



Real-time Communication Protocol With JSON Syntax

Serkan Ayaz and Semih Bilgen

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

September 25, 2019

JSON Sözdizimli Gerçek Zamanlı İletişim Protokolü

Serkan Ayaz¹ ve Semih Bilgen²

¹ Yüksek Lisans Öğrencisi, İstanbul Okan Üniversitesi, Akfırat 34959, Türkiye
serayaz@stu.okan.edu.tr

² Danışman, İstanbul Okan Üniversitesi, Akfırat 34959, Türkiye
semih.bilgen@okan.edu.tr

Abstract. IoT, kritik görev yazılımları, kurumsal uygulamalar, mikroservisler vb. alanlarda gerçek zamanlı ve platformdan bağımsız iletişim altyapılarına ihtiyaç duyulmaktadır. Bu konuda TCP protokolünün Unicast, Multicast ve Broadcast tipi iletişim tipleri yetersiz kalmaktadır. Aracı sunucu üzerinden yapılan veri aktarımlarında ağ trafiği azaldığından veri iletim performansı artmaktadır ve aktarılan veriler daha optimum seviyede kullanılmaktadır. Bu çalışmada platformdan bağımsız ağ istemcileri arasında gerçek zamanlı iletişim kurmak için aracı sunucu üzerinden verilerin aktarıldığı metin tabanlı ve JSON sözdizimli JTP (*JSON Transmission Protocol*) adında yeni bir protokol önerilmektedir. Yapılan çalışmada STOMP, XMPP, RESP ve NATS protokolleri incelenerek protokol boyutları ve algoritma performansı, çerçeve boyutları ve algoritma performansı ve insanlar tarafından okunup kodlanabilme seviyeleri değerlendirilmiştir. İncelenen protokoller ile JTP protokolü, performans testleri yapılarak karşılaştırılmıştır. Yapılan değerlendirme sonucunda JTP'nin diğer alternatiflerine göre daha fazla işlevsel özelliğe sahip olduğu gösterilmiştir.

Keywords: Protokol, JSON, Aracı Sunucu, TCP, STOMP, XMPP, RESP, NATS, Gerçek Zamanlı İletişim

1 Giriş

İletişim protokolleri bir mesajı veya veriyi diğer bir uç noktaya güvenli ve tutarlı bir şekilde ulaştırmak amacıyla kullanılır. Ancak pratikte bir verinin sadece diğer noktaya iletilmesi yeterli değildir. Veri iletiminin yanısıra iletim kontrollerini de üstlenecek bir yapıya ihtiyaç duyulmaktadır. Bu tip iletişim sistemleri, kurumsal uygulamalarda veri kaybına bağlı hata toleransı sifıra yakın, kritik görevler üstlenen sunucular arasında iletişimin sağlanması, çok sayıda istemciden (IoT cihazları, loglayıcı vb.) verinin toplaması, aynı anda birden fazla hedefe gerçek zamanlı veri aktarımının sağlanmasında, mikroservis [1] tabanlı altyapılardaki servisler arası iletişimin sağlanması gibi amaçlarla kullanılmaktadır. İletişim protokolleri ikili (*binary*) ve metin tabanlı olmak üzere ikiye ayrılır. Metin tabanlı protokollerin programlama dillerinden ve işletim sistemlerinden bağımsız olmaları tüm platformlar arası iletişim durumlarında kullanılmasına olanak tanımaktadır.

Aracı sunucu (*broker server*), kendisine bağlanan istemciler arasında veri yönlendirmesi yaparak istemcilerin birbiriyle haberleşmesini sağlayan sunucu yazılımıdır. Bir verinin birçok uç noktaya iletilmesinde TCP/IP protokolündeki Unicast, Multicast ve Broadcast'de karşılaşılan problemler aracı sunucu ile çözülmektedir. Ayrıca aracı sunucu ile TCP/IP'nin bir özelliği olan gerçek zamanlı iletişim de desteklenmektedir.

Bu çalışmada, önce, piyasada kullanılan aracı sunucusuna sahip metin tabanlı protokollerin özellikleri incelenerek avantaj ve dezavantajları araştırılmıştır. Araştırma sonucunda genel olarak protokollerin iletişim için yüksek veri boyutu gerektirmesi, kompleks iletişim tiplerini desteklememesi, protokol sözdizimlerinin insanlar tarafından okunup yazılamaması gibi dezavantajlarının olduğu görülmüştür. Çalışmada daha sonra bu temel dezavantajları ortadan kaldıran veya en aza indiren ve bir aracı sunucuyla çalışarak aşağıdaki özellikleri içeren gerçek zamanlı iletişim için kullanılmak üzere JSON (JavaScript Object Notation) [2] sözdizimli yeni bir protokol önerilecektir. Bu protokol, metnin devamında JTP (JSON Transmission Protocol) olarak anılacaktır. JTP protokolünün işlevsel özellikleri şunlardır:

- Gerçek zamanlı çok hedefli iletişim desteği
- Platform bağımsız veri iletimi
- Yayımcı-abone tipi iletişim desteği
- Arayan-aranan tipi iletişim desteği
- Yük dengeleyicili kuyruk desteği
- Süre limitli veri iletimi desteği
- QoS (Quality of Service) iletim denetimi desteği
- Geriye dönük mesajlara erişim desteği
- Protokolün genişletilebilir olması
- Verilerin gerçek zamanlı sıkıştırılarak iletimi
- Kimlik doğrulama desteği
- İletişim tipi ve konu bazlı yetkilendirme
- İzleme desteği

2 Literatür Araştırması

2.1 Araştırma Kriterlerinin Belirlenmesi

Araştırma kapsamının belirlenmesinde aşağıdaki kriterler göz önünde bulundurulmuştur:

- Metin tabanlı veri iletimi
- Aracı sunucu üzerinden haberleşmesi
- Gerçek zamanlı iletişim

2.2 Protokolleri Karşılaştırma Kriterleri

Her protokolün çeşitli avantaj ve dezavantajları bulunmaktadır. Mesajlar ağ üzerinde bir uçtan diğer bir uca aktarılırken protokol ve çerçeve gibi ek veriler eklenerek

aktarıldığından protokol ve çerçeve verilerinin boyutu iletişim performansını etkilemektedir [3]. Ayrıca protokol ve çerçeve verilerinin kodlama ve çözümlenme işleminde kullanılan algoritmanın karmaşıklığı işlemci kaynaklarını tüketmesi nedeniyle iletişim performansını etkilemektedir [4]. Protokol ve çerçevenin kodlama ve çözümlenme işleminin insanlar tarafından yapıldığında protokol ve çerçevede kullanılan harf ve sayıların dışındaki tüm karakter tiplerinin toplam sayı değeri insanlar tarafından anlaşılabilirliğin ölçütü olarak belirlenmiştir [5]. Bu çalışmanın kapsamına yönelik aşağıdaki karşılaştırma kriterleri belirlenmiştir.

- Protokol boyutu
- Protokol kodlama ve çözümlenmede algoritma performansı
- Çerçeve boyutu
- Çerçeve kodlama ve çözümlenmede algoritma performansı
- İnsanlar tarafından anlaşılabilirliği ve yazılabilirliği

2.3 Araştırılan Protokoller

Bölüm 2.1’de belirtilen araştırma kriterlerine göre bu çalışmada aşağıdaki protokoller araştırılmıştır. Bu kriterlere uymayan diğer protokoller bu çalışmanın kapsamı dışındadır. Araştırma kriterlerinin bir kısmını karşılayıp bir kısmını karşılamayan protokoller de mevcuttur ve bunlar da kapsam dışı bırakılmıştır.

- STOMP
- XMPP
- RESP
- NATS

Protokol karşılaştırmamızın sonuçları, Çizelge 1’de özetlenmiştir.

3 Yeni Protokol Önerisi

Bölüm 1’de verilen işlevsel özellikleri barındıran platfordan bağımsız ağ istemcileri arasında gerçek zamanlı iletişim kurmak için aracı sunucu üzerinden verilerin aktarıldığı metin tabanlı ve JSON sözdizimli yeni bir protokol önerilmektedir.

3.1 Karşılaştırma Kriterlerine Göre Protokolün Testi

Araştırılan protokollerin karşılaştırma kriterlerine göre protokol testleri, araştırılan protokollerin ve JTP protokolünün desteklediği yayımcı-abone tipi iletişimin yayınlama protokolü örneği üzerinden yapılmıştır. JTP protokolünün testi için aşağıdaki JTP protokol örneği kullanılmıştır.

```
{"Topics":["konular/konu1"], "Payload":"Merhaba Dünya"}\r\n
```

Çizelge 1. JTP protokolünün diğer araştırılan protokollerle karşılaştırılması

Kriterler	STOMP	XMPP	RESP	NATS	JTP
Çerçeve Boyutu	2 bayt	19 bayt	2 bayt	2 bayt	2 bayt
Çerçeve Kodlama	O(1)	O(1)	O(1)	O(1)	O(1)
Çerçeve Çözümleme	O(n)	O(n)	O(n)	O(n)	O(n)
Protokol Boyutu	60 bayt	520 bayt	22 bayt	22 bayt	41 bayt (14 bayt)
Protokol Kodlama	O(1)	O(1)	O(1)	O(1)	O(1)
Protokol Çözümleme	O(n)	O(n)	O(n)	O(n)	O(n)
İnsanlar Tarafından Kodlanabilme Değeri	6 veri tipi	9 veri tipi	4 veri tipi	3 veri tipi	10 veri tipi (7 veri tipi)

3.2 İşlevsel Özelliklere Göre Protokolün Testi

Protokoller, belirli sözdizimlerinin biraraya getirilmesiyle protokolün oluşturulma amacına uygun olarak birçok bildirim, komut ve direktif belirtimi yapılabilmektedir. Araştırılan protokollerin oluşturdukları komut setleri yardımıyla destekledikleri özellikler Çizelge 2’de verilmiştir ve JTP protokolünün işlevsel özellikleriyle karşılaştırılması yapılmıştır. STOMP protokolü, basit komutsal iletişim yapabilmek için tasarlandığından işlevsel özellik sayısı bakımından en az işlevsel özelliğe sahip olduğu görülmüştür. XMPP protokolünün genişletilebilir özelliği sayesinde birçok işlevsel özellik sonradan eklenmiştir ve yeni eklenen protokol ekleri standartlaştırılmıştır. RESP protokolü, Redis sunucunun özelliklerini kullanmak amacıyla tasarlanmıştır. NATS protokolü, yayımcı-abone tipi iletişim için tasarlanmıştır ancak bunun dışında Çizelge 2’de belirtilen işlevsel özelliklere de sahiptir. Ayrıca araştırılan protokoller arasında en az sayıda komut sayısına sahip olduğu görülmüştür.

Çizelge 2. JTP protokolünün işlevsellik yönünden diğer araştırılan protokollerle karşılaştırılması

İşlevsel Özellik	STOMP	XMPP	RESP	NATS	JTP
Gerçek zamanlı çok hedefli iletişim desteği	Var	Var	Var	Var	Var
Platform bağımsız veri iletimi	Var	Var	Var	Var	Var
Yayımcı-abone tipi iletişim desteği	Var	Var [6]	Var	Var	Var
Arayan-aranan tipi iletişim	Yok	Var [7]	Yok	Yok	Var

desteđi					
Yük dengeleyicili kuyruk desteđi	Yok	Var [8]	Var	Var	Var
Süre limitli veri iletimi desteđi	Yok	Yok	Var	Yok	Var
QoS (Quality of Service) iletim denetimi desteđi	Yok	Var [9]	Yok	Yok	Var
Kalp atışı desteđi	Var	Yok	Yok	Var	Var
Protokolün genişletilebilir olması	Yok	Var	Yok	Yok	Var
Verilerin gerçek zamanlı sıkıştırılarak iletimi	Yok	Var [10]	Yok	Yok	Var
Kimlik doğrulama desteđi	Yok	Var [11]	Var	Var	Var
İzleme desteđi	Yok	Yok	Var	Var	Var

4 Sonuç

JTP adında JSON sözdizimi ile formatlanmış aracı sunucu üzerinden gerçek zamanlı iletişim için kullanılan yeni bir protokol önerilmiştir. Bu protokolün kurumsal uygulamalarda, IoT projelerinde, gerçek zamanlı iletişim gereksinimini karşılamak ve tüm platformlar arasında iletişim kurmak için uygun olduğu gösterilmiştir. Araştırılan protokollerden işlevsel özellikleri bakımından daha fazla özelliğe sahip bir protokoldür. Ancak insanlar tarafından okunabilme ve kodlanabilme özelliđi bakımından iyi seviyede olmadığı değerlendirilmektedir. JTP protokolünün desteklemediđi ancak literatürde kabul görmüş iletişim tipleri araştırılarak JTP protokolünün geliştirilmesi sağlanabilir. Ayrıca JTP protokolünün kullandığı standart model JSON yerine çalışmanın kapsamı dışında olan daha hızlı kodlama ve çözümleme performansı sunan sözdizimleri kullanılırsa daha da performanslı hale getirilebilecektir. Aracı sunucu ile uç birimler arasında TLS protokolüyle veriler kriptolanarak daha güvenli bir iletişim gerçekleştirilebilecektir.

Referanslar

1. Namiot, D., Sneps-Snepe, M.: On Micro-services Architecture, International Journal of Open Information Technologies, 2(9), 2307-8162 (2014).
2. Introducing JSON, <https://www.json.org>, son erişim 2018/04/23.
3. Bianchi, G.: Performance analysis of the IEEE 802.11 distributed coordination function, IEEE Journal on Selected Areas in Communications, Vol: 18 (2000).
4. Sitaraman, M., Kulczycki, G., Krone, J., Ogden, W.F., Reddy, A.L.N.: Performance specification of software components, New York: ACM (2001).
5. Friedman, S.L., Klivington, K.A., Petersan, R.W.: The Brain, Cognition and Education, Cambridge: Academic Press Inc (1986).
6. Publish-Subscribe, <https://xmpp.org/extensions/xep-0060.html>, son erişim 2018/04/23.
7. Jingle Session Transfer, <https://xmpp.org/extensions/xep-0251.html>, son erişim 2018/04/23.
8. Workgroup Queues, <https://xmpp.org/extensions/xep-0142.html>, son erişim 2018/04/23.
9. Quality of Service, <https://xmpp.org/extensions/inbox/qos.html>, son erişim 2018/04/23.

10. Message Archiving, <https://xmpp.org/extensions/xep-0136.html>, son erişim 2018/04/23.
11. Stream Compression, <https://xmpp.org/extensions/xep-0138.html>, son erişim 2018/04/23.