



Sender-Side Track Switching (SSTS)

<https://github.com/moq-wg/moq-transport/pull/1638>

Will Law, Suhas Nandakumar
IETF #126, July 2026

SSTS timeline

- 9 February - Boulder - design team initial proposal
- 12 Feb - draft established at <https://github.com/wilaw/dts4moq>
- March - POC built by Cisco
- IETF #125 - Shenzhen - presentation.
- 11 May - IETF moq interim - presentation and vote to include in MOQT
- May - POC built by Nokia
- 1 June - PR #1638 submitted against MOQT
- June 11 - presented at London Interim
- July 1 - updates to PR #1638 incorporating London feedback.

Renaming

Old name: Dynamic Track Switching (DTS)

New name: **Sender-Side Track Switching (STSS)**

Name was changed to better describe the functionality being added.

Making the design extensible to support future algorithms

SSTS has been modified to be algorithmically extensible.

We now define a baseline SWITCHING-SET-ASSIGNMENT parameter with two mandatory fields - the switching set ID and the algorithm.

```
SWITCHING-SET-ASSIGNMENT {  
    Switching set ID (vi64),  
    Algorithm (vi64),  
    [Algorithm dependent fields]  
}
```

Algorithms are registered in a new "SSTS Algorithms" IANA registry. Each algorithm is responsible for defining:

- The additional fields to be passed in the SWITCHING-SET-ASSIGNMENT parameter
- The server behavior
- The client behavior

During SETUP, the server communicates the list of SSTS algorithms which it supports, via the new SSTS_ALGORITHMS option (Option Type 0x09). Supported algorithms are serialized as a sequence of varints. Returning an empty sequence is acceptable and indicates that SSTS is not supported.

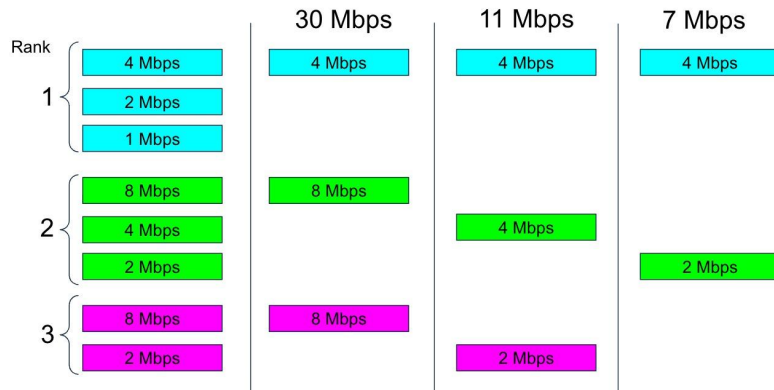
Making the design extensible to support future algorithms

A default algorithm 0 has been defined by MOQT.

It adds a series of fields to the SWITCHING-SET-ASSIGNMENT parameter and defines the server and client behaviors in implementing the algorithm.

These rules are identical to those originally prescribed by the PR.

```
SWITCHING-SET-ASSIGNMENT {  
    Switching set ID (vi64),  
    Algorithm ID (vi64),  
    Throughput threshold (vi64),  
    Set throughput fraction (vi64),  
    Activate switching (vi64),  
    [Set rank (8)]  
}
```



DDOS protection

We originally proposed limits communicated at SETUP, including `MAX_DTS_CONCURRENT-THROUGHPUT` and `MAX_DTS_CONCURRENT_TRACKS`.

In the London discussion, it was pointed out that these limits are trivially bypassed: a limit of max concurrent tracks per session can be circumvented by opening up 1,000 sessions.



Since SETUP restrictions apply per connected session, and we have no limit on the number of sessions a client can establish, we decided to remove the SETUP limits and instead rely on native relay protection limits, which need to be implemented anyway.

This does not mean that SSTS does not represent a significant DDOS vector, as there can be high asymmetry in the downstream versus upstream traffic. Similar asymmetry can be achieved by other mechanisms, such as subscribing to a high bandwidth stream and then reading it very slowly.

Other issues decided in London

1. Do we need this ability to remove a SUBSCRIPTION from a set without unsubscribing?
 - a. Not with the simple algorithm 0. Future algorithms may implement a more sophisticated removal mechanism.
2. Does repeating the Switchign Set ID , throughput fraction and rank a good design?

	1080P	720P	360P
SWITCHING-SET-ASSIGNMENT {	{	{	{
Switching set ID (vi64),	1,	1,	1,
Throughput threshold (vi64),	4000,	2000,	1000,
Set throughput fraction (vi64),	10,	10,	10,
Activate switching (1),	0,	0,	1,
[Set rank (8)]	1	1	1
}	}	}	}

- a. Yes, as it avoids an extra RTT in defining the set separately from the assignment.
3. Should we add timeout and stability measures to the input parameters?
 - a. Not with the simple algorithm 0. Future algorithms may implement a more sophisticated removal mechanism.

Status of PR #1638

- All major issues have resolved or have comments which we believe satisfy. If you have an open comment, please check if it has been resolved.
- Decision and changes from London have been implemented.
- Merge conflicts resolved.

We believe the PR is ready to be merged.

Thank you for your time.