About SIP

SIP Routing

Jonas Borjesson

NOTE

- This version has been adapted to be viewed without transitions.
- Go to **aboutsip.com** to download the original version.
- Also be sure to check out **vimeo.com/aboutsip** for any recorded presentations.
- Follow **@borjessonjonas** to receive updates.

Routing

- Understanding SIP routing is a must!
- SIP's flexible routing is what makes SIP so great*.
- The routing capabilities of SIP enables:
  - Loosely coupled systems
  - Ability for a very flexible application composition model
  - NAT/FW Traversal
  - CONFUSION!

*According to myself. Many people may disagree...

Questions

- How does a SIP request traverse the network?
- How do we know which transport protocol to use?
- How do the responses find their way back?
- Any difference for in-dialog requests?

What do we need?

- In order to send a request we need to know:
  - The IP-address of the destination (or the next hop)
  - The port to send it to.
  - Which transport to use (udp, tcp, tls or sctp?)
- So, how do we do this?

Locating SIP Servers

- RFC 3263 has all the answers.
- Makes use of DNS
  - NAPTR lookup to find transport
  - SRV lookup to find the port
  - A-record lookup to find the IP-address

Find the transport

- If transport is specified, use it

  sip:alice@aboutsip.com;transport=udp

- If target is numeric IP, use UDP

  sip:alice@192.168.0.100

- If no transport and target is not numeric but port is specified, use UDP*

  sip:alice@aboutsip.com:5090

- If none of the above, do a NAPTR lookup on target

```
IN   NAPTR  10 10 "S" "SIPS+D2T" "" _sips._tcp.aboutsip.com
IN   NAPTR  20 10 "S" "SIP+D2T" "" _sip._tcp.aboutsip.com
IN   NAPTR  30 10 "S" "SIP+D2U" "" _sip._udp.aboutsip.com
```

* but you may use another transport if necessary (e.g. msg > MTU)

Find the port

- If port specified, use it.

  sip:alice@aboutsip.com:5070

- If target is a numeric IP address and no port specified, use default for the selected transport.

  sip:alice@192.168.0.100;transport=udp => 5060
  sip:alice@192.168.0.100;transport=tcp => 5060
  sip:alice@192.168.0.100;transport=tls => 5061

- Otherwise, perform a SRV query

  _sip._udp.aboutsip.com 1800 IN SRV 10 10 5060 lb1.aboutsip.com
  _sip._udp.aboutsip.com 1800 IN SRV 10 10 5060 lb2.aboutsip.com

Find the IP Address

- If numeric IP address, use it.

  sip:alice@192.168.0.100

- If SRV record lookup was performed, perform A record lookup based on that result.

  lb1.aboutsip.com

- Otherwise, do a A record lookup based on the domain in the SIP URI.

  sip:alice@aboutsip.com

- If many IP's are returned, try them top down.

  lb1.aboutsip.com. 1800 IN A 10.36.10.10
  lb1.aboutsip.com. 1800 IN A 10.36.10.11

Which URI?

```
INVITE sip:alice@aboutsip.com SIP/2.0
To: “Alice” <sip:alice@aboutsip.com>
From: “Bob” <sip:bob@aboutsip.com>;tag=oiu3rlkj
...
Contact: <192.168.0.100;transport=tcp>
```

```
INVITE sip:alice@aboutsip.com SIP/2.0
To: “Alice” <sip:alice@aboutsip.com>
From: “Bob” <sip:bob@aboutsip.com>;tag=oiu3rlkj
...
Contact: <192.168.0.100;transport=tcp>
Route: <sip:outbound.aboutsip.com;lr>
Route: <sip:something.else.com;lr>
```

```
INVITE sip:alice@aboutsip.com SIP/2.0
To: “Alice” <sip:alice@aboutsip.com>
From: “Bob” <sip:bob@aboutsip.com>;tag=oiu3rlkj
...
Contact: <192.168.0.100;transport=udp>
Route: <sip:outbound.aboutsip.com>
Route: <sip:something.else.com;lr>
```

@borjessonjonas

For Requests

- If no Route-headers:
  use Request URI
- If Route-headers:
  use the top most Route-header
- BUT
  - Is “lr” parameter present or not?
  - Strict routing vs loose routing
  - RFC 2543 vs RFC 3261

Hello Bob

sip:bob@10.36.10.11;transport=udp

```
INVITE sip:bob@aboutsip.com SIP/2.0
To: <sip:bob@aboutsip.com>
From: <sip:alice@sipflow.io>;tag=oiu3rlkj
Call-Id: asik3-afj3-knoiu2lkj
CSeq: 1 INVITE
Max-Forwards: 70
Contact: <192.168.0.100;transport=tcp>
Route: <sip:outbound.sipflow.io;lr>

Via: SIP/2.0/TCP 62.4.10.10;branch=z9hG4bK-32BDjdfd
Via: SIP/2.0/TCP 192.168.0.100;branch=z9hG4bK-kljhasdf
Via: SIP/2.0/UDP 82.67.45.50;branch=z9hG4bK-dk3imiuj3
```

@borjessonjonas

Hello Back

outbound.sipflow.io

aboutsip.com

```
SIP/2.0 200 OK
To: <sip:bob@aboutsip.com>;tag=jalskdjfi
From: <sip:alice@sipflow.io>;tag=oiu3rlkj
Call-Id: asik3-afj3-knoiu2lkj
CSeq: 1 INVITE
Contact: <sip:10.36.10.11;transport=udp>
Via: SIP/2.0/UDP 82.67.45.50;branch=z9hG4bK-dk3imiuj3
Via: SIP/2.0/TCP 62.4.10.10;branch=z9hG4bK-32BDjdfd
Via: SIP/2.0/TCP 192.168.0.100;branch=z9hG4bK-kljhasdf
```

Via-headers

A closer look


```
Via: SIP/2.0/UDP 82.67.45.50;rport=5060;received=10.36.10.0.1
Via: SIP/2.0/TCP 62.4.10.10;rport=87564;received=62.4.10.10
Via: SIP/2.0/TCP 192.168.0.100;rport =5099;received=192.168.0.100
```

Subsequent Requests

- During the dialog initiation, a route-set is built (may be empty).
- The route-set is part of the dialog-state and must be preserved.
- Future in-dialog requests will follow that established route.
- The actual request itself is no different from a out-of-dialog request. It will follow the route-headers + request-uri...
-

Establishing the Route Set


```
INVITE sip:bob@aboutsip.com SIP/2.0
To: <sip:bob@aboutsip.com>
From: <sip:alice@sipflow.io>;tag=oiu3rlkj
Call-Id: asik3-afj3-knoiu2lkj
CSeq: 1 INVITE
Max-Forwards:
Contact: <192.168.0.100;transport=tcp>
Route: <sip:outbound.sipflow.io;lr>

Record-Route: <sip:82.67.45.50;transport=tcp;lr>

Record-Route: <sip:62.4.10.10;transport=tcp;lr>
```

Establishing the Route Set

outbound.sipflow.io

aboutsip.com

```
SIP/2.0 200 OK
To: <sip:bob@aboutsip.com>;tag=klajsdf
From: <sip:alice@sipflow.io>;tag=oiu3rlkj
Call-Id: asik3-afj3-knoiu2lkj
CSeq: 1 INVITE
Contact: <sip:10.36.10.11;transport=udp>

Via: ...

Record-Route: <sip:82.67.45.50;transport=tcp;lr>
Record-Route: <sip:62.4.10.10;transport=tcp;lr>
```

Dialog State

```
Remote Target: sip:10.36.10.11;transport=udp

Route Set:     sip:62.4.10.10;transport=tcp;lr
               sip:82.67.45.50;transport=tcp;lr
```

```
Remote Target: sip:192.168.0.100;transport=tcp

Route Set:     sip:82.67.45.50;transport=tcp;lr
               sip:62.4.10.10;transport=tcp;lr
```

Note: There is more information stored in the dialog state. This example only highlights the routing information.

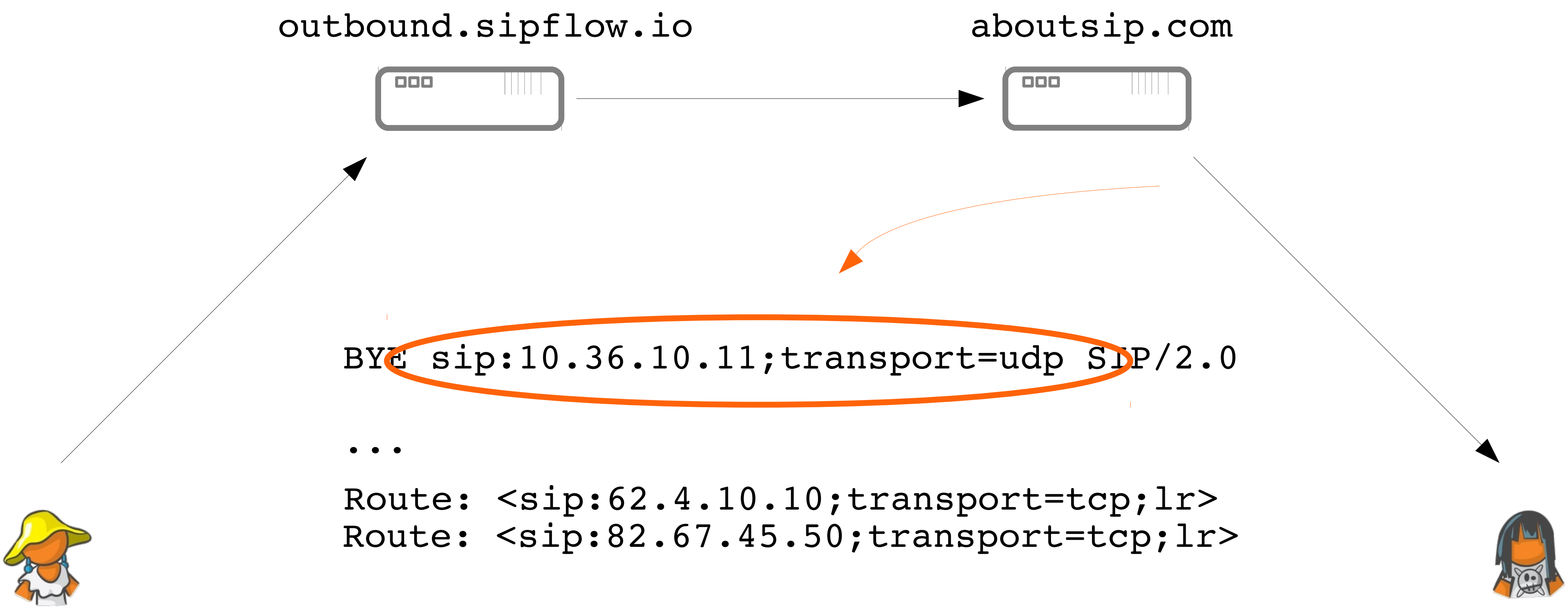
Constructing a Subsequent Request

- Remote target → Request URI
- Route set → Route Headers
- Send like any other request

```
Remote Target:  sip:10.36.10.11;transport=udp

Route Set:
                sip:62.4.10.10;transport=tcp;lr
                sip:82.67.45.50;transport=tcp;lr
```

```
BYE sip:10.36.10.11;transport=udp SIP/2.0

...

Route: sip:62.4.10.10;transport=tcp;lr
Route: sip:82.67.45.50;transport=tcp;lr
```

Sending


outbound.sipflow.io
aboutsip.com
BYE sip:10.36.10.11;transport=udp SIP/2.0
...
Route: <sip:62.4.10.10;transport=tcp;lr>
Route: <sip:82.67.45.50;transport=tcp;lr>

Subsequent Requests

Check this out

outbound.sipflow.io

aboutsip.com

Not Record-Routing!

```
INVITE sip:bob@aboutsip.com SIP/2.0
To: <sip:bob@aboutsip.com>
From: <sip:alice@sipflow.io>;tag=oiu3rlkj
Call-Id: asik3-afj3-knoiu2lkj
CSeq: 1 INVITE
Max-Forwards:
Contact: <192.168.0.100;transport=tcp>
Route: <sip:outbound.sipflow.io;lr>

Record-Route: <sip:62.4.10.10;transport=tcp;lr>
```

Dialog State

Remote Target: sip:10.36.10.11;transport=udp

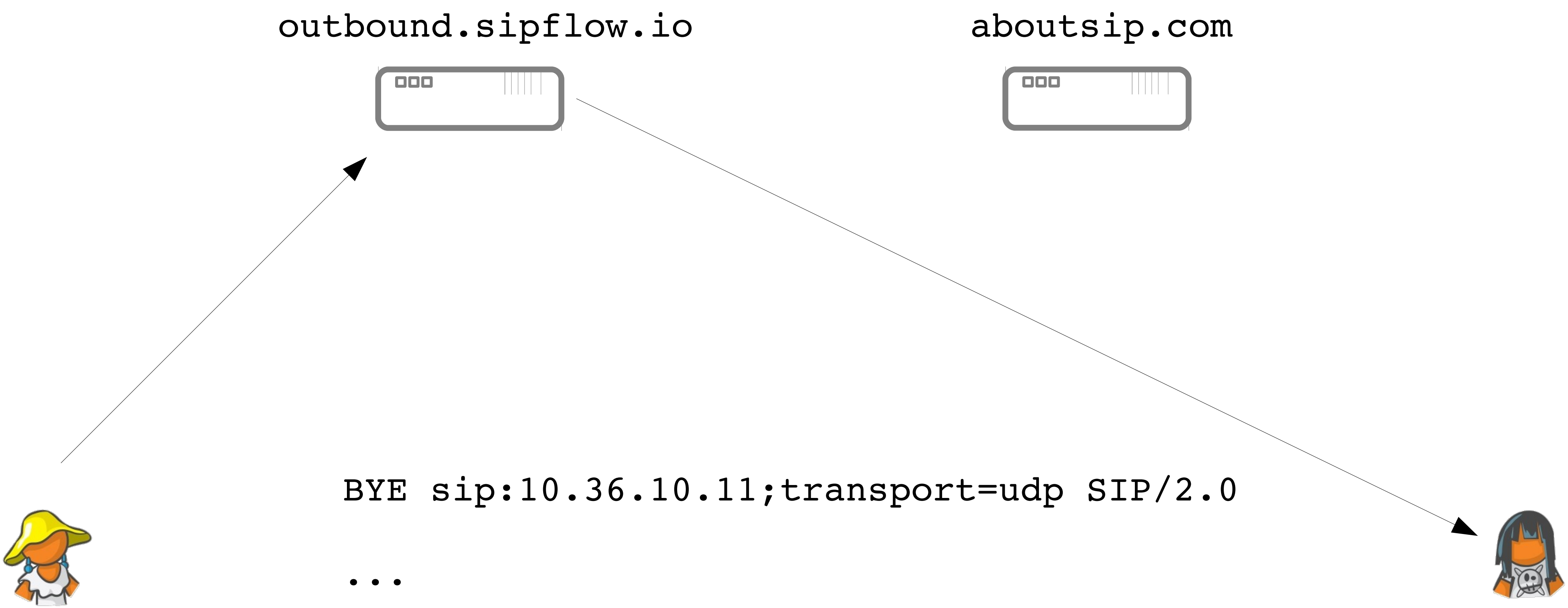
Route Set: sip:62.4.10.10;transport=tcp;lr
~~sip:82.67.45.50;transport=tcp;lr~~

Remote Target: sip:192.168.0.100;transport=tcp

Route Set: ~~sip:82.67.45.50;transport=tcp;lr~~
sip:62.4.10.10;transport=tcp;lr

Note: There is more information stored in the dialog state. This example only highlights the routing information.

Sending




```
BYE sip:10.36.10.11;transport=udp SIP/2.0

...

Route: <sip:62.4.10.10;transport=tcp;lr>
```

Summary

Summary

- RFC 3263 is important! Read it!
- RFC 3261 explains which the “next uri” is:
  - Requests – follows Route-headers plus Request-URI
  - Responses – follows Via-headers
  - Watch out for the old strict routing.
- Subsequent requests are really no different.
- Make sure you understand remote-target + route set as stored in the Dialog State.

More presentations and material
at aboutsip.com

Thanks!