## Tampere University of Technology DMI / Institute of Communications Engineering



02-04-02

1

## 83551 Protokollaohjelmointi/Practical Protocol Programming Major Assignment with SDT

By now, you would have learnt and implemented static service discovery mechanisms with SLP as described in RFC 2608. In the SDT part of the practical exercise, you will design and implement User Agents (UAs), Directory Agents (DAs) and Service Agents (SAs) that perform dynamic service discovery with SLP.

UAs use multicast discovery to find the DA. You may assume that the DA is persistent throughout the simulation, and the DA advertises its presence regularly. SAs must register with DA after using multicast discovery also. All agents use the default scope.

As described in RFC 2608, multicast requests should be reissued over CONFIG\_MC\_MAX seconds until a result has been obtained. UAs need only wait till they obtain the first reply which matches their request. You may change the timing defaults in SLP to suit your needs, but they must be documented. Both configurations illustrated in Page 4, Section 3 must be supported. Your final demonstration will consist of two simulations: The first with a DA present, and the next without a DA present. In both cases, the UAs must successfully find the services they are looking for without knowing the types of agents present in the network beforehand. For simplicity, you may assume that the UA performs service discovery after all the SAs have already become active and/or registered with the DA.

The additional requirements for the simulation are as follows:

- 1. UAs, SAs and DAs must be designed in separate systems. No direct communication exists between similar agents (between UAs, or between SAs). All unicast and multicast communication schemes take place with PostMaster.
- 2. At least 1 UA needs to be implemented but only 1 DA is needed. UAs perform active DA discovery. 2 SAs are also needed: 1 performing active DA discovery, the other performing passive DA discovery. SAs can either be implemented as separate blocks within the same system or as separate systems.
- 3. No UA solicitation of SA adverts must take place. All Service Request messages are multicast but they need not contain SPI, predicate or PRList fields. All SLP message headers must contain the Version, Function-ID, XID and O, F and R flags. For DA Adverts, the implementation of min-refresh-interval is not needed.
- 4. No PDU encoding necessary, and no length fields in the PDUs are necessary

## Tampere University of Technology DMI / Institute of Communications Engineering



02-04-02

2

5. As PostMaster does not really distinguish between the concept of unicast and multicast, you may wish to include an additional field in your signals to indicate the destination for unicast messages. This way, your other systems may ignore all directed signals not targeted for them.

## Bonus:

- 1. Implement the AttrReq messages and their corresponding replies.
- 2. Implement UA Scope usage and Scope Discovery (DAAdverts, SAAdverts)

The deadline for the assignment is <u>Friday 10.5.2002</u>. When you have completed your major assignment, please submit to me your SDL diagrams, MSC trace and a short design report. Submission by email as postscript files are ok, and so are printouts. (.ssy, .sdt, .spr, .sbk, .sli, .spt). Do not submit any other generated files. Please email me at bilhanan@cs.tut.fi so that we can arrange a suitable time for me to check your work. After your report is submitted, we can meet (preferably in the mornings) and run a short demo. The assignment deadline refers to your report submission, **not** the demo.

There is no strict format for your design report. However, as a guideline, the following would be useful inclusions:

How did you design your UAs, SAs and DA?

Error-handling in your systems.

Exceptional situations and biggest difficulties?

Did your implementations conform to the RFC, and if not why?

Were there any other design enhancements or features which you had to add?

You should refer to the SDT online help pages should you require any documentation.

All information about the assignment such as deadlines, updates, FAQs, etc are accessible from http://www.cs.tut.fi/~bilhanan/83551.html which I would recommend checking regularly.

Finally, you are encouraged to experiment working with SDT <u>as much as possible</u>. Should you require any additional documents or help regarding SDT, please visit my office at H212.