteristics”*. In Proceedings of ACM SIGCOMM’99, Boston MA, USA, September 1999.

[9] P. Faratin, N. Jennings, P. Buckle, and C. Sierra. *“Automated Negotiation for Provisioning Virtual Private Networks using FIPA-Compliant Agents”*. In The Fifth International Conference and Exhibition on the Practical Application Of Intelligent Agents And Multi-Agent Technology (PAAM-2000), pages 185–202, Manchester, UK, 2000.

[10] P. Faratin, C. Sierra, and N. R. Jennings. *“Using similarity criteria to make issue trade-offs in automated negotiations”*. Journal of Artificial Intelligence, Elsevier Science, 142(2):205–237, 2002.

[11] P. Faratin, J. Wroclawski, G. Lee, and S. Parsons. *“The Personal Router: An Agent for Wireless Access”*. In In Proceedings of the AAAI Fall Symposium on Personal Agents, pages 13–21, N. Falmouth, Massachusetts, US, 2002.

[12] P. Faratin, J. Wroclawski, G. Lee, and S. Parsons. *“Social User Agents for Dynamic Access to Wireless Networks”*. In Proceedings of the AAAI Spring Symposium on Human Interaction with Autonomous Systems in Complex Environments, Stanford, PA, US, 2003.

[13] S. S. Fatima, M. Wooldridge, and N. R. Jennings. *“Multi-Issue Negotiation Under Time Constraints”*. In First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002), Bologna, Italy, July 2002.

[14] *“FIPA Brokering Interaction Protocol Specification, SC00033H”*, December 2002. In [22].

[15] *“FIPA Contract Net Interaction Protocol Specification, SC00029H”*, December 2002. In [22].

[16] *“FIPA Dutch Auction Interaction Protocol Specification, XC00032F”*, August 2001. In [22].

[17] *“FIPA English Auction Interaction Protocol Specification, XC00031F”*, August 2001. In [22].

[18] *“FIPA Iterated Contract Net Interaction Protocol Specification, SC00030H”*, December 2002. In [22].

[19] *“FIPA Network Management and Provisioning Specification, XC00082B”*, August 2001. In [22].

[20] *“FIPA Propose Interaction Protocol Specification, SC00036H”*, December 2002. In [22].

[21] *“FIPA Quality of Service Ontology Specification, SC00094A”*, December 2002. In [22].

[22] *“Foundation for Intelligent Physical Agents (FIPA)”*, 2003. `http://www.fipa.org/`.

[23] C. Fraleigh, C. Diot, B. Lyles, S. Moon, P. Owezarski, D. Papagiannaki, and F. Tobagi. *“Design and Deployment of a Passive Monitoring Infrastructure”*. In Proceedings of Passive and Active Measurement Workshop (PAM 2001), Amsterdam, The Netherlands, April 2001.

[24] *“INTERMON: Advanced architecture for INTER-domain quality of service MONitoring, modelling and visualisation”*, 2003. IST Project IST-2001-34123, `http://www.ist-intermon.org/`.

[25] *“IP Flow Information Export (ipfix) IETF Working Group”*, 2003. `http://www.ietf.org/html.charters/ipfix-charter.html`.

[26] *“IP Performance Metrics (ippm) IETF Working Group”*, 2003. `http://www.ietf.org/html.charters/ippm-charter.html`.

[27] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. *“Automated negotiation: prospects, methods and challenges”*. International Journal of Group Decision and Negotiation, 10(2):199–215, 2001.

[28] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. *“Protocols for Negotiating Complex Contracts”*. IEEE Intelligent Systems Journal, special issue on Agents and Markets, 18(6):32–38, 2003.

[29] *“OpenIMP Project: Internet Measurement Platform”*, 2004. hosted by SourceForge.net, `http://sourceforge.net/projects/openimp/`.

[30] G. Pohl, L. Mark, C. Schmoll, and T. Zseby. *“IPFIX Export of packet information for QoS Meaurements”*. IETF Internet Draft (work in progress), `draft-pohl-pktid-00.txt`, October 2003. expires May 2004.

[31] *“Packet Sampling (psamp) IETF Working Group”*, 2003. `http://www.ietf.org/html.charters/psamp-charter.html`.

[32] *“RIPE, Rseaux IP Europens”*, 2004. `http://www.ripe.int/`.

[33] *“RIPE NCC Test Traffic Measurements (TTM) Service”*, 2003. `http://www.ripe.net/ttm/`.

[34] E. Stephan. *“IPPM measurement signature”*. IETF Internet Draft (individual submission, work in progress), `draft-stephan-ippm-test-packet-header-01.txt`, October 2002.

[35] E. Stephan. *“IPPM spatial metrics measurement”*. IETF Internet Draft (individual submission, work in progress), `draft-stephan-ippm-spatial-metrics-00.txt`, September 2002.

[36] E. Stephan. *“IPPM metrics registry”*. IETF Internet Draft (work in progress) `draft-ietf-ippm-metrics-registry-04.txt`, April 2003.

[37] E. Stephan and J. Jewitt. *“IPPM reporting MIB”*. IETF Internet Draft (work in progress), `draft-ietf-ippm-reporting-mib-04.txt`, October 2003.

[38] E. Stephan and J. Palet. *“RMON Protocol Identifiers for IPv6”*. IETF Internet Draft (work in progress), `draft-ietf-rmonmib-pi-ipv6-01.txt`, November 2003.

[39] S. Waldbusser. *“Remote Network Monitoring Management Information Base”*. Internet RFC 2819 (Standards Track), IETF, May 2000.

[40] T. Zseby, M. Molina, F. Raspall, and N. Duffield. *“Sampling and Filtering Techniques for IP Packet Selection”*. IETF Internet Draft (work in progress), `draft-ietf-psamp-sample-tech-03.txt`, October 2003. expires April 2004.

[41] T. Zseby, R. Penno, N. Brownlee, and B. Claise. *“IPFIX Applicability”*. IETF Internet Draft (work in progress), `draft-ietf-ipfix-as-01.txt`, October 2003. expires April 2004.