Real-Time Communication in Web-browsers (RTCWeb)

Michael Tüxen

(tuexen@fh-muenster.de)

Outline

- RTCWeb overview
- Peer to peer protocol stack
- RTCWeb datachannels
- SCTP introduction
- Live demo

Motivation

- Multiple solutions for media and non-media peer to peer communication:
  - Skype (Microsoft)
  - Google Hangouts (Google)
  - Facetime (Apple)
  - Adobe Connect (Adobe)
  - WebEx (Cisco)
- Limitations:
  - Require proprietary software
  - Not interoperable

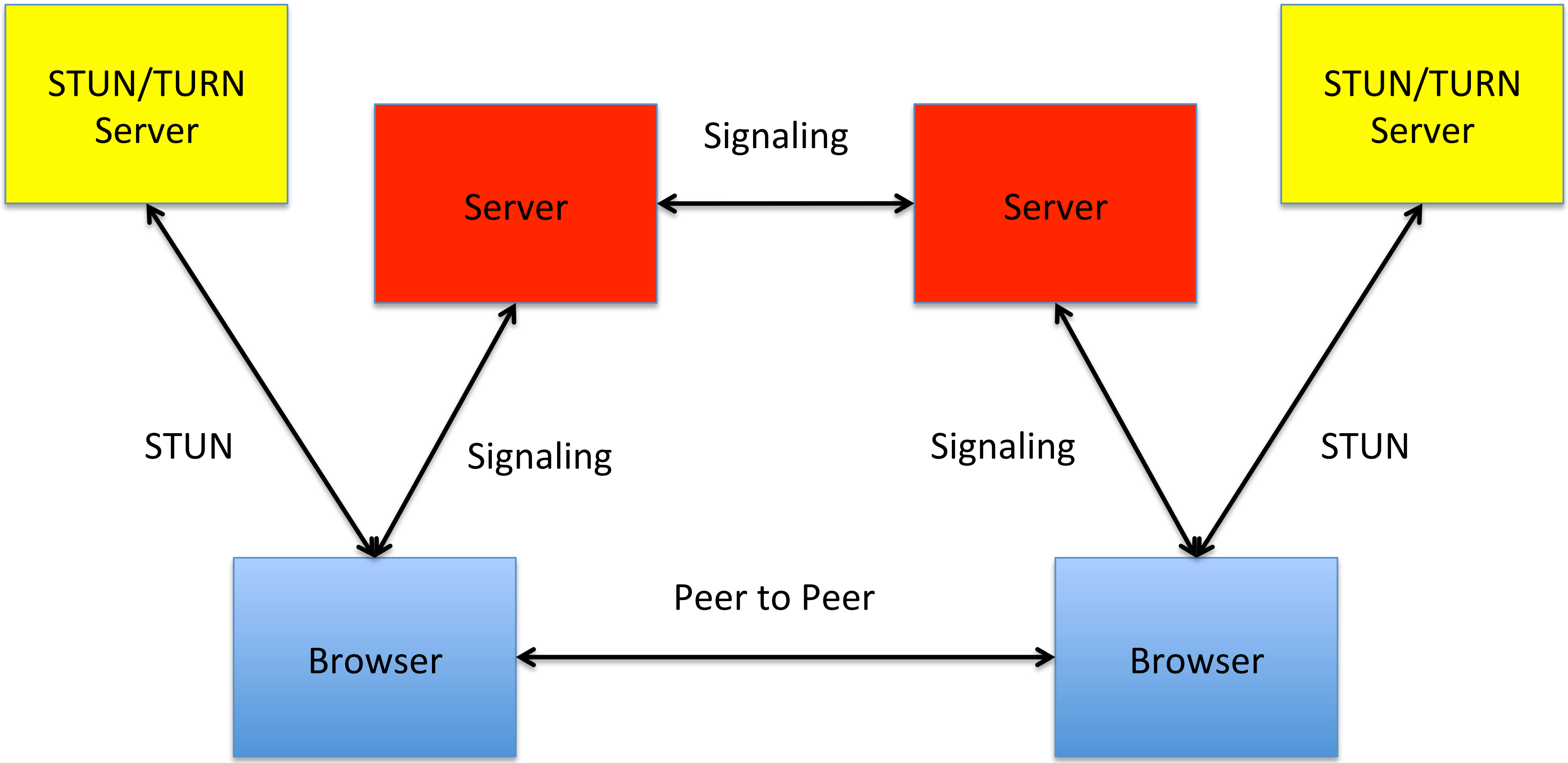
RTCWeb

- Joint activity between
  - the Real-Time Communications in Web Browsers (RTCWeb) Working Group of the Internet Engineering Task Force (IETF) defining the protocols.
  - The Web Real-Time Communication (WebRTC) Working Group of the World Wide Web Consortium (W3C) defining the Javascript API.

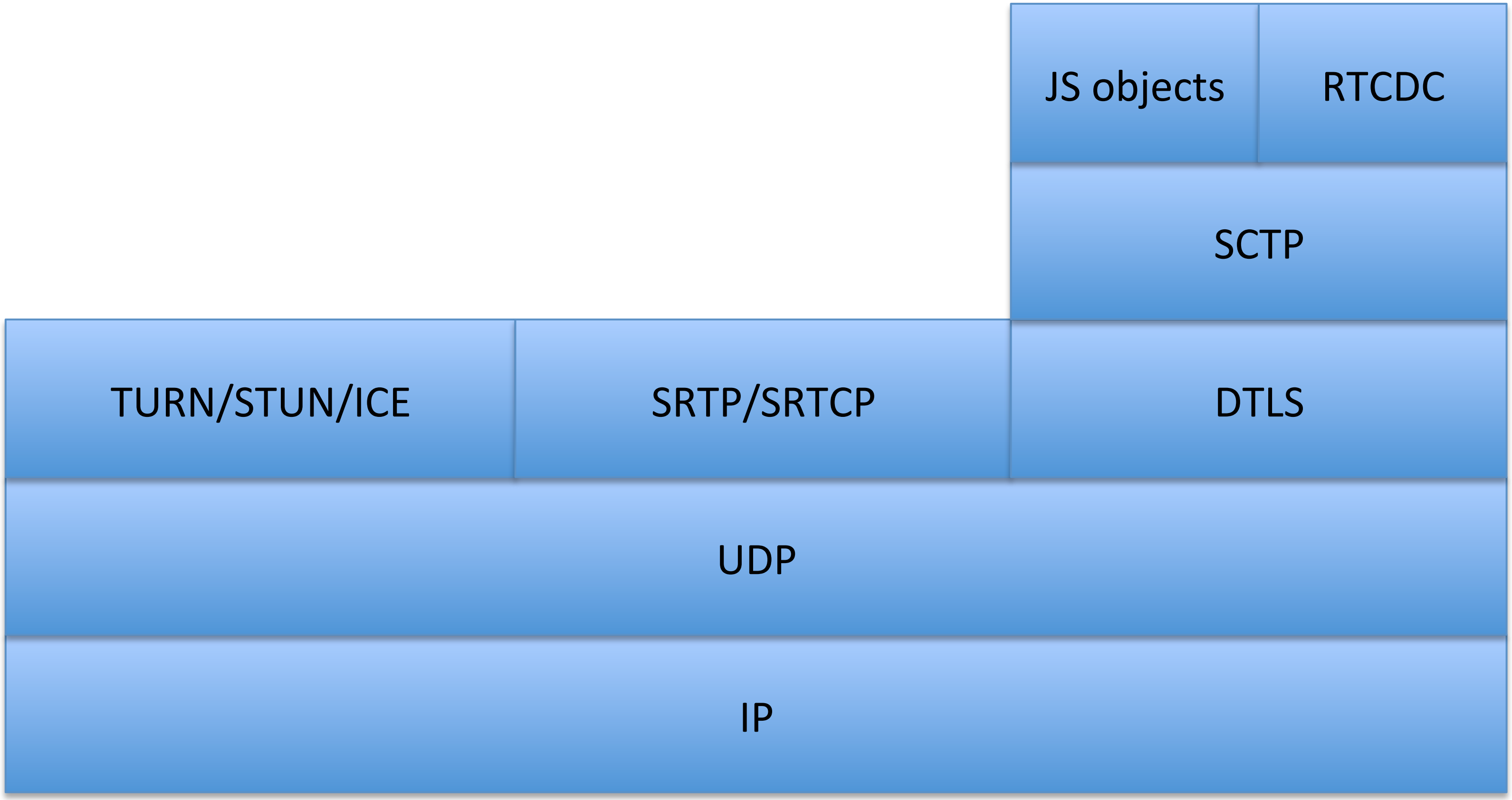
Current Status

- Implemented in
  - Firefox
  - Chrome
- Successful interoperability test for voice/video call at the beginning of February 2013.
- Protocol issues mostly resolved.
- Mandatory to implement audio codec: Opus and G.711.
- Still open
  - Mandatory to implement video codec: H.264 or V8?
  - Congestion control for media channel and handling of media and non-media channel.
  - SDP related things.
  - Many details.

Architecture



Peer to Peer Protocol Stack



Data Channels

- Bidirectional
- Message oriented
- Priorities
- Reliability
  - Fully reliable
  - Partial reliable by limiting the number of retransmissions.
  - Partial reliable by limiting the life-time.
- Congestion controlled

Timeline of SCTP RFCs

- Core Protocol
  - Initial Base Specification (RFC 2960, October 2000)
  - Checksum Change (RFC 3309, September 2002)
  - Errata and Issues (RFC 4460, April 2006)
  - Updated Base Specification (RFC 4960, September 2007)
- Protocol Extensions
  - Partial Reliability (RFC 3758, May 2004)
  - Chunk Authentication (RFC 4895, August 2007)
  - Address Reconfiguration (RFC 5061, September 2007)
  - Stream Reconfiguration (RFC 6525, February 2012)
- API
  - Socket API (RFC 6458, December 2011)

SCTP Protocol Overview

- Connection oriented (SCTP association)
- Supports unicast
- Same port number concept as other transport protocols
- Message oriented
  - Supports arbitrary large messages (fragmentation and reassembly)
  - Supports bundling of multiple small messages in one SCTP packet
  - Flexible ordering and reliability
- Supports multihoming using IPv4 and IPv6
- Packet consists of a common header followed by chunks
- Extendable

Association Setup

- Four way handshake
- Resistance against “SYN flooding”
- Negotiates
  - Initial number of streams
  - Initial set of IP addresses
  - Supported extensions
- User messages can already be transmitted on the third leg (after one RTT i.e. same as TCP)
- Handles the case of both sides initiating the association.

Data Transfer

- TCP friendly congestion control
- User messages are put into DATA chunks (possibly multiple in case of fragmentation)
- Each DATA chunk is identified by a Transmission Sequence Number (TSN)
- Acknowledgements (SACKs) reporting
  - Cumulative TSN
  - Gaps (up to approximately 300 in a sack)
  - Duplicate TSNs
- Retransmissions
  - Based on timer
  - Based on gap reports

Association Teardown

- Graceful shutdown
  - Teardown without message loss.
  - Based on an exchange of three messages.
  - Supervised by timer
  - No half close state is allowed
- Non-graceful shutdown
  - Possibly message loss
  - Uses a single message

Service: Preservation of Message Boundaries

- Most application protocols are message based
- Simplifies application protocols and its implementation
- Awareness of message boundaries makes optimal handling at the transport layer / application layer boundary possible
- But special attention is needed for supporting arbitrary large messages

Service: Partial Reliability

- Allows to avoid spending resources on user messages not being relevant anymore for the receiver.
- The sender can abandon user messages base on criteria called PR-SCTP policy
- PR-SCTP policies are implemented on the sender side and does not require negotiation.
- Examples of PR-SCTP policies:
  – Lifetime
  – Number of retransmissions
  – Priority with respect to buffering

Service: Partial Ordering

- An SCTP association provides up to 2^16 uni-directional streams in each direction.
- The application is free to send a message on a stream of its choice.
- Minimizes head of line blocking, because message ordering is only preserved within each stream.
- In addition, messages can be marked for unordered delivery.
- The stream reconfiguration extension (RFC 6525) allows to
  - Add streams during the lifetime of an association
  - Reset streams (i.e. start over at stream sequence 0)

Service: Network Fault Tolerance

- Each end-point can have multiple IP-addresses
- Each path is continuously supervised
- Primary path is used for initial transmission of user data
- In case of a failure, another (working) address is used
- The Address Reconfiguration extension (RFC 5061) allows
  - Add and delete IP-addresses during the lifetime of an association
  - Select the local and remote primary path
- Currently being specified: loadsharing

SCTP Implementations

- Provided by OS vendor for
  - FreeBSD
  - Linux
  - Solaris
- The FreeBSD has been ported to support
  - Mac OS X as a network kernel extension (NKE)
  - Windows as a kernel driver
  - Windows, Linux, FreeBSD, MacOS X as a userland stack (included in Firefox)
- Commercial implementations for various operating systems
- Implementations are interoperable as shown in nine interoperability tests.

SCTP in the RTCWeb Context

- Lower layer is not IP, but DTLS, which is connection oriented.
- Kernel implementations not usable.
- Multihoming and dynamic address reconfiguration is not used.
- Support for multiplexing user messages from different streams required.
- Usage of an appropriate congestion control is required.

Management of Data Channels

- Opening can be in-band (using a simple protocol) or out-of-band (using SDP).
- In-band data channel opening:
  – Old version used a three way handshake for setup.
  – New version uses a single message for setup. Glare handled by using DTLS roles.
- Data channels are closed by resetting the corresponding SCTP streams.
- In case of SCTP stream shortage, the number of streams is increased, if possible.

Live Demonstration

- For Firefox Nightly set two environment variables:
  - NSPR_LOG_MODULES to SCTP:5,DataChannel:5
  - NSPR_LOG_FILE to /Users/tuexen/logfile
- To extract SCTP packet information use
- `grep SCTP_PACKET logfile > sctp.log`
- `Text2pcap -n -l 248 -D -t '%H:%M:%S.'\ sctp.log sctp.pcapng`
- The .pcapng file can be read by Wireshark.
- You need a recent version of Wireshark…

Conclusion

- RTCWeb
  - is a major effort.
  - might have a huge impact.
  - still requires research efforts.
  - provides peer to peer networking support to web browsers.
  - allows new classes of web based applications
  - can be used for development and research right now…