FFI Forsvarets forskningsinstitutt

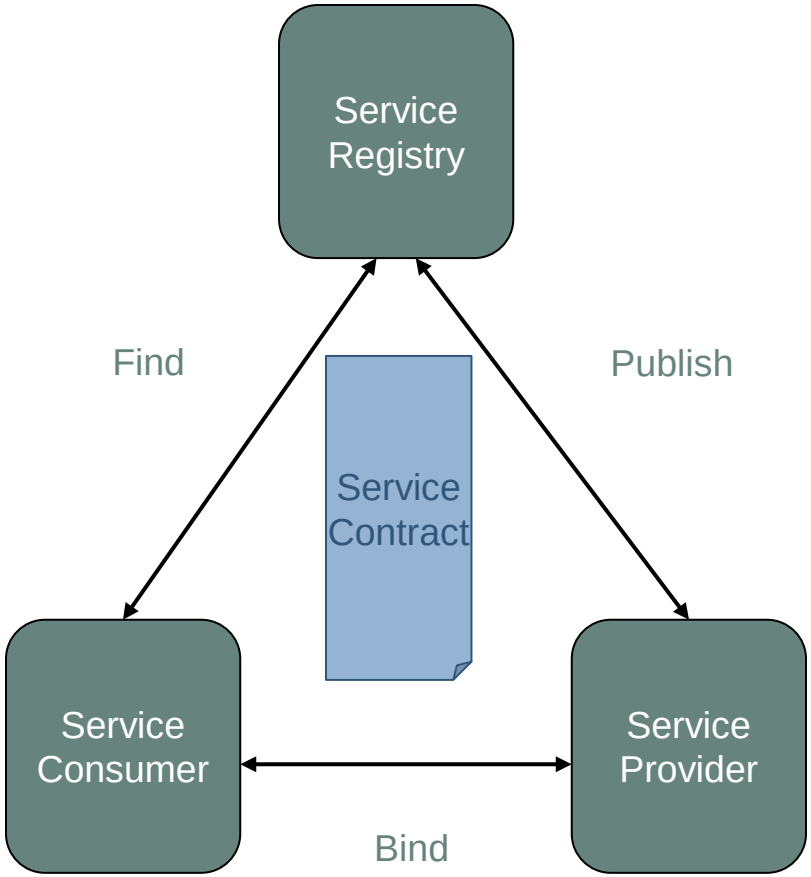
Adapting WS-Discovery for use in tactical networks

Frank T. Johnsen
Trude Hafsøe

Outline

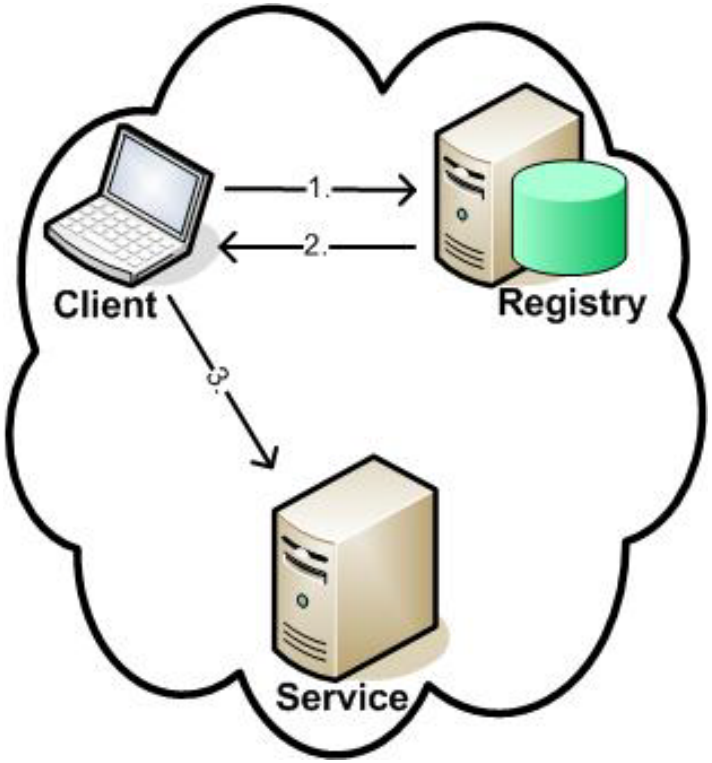
- Background
  - SOA / Web services
  - Discovery standards / drawbacks
- WS-Discovery
  - Decentralized discovery
  - Reduce overhead with emerging compression standard
    - Proof-of-concept implementation
    - Evaluation using
      - 1-100 services
      - 1-250 nodes
- Conclusion

Background



- SOA, implemented using Web services, is a key enabler for NNEC.
- The ability to find services dynamically is a requirement
  - but challenging in dynamic settings.
- At the same time, we need flexibility
  - clients must be able to access information in a manner that is suited to their needs and abilities.

Service Registries

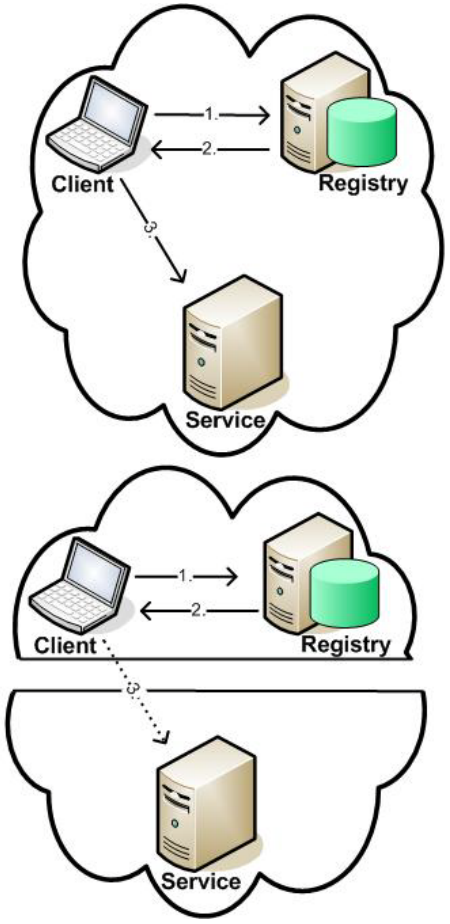


Communication between client, registry and service:

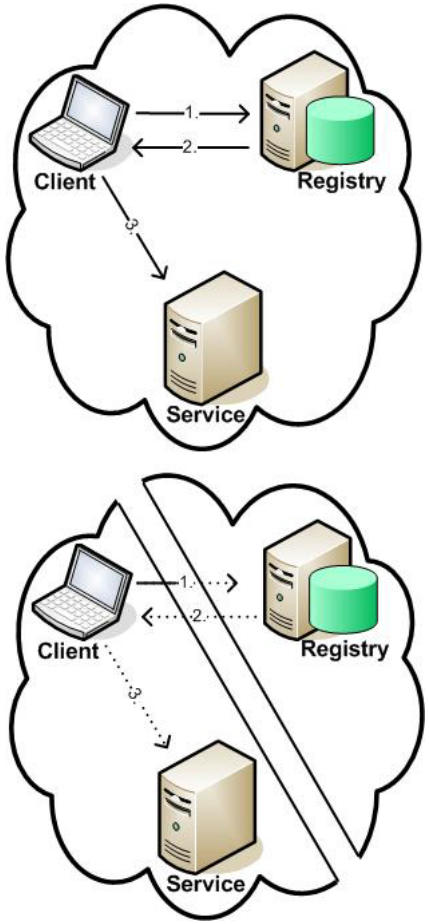
1. Look-up/Search
2. Response
3. Contacting the service

Service Registries

The liveness problem



The availability problem



Web services discovery standards

- There are three standards addressing Web services discovery:
  - Two registries, UDDI and ebXML
    - Registries suffer from liveness and availability problems in dynamic environments.
  - The third standard for Web services discovery is WS-Discovery.
    - Decentralized and better suited to dynamic networks.

WS-Discovery

- Open source:
  - http://code.google.com/p/java-ws-discovery/
- Decentralized mechanism:
  - Robustness: Resilience to partial failure of the network.
  - Liveness: An up-to-date view of available services.
- Decentralized solutions are “chatty”:
  - Need to optimize data rate requirements!
    - Compression.

An emerging XML compression standard

- The W3C has created a specification for efficient XML interchange (EXI), which reduces XML overhead by defining a binary interchange format.

- Open source:
  - http://exificient.sourceforge.net/

- We can apply EXI to SOAP-over-UDP in WS-Discovery, and still remain compliant to the standard as discussed in the SOAP Messaging Framework standard (section 4.2).

WS-Discovery

- Intended for LAN environments.
  - Generates a lot of network traffic.
- Three phases
  - Multicast Hello messages (i.e. new services published)
  - Multicast Probe messages (i.e. search network), gets unicast probe match reply (which can grow very large)
  - Multicast Bye messages (i.e. clean removal of services – *not always possible!*)

- We investigate the bandwidth requirements for WS-Discovery Hello and Probe messages
  - without compression
  - Using EXI

Test data

- 100 WSDL files (Web services interfaces) from actual services.
- Fetched from “http://www.webservicex.net/” and “http://www.webservicelist.com/”.
- Also, the WSDLs from Google and Amazon’s search services were included, providing us with a representative set of interfaces (table below, sizes in bytes).

Minimum size	Maximum size	Average	Standard deviation	Median
1643	149342	13830	19202	8514

Evaluation

- The standard requires all multicast packets (i.e., HELLO, PROBE, and BYE messages) to be sent twice, and the unicast PROBE MATCH messages to be sent once.
  - We are concerned with WS-Discovery in dynamic environments and focus on the HELLO, PROBE, and PROBE MATCH messages.
  - We use a small network to capture actual data, then an analytical evaluation using the results.

WS-Discovery resource use

- WS-Discovery is based on a query-response model, where a multicast query (probe) triggers unicast responses (probe match).

- The load incurred on the network by the number of querying nodes (q) in a network with a total number of n nodes can be calculated using this formula:
  - LOAD = (sizeof(probe) + sizeof(probe match) * (n − 1)) * q
    - If all nodes should have an up-to-date view of the currently available services, then q = n.
    - If only one node is querying, then q = 1.

The HELLO message sizes (in bytes) vary with different WSDLs

Compression	Minimum size	Maximum size	Average	Standard deviation	Median
Uncompressed	807	887	834	17,04	830
EXI	373	420	390	9,22	388

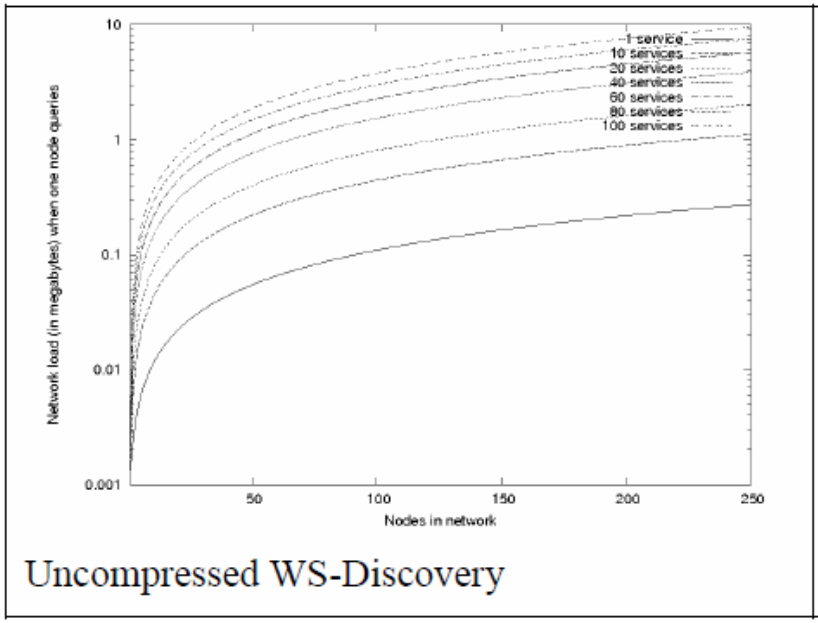
PROBE (sizes in bytes)

- PROBE
  - Uncompressed always 581 (using a generic probe querying for all available services with no scope limitations).
  - An EXI compressed PROBE message varied between 272 and 274 (compression varying with varying UUID and time stamp in message)

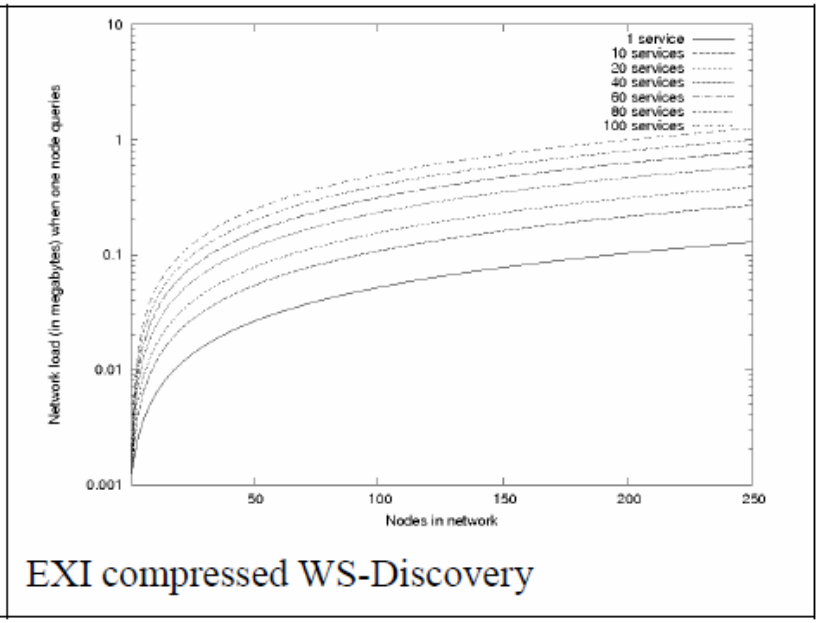
PROBE MATCH (sizes in bytes)

Number of services	Uncompressed PROBE MATCH	EXI compressed PROBE MATCH
100	38200	5009
80	30292	4000
60	22902	3146
40	15531	2339
20	8122	1552
10	4476	1072
1	1092	511

WS-Discovery's resource use when ***one*** node queries (q = 1)

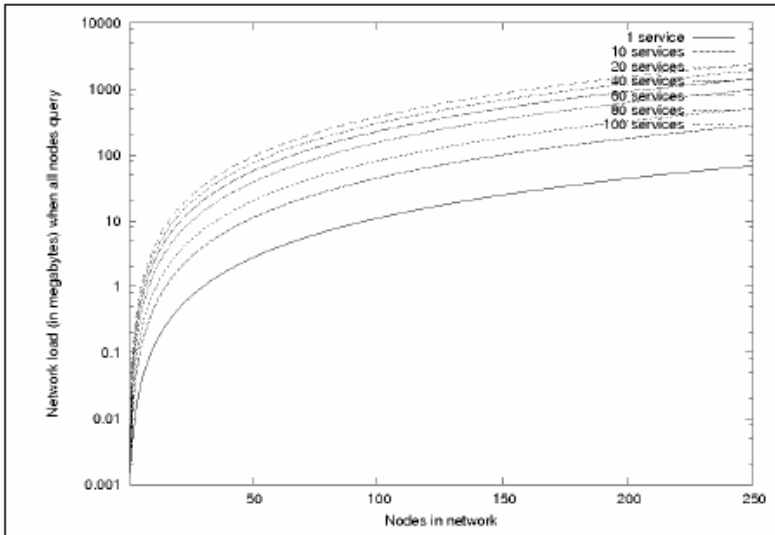


Uncompressed WS-Discovery

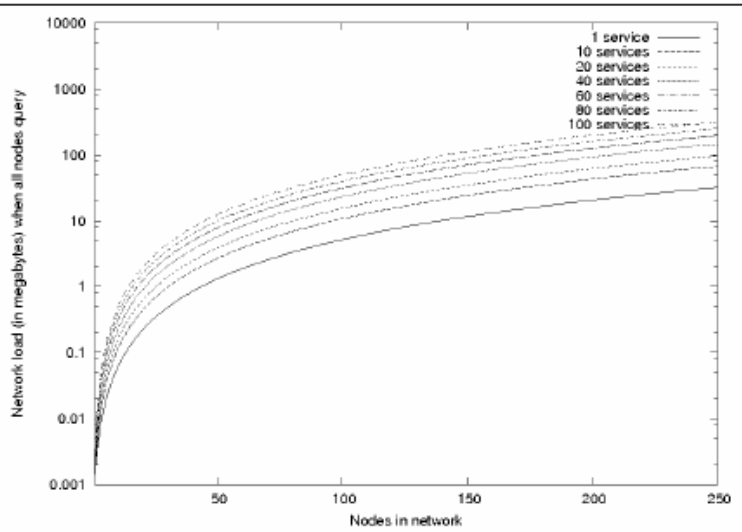


EXI compressed WS-Discovery

WS-Discovery's resource use when ***all*** nodes query (q = n)



Uncompressed WS-Discovery



EXI compressed WS-Discovery

Conclusion

- Standards are preferable to proprietary solutions because they ease interoperability and reduce the chances of vendor lock-in.
  - Though WS-Discovery in our previous research has proven itself to be less than optimal for use in tactical networks, its resource use is significantly reduced when coupled with EXI as our results show.

- WS-Discovery could well be of value in, e.g., a civil-military operation, where it could provide a standardized Web services discovery capability for both the civil and the military dynamic networks.

Thank you for your attention!

- Questions?

SOAP Messaging Framework standard, section 4.2

- “SOAP Message Construct provides that all SOAP envelopes are serializable using an XML 1.0 serialization, so XML 1.0 or later versions of XML MAY be used by bindings as the "on the wire" representation of the XML Infoset. However, the binding framework does not require that every binding use an XML serialization for transmission; compressed, encrypted, fragmented representations and so on can be used if appropriate.”