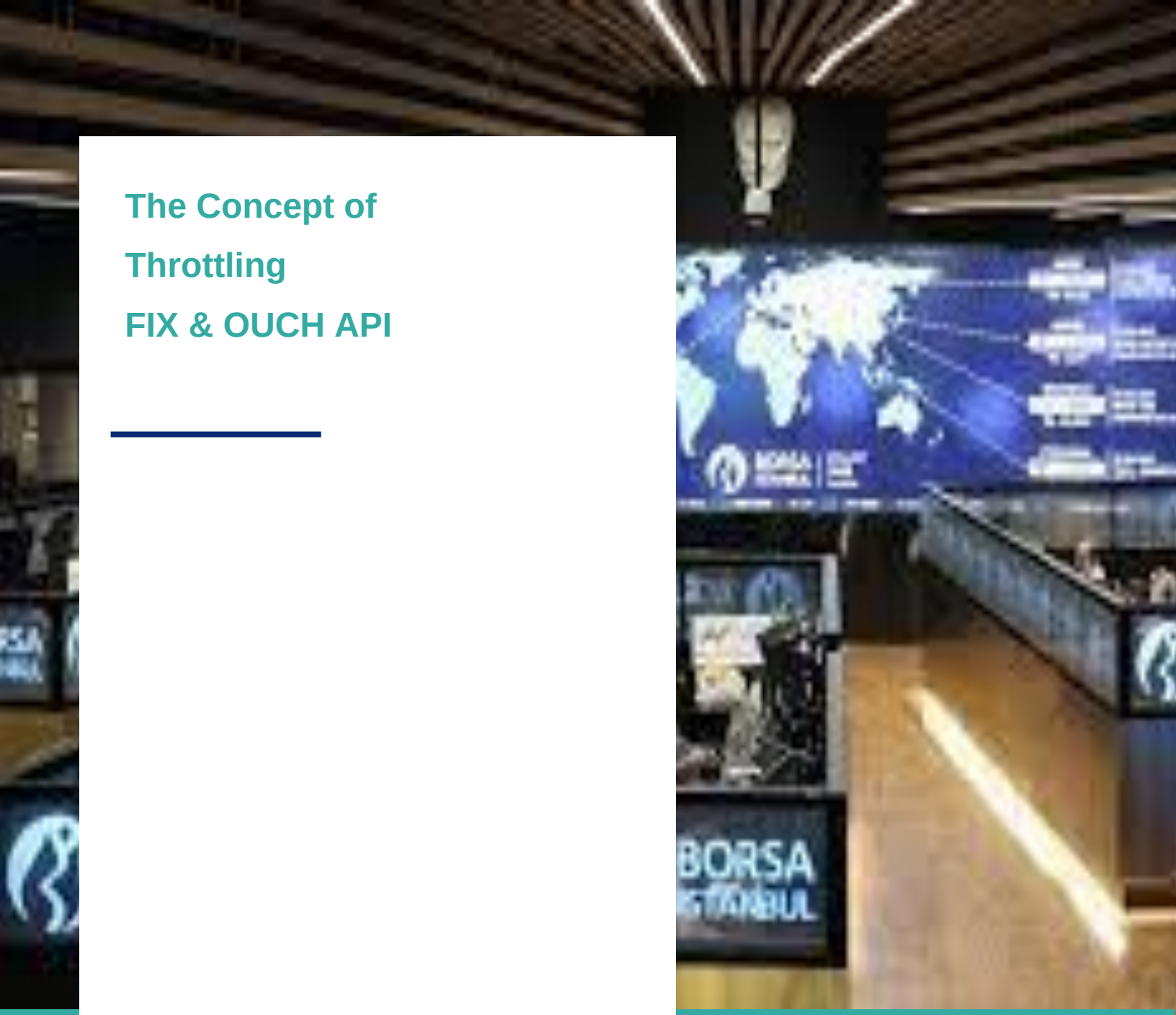


The Concept of Throttling FIX & OUCH API

2022

BISTECH

INDEX

FIX API - THROTTLING

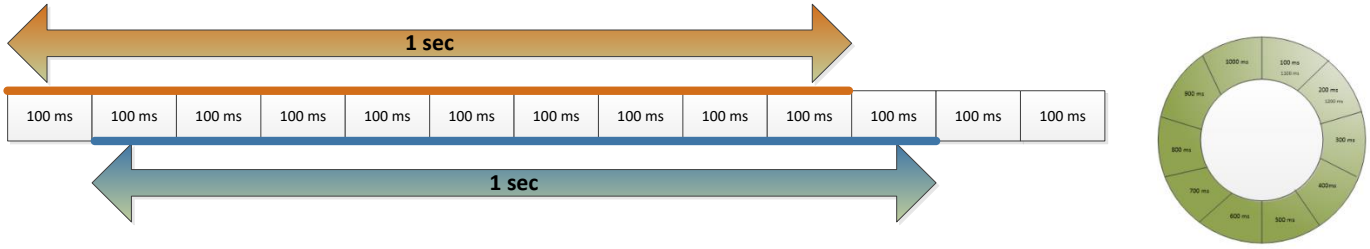
EMİR GÖNDERİM LİMİTİ \ORDER TRANSMISSION LIMIT

BISTECH FIX sisteminde emir iletimi yapılırken üyenin kendi emir/saniye kapasitesini (kota-throttling) aşması durumda üyeye Business Message Reject (35= j) hata mesajı dönülmektedir. Emir/saniye kapasitesi izleyen saniyelik dilimlerde hesaplanmakta ve işlenmektedir. Burada 100 milisaniyelik (ms.) kayan pencere yapısı söz konusudur. Yani saat akışındaki 1 saniyelik (sn.) limitler değil zamanın her anındaki birbirini takip eden 10 adet 100 ms.'lik pencereler kontrol edilerek emir kabul limitleri hesaplanmaktadır. Bu pencereler FIX sunucularında (GW'de) dairesel şekilde birbirlerini takip etmektedir. Kapasitesinin aşılması durumunda ise FIX oturumu sonlandırılmamaktadır.

Pencere yapısının detayını aşağıda görebilirsiniz.

BISTECH FIX System returns a Business Message Reject (35= j) in case members exceed the order limit per second (Throttling Quota) that have been predefined for their users while sending an order through FIX - BISTECH system. The capacity of the order limit per second is calculated and processed in the following one second period. There is a "sliding windows" structure consisted of ten 100 millisecond-units. Throttling limit is not checked according to the system clock but via 10 consecutive windows each lasts 100 milliseconds through the entire time. These windows follow each other in a circular manner on FIX gateways. In case the capacity of the order limit per second is exceeded, FIX session will not be terminated.

You can see the details of the structure below;



Üye FIX bağlantılarında Throttling mekanizması nedeni ile aşağıdaki konuların dikkate alınması tavsiye edilmektedir.

1- Saniyelik throttling hakkı üye tarafından milisaniye mertebesinde dilimlere bölünerek gönderilebilir.
2- Gönderilen FIX mesajlarının üzerindeki sending time tag'i kullanılmamalı (FIX Client uygulamasının bastığı), network üzerinde geçen zamanı da dikkate alınarak bir zaman dilimi planlanmalıdır.
3- Throttling limitine ulaşıldığı anda FIX sunucuları ilgili kullanıcıdan gelen mesajları kabul etmez ve Business Message Reject (BMR) hata mesajı ile reddedilir. Bir sonraki 100 milisaniyelik dilim içerisinde üyenin geriye doğru toplam 10 penceresine bakılarak mesajların işlenmesine kapasite kullanımına göre tekrar başlanabilir.
4- Gönderilen istek mesajları ile Borsa tarafından geri iletilen işlendi mesajları (Execution Report – Order Ack (out) içindeki 52 tag'li "SendingTime" alanı) arasındaki zaman farkı dinamik olarak kontrol edilerek emir gönderim zamanlaması belirlenebilir. Örneğin, kota dahilindeki her saniyedeki ilk emir ile son emirin sistemden (TE) dönen ExecutionReport (ER) mesajlarındaki time stamp'leri arasındaki zaman farkı (tfark) hesaplanır (tn-t1= tfark). Ardından son mesajın TE'den dönen ER mesajındaki zaman değerine 1 saniye eklenip (tn+1000) ikinci emir paketi

Because of Throttling Mechanism, the recommendations described below should be taken into consideration on client side during FIX connection.

1- Throttling limit can be managed in millisecond-level instead of second-level on client side.
2- SendingTime of sent FIX messages should not be used (generated via FIX Client tool). Instead timestamp on the Execution Report message should be used. A time interval should be planned by considering the time passed on network.
3- When the throttling limit is reached, incoming messages are not accepted by FIX servers and they are rejected with a Business Message Reject (BMR). It can be started again processing messages according to capacity utilization within the next 100 milliseconds period by looking backwards through a total of 10 windows.
4- Order transmission timing can be determined by controlling time lag between sent messages and reply execution messages dynamically (Tag52 : SendingTime in Execution Report – Order Ack (out)). For example, the time lag(tdiff.) for Execution Report (ER) messages of first order and the last order within the capacity in a single second is calculated (tn-t1= tdifference). For the next message packet, client should wait 1 second after the execution message of

gönderilebilir. İkinci paketin gönderim zamanı tfark değerinden büyük olmamak üzere belli bir miktar geriye çekilebilir. Bu değer tamamen üyenin kendi deneyimleriyle belirlenmelidir.

last message is received (tn +1000). Timing of second packet can be brought to an earlier time for a certain amount, as long it is not brought earlier more than the time difference (tdifference). The amount of time to bring earlier should be determined by user experience.

ÖRNEK\SAMPLE (FIX API)

Throttling mekanizması sonucunda karşılaşılabilecek durumları aşağıdaki örnekte görebilirsiniz.

A: Pencere sıra numarası
B: Toplam süre (ms.)
C: Kayan pencere büyüklüğü (ms.)
D: İlgili pencere içerisinde gönderilen emir miktarı

Instances resulting from Throttling mechanism can be seen below example.

A: Window sequence number
B: Total time (ms.)
C: Sliding window size (ms.)
D: The amount of order sent within the window

A	1	2	3	4	5	6	7	8	9	10	11	…	20
B	100	200	300	400	500	600	700	800	900	1000	1100		2000
C	100	100	100	100	100	100	100	100	100	100	100	100	100
D	30	56	14	0	0	0	0	0	0	0	100 (!)		

(!)(Pacing on),
70 messages rejected with BMR.

(!)(Pacing on),
70 mesaj BMR ile reddedilir.

100 kotalık bir kullanıcının gönderdiği emirler yukarıdaki gibi farklı pencerelerde yer alabilmektedir.

100. emri gönderdikten sonra ikinci 100'ün hiçbir kontrol yapılmadan 1001. ms.'de gönderilmesi halinde ilk saniyeye ait 2. ve 3. 100 ms.'lik frame'lerdeki 70 mesajdan dolayı 30 mesaj kabul edilecek ve kalan 70 mesaj BMR ile reddedilecektir.

Orders coming from a user having throttling limit of 100 messages/second can be received in separate time windows as above.

After sending 100th order, if the second group of 100 orders is sent in 1001th ms. without doing any control, only first 30 messages will be accepted and remaining 70 messages will be rejected with BMR. This is because at 11th time interval, 30 of the capacity are freed at 1st time interval.

OUCH API - THROTTLING

EMİR GÖNDERİM LİMİTİ \ORDER TRANSMISSION LIMIT

BISTECH sisteminde OUCH emir iletimi yapılırken, üyenin kendi emir/saniye kapasitesi (throttling) çerçevesinde emir göndermesi beklenmektedir.
u
Emir/saniye kapasitesi izleyen saniyelik dilimlerde hesaplanmakta ve işlenmektedir. Burada 100 milisaniyelik (ms.) kayan pencere yapısı söz konusudur. Yani saat akışındaki 1 saniyelik (sn.) limitler değil zamanın her anındaki birbirini geriye doğru toplam takip eden 10 adet 100 ms.'lik pencereler kontrol edilerek emir kabul limitleri hesaplanmaktadır. Bu pencereler OUCH sunucularında (gateway) dairesel şekilde birbirlerini takip etmektedir.

OUCH emirleri ilk olarak 64K büyüklüğündeki TCP arabelleğine (buffer), oradan da uygulama arabelleğine üye kapasitesi dâhilinde aktarılmaktadır.

Üyenin kapasitesini aşması durumda, kapasiteyi aşan emirlerin uygulama arabelleğe aktarılması geçici olarak durdurulmakta ve izleyen saniyelik dilimlerde aktarıma devam edilerek işlenmektedir.

It is expected to send orders within order limit per second (Throttling Quota) that have been predefined for member users while sending an order through OUCH.

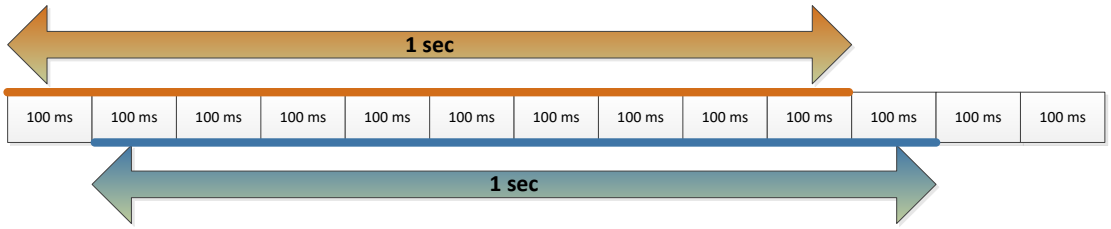
The capacity of the order limit per second is calculated and processed in the following one second period. There is a "sliding windows" structure consisted of ten 100 millisecond-units. Throttling limit is not checked according to the system clock but via 10 consecutive windows each lasts 100 milliseconds through the entire time. These windows follow each other in a circular manner on OUCH Gateways.

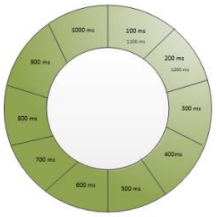
OUCH orders are first transmitted to the TCP buffer of 64K and then to the application buffer.

In case the capacity of the order limit per second is exceeded, OUCH session will be terminated. The member who is faced with this situation should re-logon his session immediately.

Uygulama arabelleğe aktarımın geçici olarak durdurulduğu zaman diliminde, gönderilmeye devam eden emirlerin TCP arabellek kapasitesini de aşması durumunda OUCH oturumu sonlandırılmaktadır. Bu durum ile karşılaşan üyenin oturumunu zaman kaybetmeden tekrar açması gerekmektedir.

The OUCH session is terminated in case the orders that continue to be sent exceed the TCP buffer capacity during the time period when the application buffering is temporarily stopped. The member who encounters this situation must log in again without wasting time.





ÖRNEK\SAMPLE (OUCH API)

Throttling mekanizması sonucunda karşılaşılabilecek durumları aşağıdaki örnekte görebilirsiniz.

A: Pencere sıra numarası
B: Toplam süre (ms.)
C: Kayan pencere büyüklüğü (ms.)
D: İlgili pencere içerisinde gönderilen emir miktarı

Instances resulting from Throttling mechanism can be seen below example.

A: Window sequence number
B: Total time (ms.)
C: Sliding window size (ms.)
D: The amount of order sent within the window

A	1	2	3	4	5	6	7	8	9	10	11	…	20
B	100	200	300	400	500	600	700	800	900	1000	1100		2000
C	100	100	100	100	100	100	100	100	100	100	100	100	100
D	30	56	14	0	0	0	0	0	0	0	100 (!)		

(!) Alarm durumu (Pacing on)
Bağlantı kopma durumu oluşabilir.

(!)Alarm situation (Pacing on)
Disconnect situation can be occurred.

100 kotalık bir kullanıcının gönderdiği emirler yukarıdaki gibi farklı pencerelerde yer alabilmektedir.

100. emri gönderdikten sonra ikinci 100'ün hiçbir kontrol yapılmadan 1001. ms.'de gönderilmesi halinde ilk saniyeye ait 2. ve 3. 100 ms.'lik frame'lerdeki 70 mesajdan dolayı 30 mesajın alınıp pacing kontrolüne girmektedir.

Orders coming from a user having throttling limit of 100 messages/second can be received in separate time windows as above.

After sending 100th order, if the second group of 100 orders is sent in 1001th ms. without doing any control, only first 30 messages will be accepted and remaining 70 messages will result in pacing enabling and will be buffered. This is because at 11th time interval, 30 of the capacity are freed at 1st time interval.

Arabellek büyüklüğü ile orantılı olarak kullanıcının kalan mesajı arabellekte tutulabilir ve bağlantı koparılmayabilir. Ancak kullanıcının arabellek kapasitesini aşması durumunda bağlantının sonlandırılması muhtemeldir.

Remaining messages will be buffered and connection will be remained active. However, if the number of messages to be buffered exceeds the throttling limit, disconnection may occur.

İlk saniyede limit kadar gönderdikten sonra, takip eden zaman dilimlerinde ardışık iki mesaj arasında 100 ms.'den daha fazla fark varsa buna bakarak "pacing on" durumuna düşülüp düşülmediği anlaşılabilmektedir.

After sending total limit in the first second, and then in the following time intervals if there is 100 ms. between 2 messages, it can be understood whether or not pacing on is active.

Throttling mekanizması nedeni ile OUCH bağlantılarının kopmaması üyelerin sorumluluğundadır.

It is the members' own responsibility to ensure that the OUCH connections are not disconnect due to the throttling mechanism.