The Negotiation Problem

Bryan Ford
Yale University
bryan.ford@yale.edu

Jana Iyengar
Franklin & Marshall College
jiyengar@fandm.edu

Presentation for TSVAREA meeting – Nov 13, 2009
http://baford.info/tng
A Proliferation of Layers and Layer Combinations

Application:
- HTTP
- FTP
- DNS
- RTP

Transport Security:
- SSL
- DTLS

Transport:
- SCTP
- TCP
- UDP
- DCCP

Network Security:
- IPsec
- IPv6

Network:
- IP
- IPv6

Data Link:
- Ethernet
- Token-Ring
- PPP

(DirectAccess)
Future: Ever More Layers/Combinations?

Multi-Streaming Transports
SCTP [rfc4960], SST [SIGCOMM'07]

Multipath Transports
SCTP [rfc4960], MPTCP [WIP]

Further Decomposition
[“Breaking Up the Transport Logjam”, HotNets’08]
The Negotiation Problem

Decisions, decisions!

Application

Transport

Security

Transport

Network

HTTP

SSL

TCP

SCTP

IPv4

IPv6

SIP

IAX

UDP

DCCP

IPv4

IPv6
Compatibility and Preference

Which combinations do both endpoints support?

Which combinations do they prefer?

Host A

Host B
Talk Outline

- Three negotiation strategies (2 explicit, 1 implicit)
  - Including a new in-band negotiation mechanism
  - Combined explicit/implicit negotiation
- A framework for negotiation
- Discussion
Negotiation Strategies

Implicit Negotiation
Approach 1: Try and Fall Back

Host A

SCTP INIT

TCP INIT

Host B

SCTP RST

TCP ACK
Challenge 1: Controlling Delay

- Failures can incur *timeouts* (e.g., due to NATs)
- ... potentially *compounded* by layering

---

**Diagram:**

```
Host A: IPv4, IPv6, UDP, DCCP, DTLS, SIP, IAX

Host B: IPv4, IPv6, UDP, DCCP, DTLS, SIP, IAX

---

Timeout(s)
```
Approach 2: Try in Parallel

Host A

SCTP INIT

TCP INIT

Host B

SCTP RST

TCP ACK
Challenge 2a: Redundant State

Host A

SCTP INIT

TCP INIT

Host B

SCTP ACK

TCP ACK
Challenge 2b: Combinations

Layering can lead to explosion of choices
Negotiation Strategies

- Implicit Negotiation
- Explicit Out-of-band Negotiation
Approach 3: Out-of-Band Information

Host A → DNS Server → Host B

DNS++ Req → DNS Server → DNS++ Reply

SIP, DTLS, DCCP, IPv4, IPv6
Challenge 3a: Administration

DNS server must know:

- Name → IP mapping (as before)
- Entire protocol stack supported by Host B
- Protocol options?

⇒ Synchronization Nightmare?
Challenge 3b: E2E Robustness

If endpoints agree on configuration X, will it work?

IPv4
IPv6
UDP
DCCP
DTLS
SIP
IAX

Host A

Middlebox

Host B
Negotiation Strategies

- Implicit Negotiation
- Explicit Out-of-band Negotiation
- Explicit In-band Negotiation
Approach 4: In-band Negotiation

- Hosts explicitly describe possible configurations during initial “meta-communication” exchange, before actual communication commences.
Message 1: Initiator → Responder: Propose Protocol Graph

Negotiation Message 1

- goal (SIP)
  - opt1
  - opt2

- (alternatives)

- TLS
  - opt1
  - opt2

- DTLS
  - opt1
  - opt2

- TCP
  - opt1
  - opt2

- DCCP
  - opt1
  - opt2

- base (IP)

Host A

Host B
Message 2: Responder → Initiator:
Revise Protocol Graph
Message 3: Initiator → Responder:

Acknowledge Protocol Graph

Host A

Target Message 3

goal (SIP)

opt2

TLS

opt1

TCP

base (IP)

Host B
Message 4+:
According to Negotiated Stack

Host A

Normal Packets

SIP
TLS
TCP

Host B
Concurrent Protocol Initialization

Whenever feasible:

- *embed* protocol-specific handshake info into graph
- *run handshakes concurrently* while negotiating
Key Benefits of Negotiation Model

- Happens strictly between nodes concerned
  - Users, Name server admins don't have to care
- Middleboxes can participate in process
- Protocol graph representation scales to handle:
  - Arbitrarily deep protocol stacks
  - Many alternatives per layer
- Setup whole “layer cakes” in minimal # of RTTs
  - With options

(For representing and transmitting graph, negotiation transport protocol, etc., see our HotNets '09 paper)
**Contexts and Stacks**

- **Context** $\equiv$ underlying substrate; *cannot change*
- **Stack** $\equiv$ protocols to be set up; *can change*

**Example 1:** Application-Level VoIP Protocol Stack Negotiation

- **Stack**
  - SIP
  - IAX
  - DTLS
  - UDP
  - DCCP
- **Context**
  - IPv4
  - IPv6

**Example 2:** OS-Level, Application-Transparent Transport Stack Negotiation

- **Stack**
  - HTTP
  - TCP
  - SST
- **Context**
  - IPv4
  - IPv6
Scenario 1: Application-Level VoIP Protocol Stack Negotiation

- SIP
- IAX
- DTLS
- UDP
- DCCP

Stack:
- IPv4
- IPv6

Context:
- App can't send 1 packet that's both UDP & DCCP!

Scenario 2: Application-Transparent Transport Protocol Negotiation

- HTTP
- TCP
- SST
- DCCP

Stack:
- IPv4
- IPv6

Context:
- OS can't send 1 packet that's both IPv4 & IPv6!

⇒ must try each context separately
Combined Solution

1. Identify feasible communication Context(s)
   - e.g., UDP session (IP_a:port_a, IP_b:port_b)

2. Negotiate Stack within each context:
   a) Initiator sends a Protocol Graph Proposal
   b) Responder returns Revised Protocol Graph
   c) (Optional) further protocol graph revision steps
   d) Peers commit, Acknowledge Protocol Graph
   e) Communication proceeds via negotiated protocols
Combined Implicit/Explicit Solution

- Implicit, parallel negotiation *across contexts*
- Explicit, in-band negotiation *within a context*
A Framework for Negotiation
Negotiation Strategies

- Implicit Negotiation
- Explicit Out-of-band Negotiation
- Explicit In-band Negotiation
The Negotiation Triangle

- Implicit Negotiation
- Explicit Negotiation
- Multi-Context Support
- End-to-end Robustness
- Explicit Out-of-band Negotiation
- Explicit In-band Negotiation
- Combinatorial Scalability
For any given negotiation strategy, you get two of three desirable properties.

To get all three properties, a hybrid of at least two strategies is necessary.
Arigato!

The floodgates are open!

(Please join tae@ietf.org for discussions)