

## How to enable TCP Fast Open in NetScaler?

TCP Fast Open (TFO) is a mechanism in TCP connection establishment process, which helps to speed up the opening of the connections and data flow. It allows data to be carried during the initial TCP connection handshake, i.e. in SYN and SYN-ACK packets and enables the data to be consumed by the receiving node during the connection establishment thus speeding up data transfer while the connection is being established.

### Why TCP Fast Open?

Let us understand how TFO works and how it is useful under different use cases. The idea of TFO originated because of the increasing performance requirements of today's applications.

TFO process saves up to one full round trip time commonly referred as RTT in TCP handshake when compared with the conventional TCP three way handshake. For applications that have short web transfers, this additional RTT makes a significant impact in overall latency in the network.

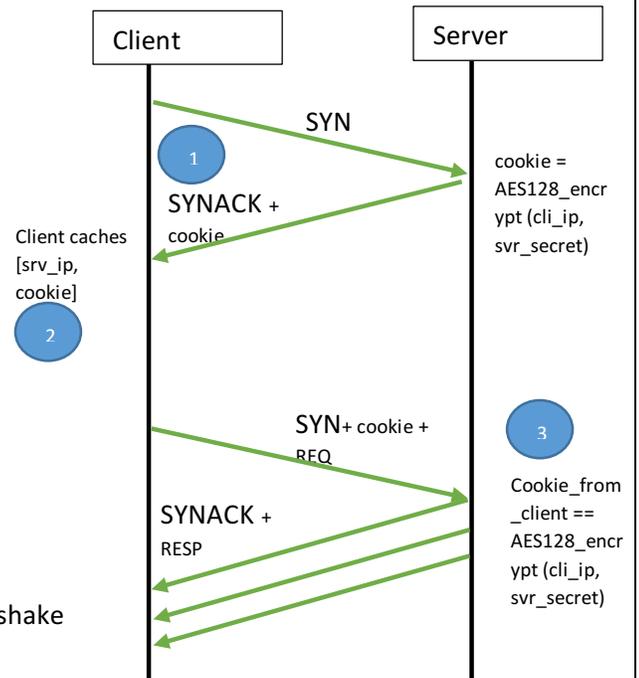
For example in Chrome browser, it is found out that on an average, one-third of the connections are new TCP connections and thus with TFO involved, it can provide substantial improvements in performance of the network.

Now what about the security of data that is transferred in the initial handshake? TFO makes the data exchange secure by using a TFO cookie, a cryptographic cookie which is sent from receiving node and gets stored on the client who initiated the connection. When the client tries to connect to the same node again, it sends the TFO cookie along with the SYN packet during the handshake thus authenticating itself with the receiving node.

Upon successful authentication, the receiving node will send data to the client without receiving the final acknowledgement thereby saving one RTT to start data transmission. This reduces the overall network latency to a significant level when short lived connections are very high.

## TCP TFO flow

1. Server
  - Receives SYN + TFO cookie request
  - Generates cookie by encrypting client IP
  - Sends SYN-ACK + TFO cookie
2. Client
  - Caches cookie for this server IP
3. Server
  - Receives SYN + TFO cookie + data
  - Validates client TFO cookie + accepts connection + data is made available to application
  - Sends SYN + ACK for SYN+ Data in SYN packet
  - Sends more data packets to client while handshake is in progress



## To enable TCP TFO by using the NetScaler GUI

Navigate to **System > Profiles > TCP Profiles** and click **Edit** after selecting the TCP Profile to be edited

System / Profiles / TCP Profiles

### Profiles

TCP Profiles 12 HTTP Profiles 2 Database Profiles 0 SSL Profile 2 DTLS Profile 1 DNS Profile 1 LB Profile 0

Add Edit Delete Search ▾

<input type="checkbox"/>	Name	Maximum Burst Limit	Initial Congestion Window Size	TCP Delayed ACK Time-out (msec)
<input checked="" type="checkbox"/>	nstcp_default_profile	6	4	100
<input type="checkbox"/>	nstcp_default_tcp_lfp	10	6	100
<input type="checkbox"/>	nstcp_default_tcp_lnp	4	4	100
<input type="checkbox"/>	nstcp_default_tcp_lan	10	6	100

# Configure TCP Profile

Name  
nstcp\_default\_profile

Window Scaling

Window scaling factor  
4

Maximum Burst Limit  
6

Initial Congestion Window Size  
4

TCP Delayed ACK Time-out (msec)  
100

Maximum ooo packet queue size  
64

MSS  
1460

Maximum Packets per MSS  
0

Maximum Packets Per Retransmission  
1

Minimum RTO (in millisecc)  
1000

Slow start increment  
2

TCP Buffer Size (bytes)  
8190

TCP Send Buffer Size (bytes)  
8190

TCP Maximum Congestion Window Size  
524288

TCP Dupack Threshold  
3

TCP Burst Rate Control  
DISABLED

TCP Connection Payload Send Rate (Kb/s)  
0

Maximum Connection Queue Size (bytes)  
0

TCP Flavor\*  
Default

Establish Client Connection  
AUTOMATIC

TCP Segmentation Offload  
AUTOMATIC

TCP Optimization Mode  
TRANSPARENT

Reference Count  
22

### Keep Alive

Keep-alive probes

Connection idle time before sending probe (secs)  
900

Keep-alive probe interval (secs)  
75

Maximum Keep-alive probes  
3

Update last activity for KA probes

### Multipath TCP

Multipath TCP

Drop Data on Pre-Established subflow

Fast Open

Session Timeout  
0

Selective Acknowledgement

Immediate ACK on receiving packet with PUSH

RST Window Attenuation

Explicit Congestion Notification (ECN)

ACK Aggregation

CUBIC Hystart

Forward Acknowledgement

TCP SYN Cookie

RST Acceptance

TCP Timestamp

Forward RTO-Recovery

Use Nagle's algorithm

Dynamic Receive Buffering

SYN Spoof Protection

Duplicate SACK

TCP Fast Open

OK

Close

## To enable TCP TFO by using the NetScaler Command prompt

```
>set tcpprofile nstcp_default_profile -tcpFastOpen ENABLED
```

```
show tcpprofile
```

```
1) Name: nstcp_default_profile
Window Scaling status: DISABLED
Window Scaling factor: 4
Selective Acknowledgement(SACK) status: DISABLED
Maximum Segment Size(MSS): 1460
Maximum TCP segments allowed in a burst: 6 MSS
Initial congestion window(cwnd) setting : 4 MSS
TCP Delayed-ACK Timer: 100 millisecc
Nagle's Algorithm: DISABLED
Maximum out-of-order packets to queue: 64
Immediate ACK on PUSH packet: ENABLED
Maximum packets per MSS: 0
Maximum packets per retransmission: 1
TCP minimum Retransmission Timeout(RTO) in millisecc: 1000
TCP Slow start increment: 2
TCP Buffer Size: 8190 bytes
TCP Send Buffer Size: 8190 bytes
TCP Syncookie: ENABLED
Update Last activity on KA Probes: ENABLED
TCP flavor: Default
TCP Dynamic Receive Buffering: DISABLED
Keep-alive probes: DISABLED
Connection idle time before starting keep-alive probes: 900 seconds
Keep-alive probe interval: 75 seconds
Maximum keep-alive probes to be missed before dropping connection: 3
Establishing Client Connection: AUTOMATIC
TCP Segmentation Offload: AUTOMATIC
TCP Timestamp Option: DISABLED
RST window attenuation (spoof protection): DISABLED
Accept RST with last acknowledged sequence number: ENABLED
SYN spoof protection: DISABLED
TCP Explicit Congestion Notification(ECN): DISABLED
Multipath TCP: DISABLED
Multipath TCP drop data on pre-established subflow: DISABLED
Multipath TCP fastopen: DISABLED
Multipath TCP session timeout: 0 seconds
Duplicate Selective Acknowledgement(DSACK): ENABLED
ACK Aggregation: DISABLED
Forward RTO recovery(FRTO): DISABLED
```

TCP Max congestion window(CWND): 524288 bytes  
TCP Forward Acknowledgment(FACK): DISABLED  
TCP Optimization mode : TRANSPARENT  
**TCP Fastopen: ENABLED** >>> TCP Fast Open is Enabled  
TCP Hybrid Start(HYSTART): DISABLED  
TCP dupack threshold: 3  
Burst Rate Control: DISABLED  
TCP Rate: 0  
TCP Rate Maximum Queue: 0  
Reference count: 21

Note: Default value for TCP Fast Open is DISABLED. NetScaler supports TCP Fast open feature from release 11.1 onwards.

Thus using TCP Fast Open feature in NetScaler, overall network latency can be improved significantly in networks with short lived web connections and high rate of new TCP connections.