

„Connection reset by peer“ (java.io.IOException) – Analyse eines systematischen TCP-Problems

Hintergrund

Die Ausnahme `java.io.IOException: Connection reset by peer` trat bereits auf, bevor ich dem Team beitrat. Auf den Grafana-Dashboards war deutlich zu erkennen, dass zahlreiche Dienste bzw. deren Client-Komponenten diese Ausnahme regelmäßig auswarfen. Das Muster war auffällig:

- gehäufte Ausnahmen etwa alle **30 Minuten**
- deutliche Zunahme nach **Leerlaufphasen**

Dank Retry-Mechanismen entstand zwar keine Geschäftsunterbrechung, dennoch stellte das Verhalten ein operatives **Risiko** dar und führte zu erheblichem **Log-Rauschen** im Monitoring.

Da ich durch meine technische Ausbildung mit Netzwerk- und TCP/IP-Mechanismen vertraut bin, bat mich mein Teamleiter um eine strukturierte Analyse.

Analyse

Die Fehlermeldung deutete bereits auf eine **serverseitig abgebrochene TCP-Verbindung** hin. Zwei Beobachtungen unterstützten diese Annahme:

- Anstieg nach Inaktivität → typisch für **Idle-Timeouts**
- Einsatz von **Connection Pools** → RST beim Wiederverwenden eines „toten“ Sockets ist ein bekanntes Muster

Aus meiner Sicht lag die Ursache vermutlich im Infrastruktur-Layer, insbesondere bei Idle-Eviction-Mechanismen eines Load Balancers. Das Ops-Team war jedoch zunächst zurückhaltend, da:

- C#-basierte Services dieses Verhalten nie gezeigt hatten
- Java-Services erst später migrierten und daher im Verdacht standen
- „Connection reset by peer“ häufig fälschlich als Anwendungsfehler interpretiert wird

Fehlerhafte Client-Implementierungen schloss ich nicht vollständig aus — aber wenn der Server eine Verbindung ohne **FIN** beendet, führt das zwangsläufig zu einem **RST**. Damit war klar: **Ich brauchte Beweise auf TCP-Ebene**.

Maßnahmen

Mit gezielten Paketmitschnitten in Wireshark analysierte ich den Datenverkehr. Die Ergebnisse bestätigten die Vermutung eindeutig:

Nach rund 30 Minuten Inaktivität wurden Verbindungen im Connection Pool serverseitig beendet, jedoch ohne das übliche FIN-Handshake. Der Client interpretierte dies korrekt als „Connection reset by peer“.

Damit ließ sich dem Ops-Team eine **klare, technisch fundierte Argumentation** vorlegen:

- Serverseitiges Beenden der Verbindung **ohne FIN**
- Verhalten eindeutig passend zu einer **Idle-Eviction-Konfiguration** eines Load Balancers

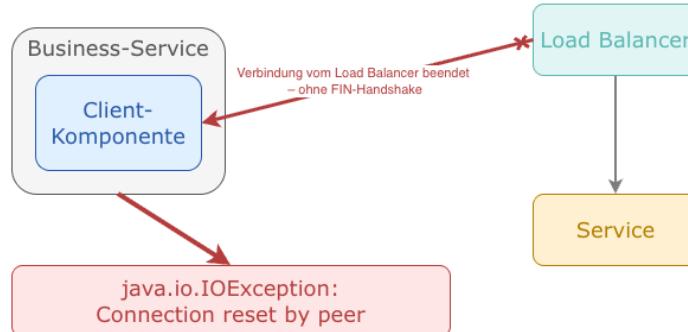


Abbildung 1: Serverseitiger Verbindungsabbruch ohne FIN-Handshake führt zur Ausnahme

Die Load-Balancer-Konfiguration wurde daraufhin überprüft und die Ursache bestätigt: Eine fehlerhafte **Eviction-Konfiguration** führte zu systematischen Verbindungsabbrüchen.

Nach Korrektur der Einstellung trat das Problem nicht mehr auf.

Ergebnisse

- Diese **seit Monaten auftretenden** Ausnahmen `java.io.IOException: Connection reset by peer` verschwanden vollständig
- Das Risiko versteckter Infrastrukturprobleme wurde nachhaltig reduziert und das störende Log-Rauschen auf den Monitoring-Dashboards vollständig beseitigt.
- Die Zusammenarbeit mit dem Ops-Team verbesserte sich spürbar durch das entstandene gemeinsame technische Verständnis