Stream Switching Control

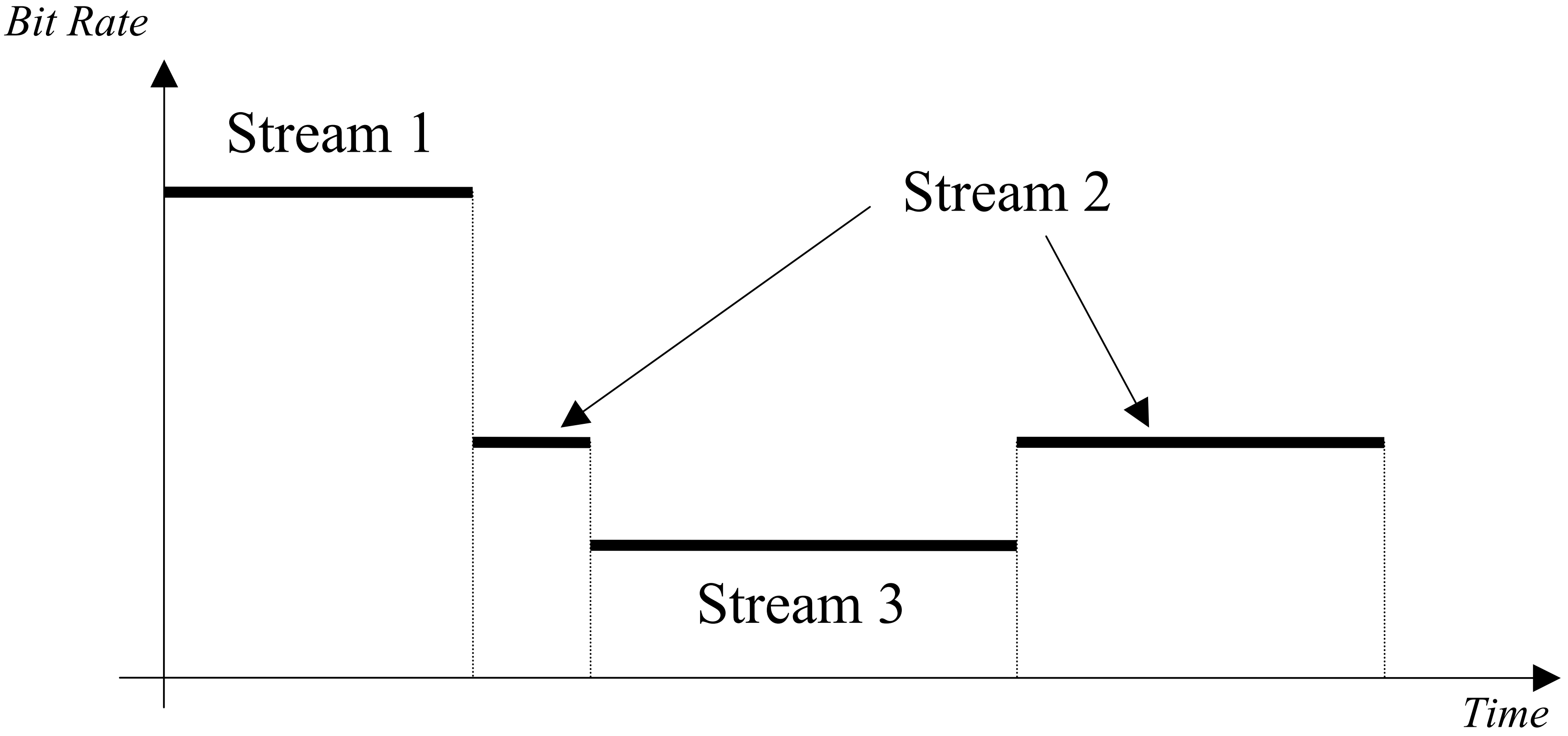
draft-gentric-mmusic-stream-switching-00.txt

Philippe Gentric
philippe.gentric@philips.com

Philips Electronics
MP4Net
http://www.platform4.com

Stream switching principle

Same ***content*** available as several ***streams*** at different bit rates



Stream Switching Control is the final building block for …

+ Streaming e.g. RTP + RTSP

+ Session negociation e.g. SDP, SMIL

+ Rate control e.g. TFRC (RFC3448)

+ Stream Switching Control (this)

= **Streaming with Congestion & Rate Control**

Stream switching in a nutshell

- Rate control:
  - ***re-use existing specification/work***
  - e.g. TFRC (RFC3448) or derived work
- Session description/content negociation e.g.
  - ***re-use existing specification/work***
  - e.g. SDP + BW modifiers, Offer/Answer, S4-020407 etc
- Switch control
  - Can use RTCP feedback
  - Can be limited to the « client-transparent » mode
  - Or requires ***explicit prior negociated agreement***
  - Cannot use RTSP PLAY/PAUSE « as is », an extension is needed
  - Server must be able to deny
  - Server should be able to « drive » (in some cases)
  - Decoding ressources should be under player control (in some cases)
  - The solution needs to cover all cases for best server/client interoperability

Stream switching control key issues

- Sender needs to receive information: what type of information can it be ?
  1. About congestion (can be RTCP, RTCP-X, could be TRIGTRAN)*
  2. About a target bit rate (player makes RC computation) **
  3. About a target stream (player does it all) **
- The whole process has to be fast and efficient
  - Sender needs to switch as soon as congestion is reported
  - Client may need/prefer RAP
  - Client may need/prefer forewarning
  - Processing ressources have to be ready on ***both*** sides
  - But this can be very wasteful
- In the general case switches are ***not*** client-transparent

* *: often described as « server initiated »*

***: often described as « player initiated »*

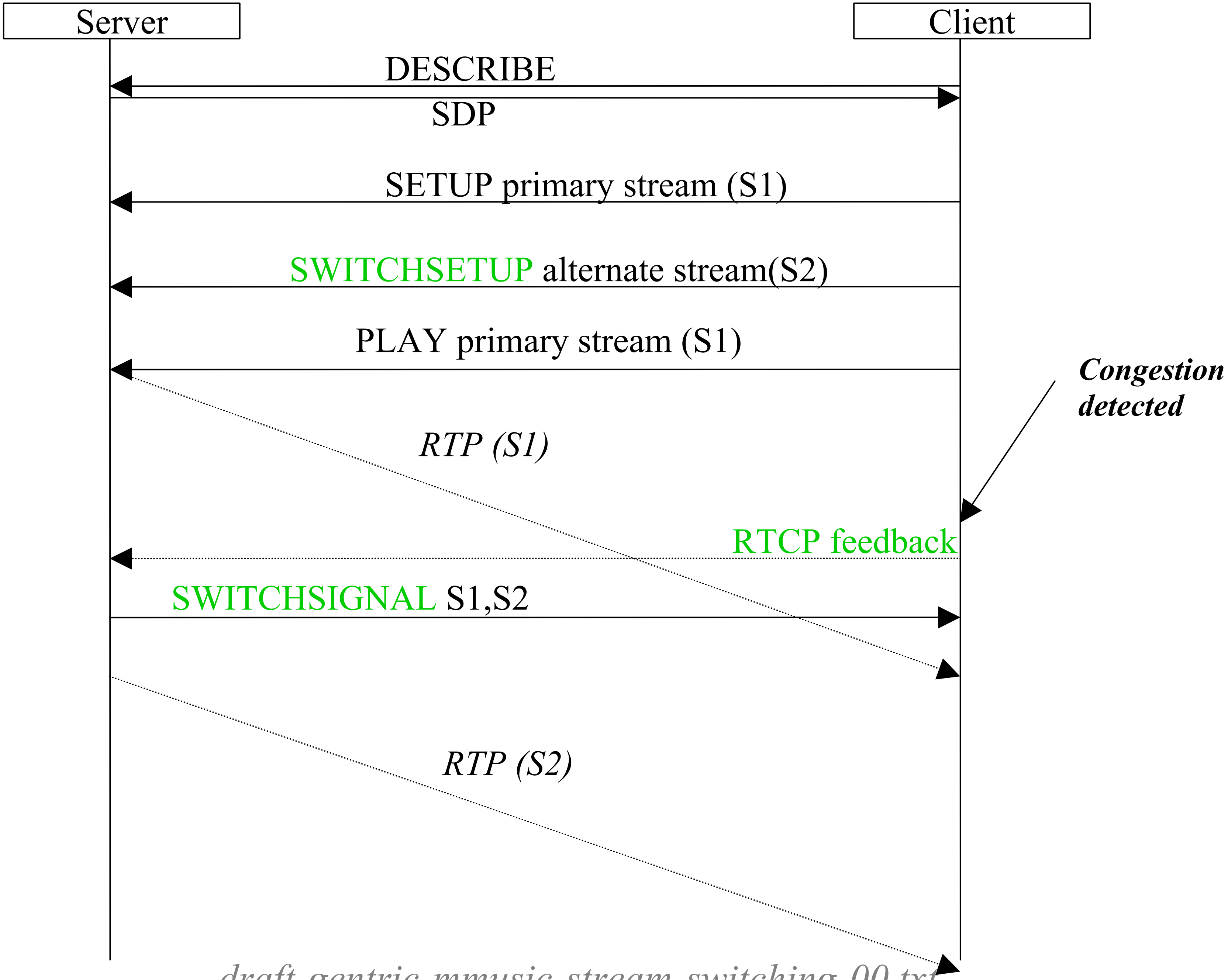
Client Transparent	• « Same everything but the bit rate » • RTP = same Payload Type • In theory: « no need to warn the client » • But warning can help • Also exact timing info (as in play response)
Non Client Transparent	• Different decoder/renderer configuration • RTP = different Payload Type • ***Client needs to be ready in advance***: • At session start (wasteful) • With explicit forewarning

Proposal: dedicated (and orthogonal) RTSP Methods

- **SWITCH (C->S)**
  - Explicit exchange of streams for the same track
  - Atomic for (PLAY+PAUSE) with « as soon as possible » as target media time
- **SWITCHSIGNAL(S->C)**
  - Sender forewarning before the switch
- **SWITCHSETUP (C->S)**
  - Same as SETUP, but explicitely for alternate streams ***only***
  - Additional header fields for parametrization of:
    - Server initiated switches and forewarnings
    - Random Access Points policy
- **SWITCHCLOSE (C->S,S->C)**
  - Symetric to SWITCHSETUP
  - Ressource garbage collection

draft-gentric-mmusic-stream-switching-00.txt

Server initiated



Server

Client

DESCRIBE

SDP

SETUP primary stream (S1)

SWITCHSETUP alternate stream(S2)

PLAY primary stream (S1)

Congestion detected

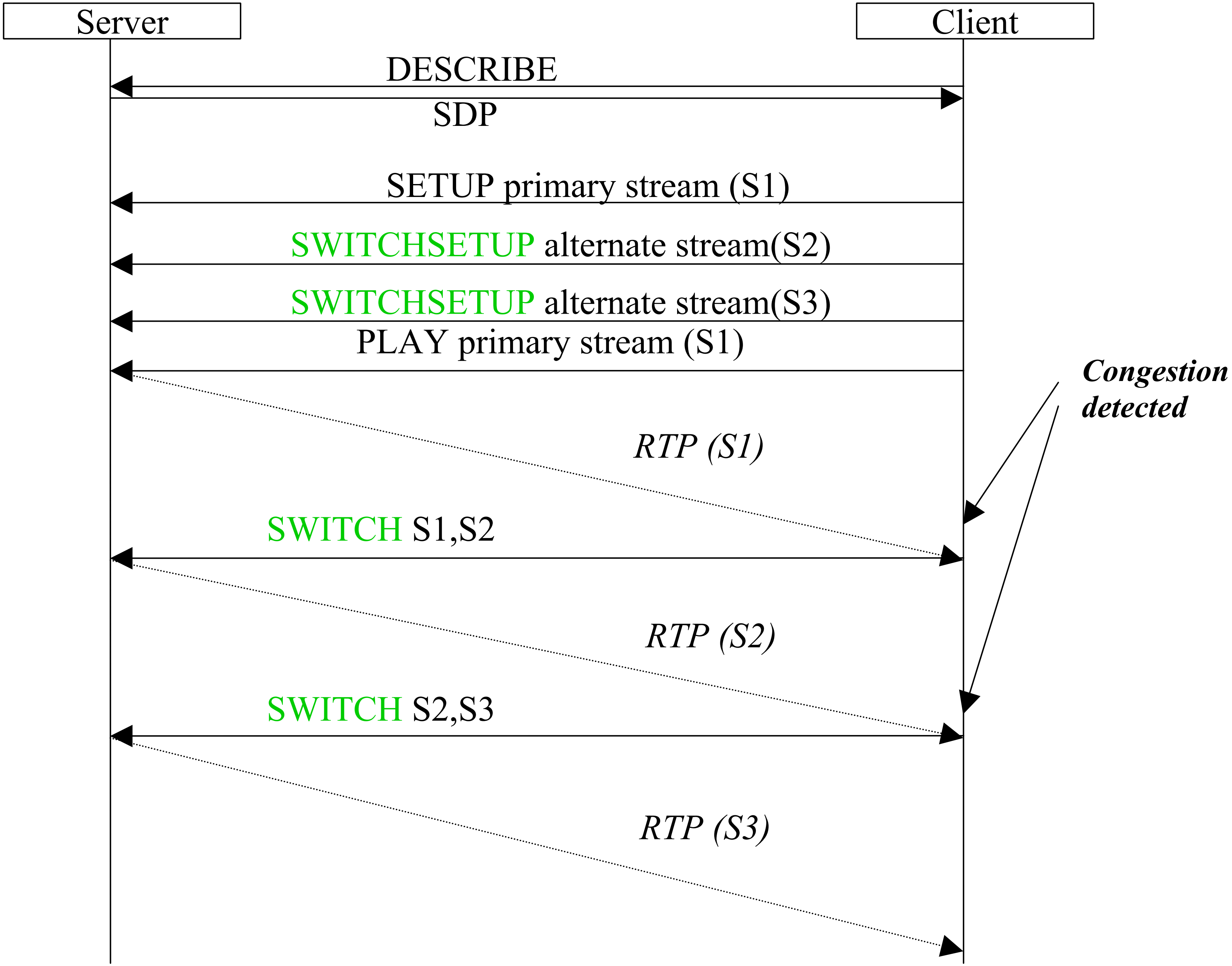
RTP (S1)

RTCP feedback

SWITCHSIGNAL S1,S2

RTP (S2)

Client initiated



Server

Client

DESCRIBE

SDP

SETUP primary stream (S1)

SWITCHSETUP alternate stream(S2)

SWITCHSETUP alternate stream(S3)

PLAY primary stream (S1)

RTP (S1)

SWITCH S1,S2

RTP (S2)

SWITCH S2,S3

RTP (S3)

Congestion detected

Conclusion

Questions ?

Working group item ?