HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

A Real-time Streaming Protocol for Large-scale Peer-to-Peer Networks

Christopher Probst

Institut für Informatik
Rechnernetze und Kommunikationssysteme
Heinrich-Heine-Universität Düsseldorf

3.12.2014

Gliederung

1. Einleitung
2. Umsetzung
3. Evaluation

Gliederung

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Motivation

- Problem: Datenverbreitung mit Client / Server Architektur skaliert nicht
  - Jeder Client erhöht die Auslastung
  - Uploadbandbreite des Servers ist limitiert
  - Uploadbandbreite der Clients ungenutzt
- Lösungsansatz: Verwendung eines Peer-to-Peer Netzwerks
  - Jeder Peer hilft bei der Datenverbreitung
  - Der Super-Peer (Peer mit vollständigem Datensatz) wird nicht stärker ausgelastet als andere Peers
  - Ein Super-Peer mit geringer Uploadbandbreite kann dennoch schnell Daten verbreiten

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Zielsetzung

Implementierung einer Peer-to-Peer Anwendung zur termingerechten Verbreitung von Daten:

- Uploadbandbreite der Peers effizient nutzen
- Zeitpunkt für den Erhalt der Daten bei allen Peers gleich
- Gesamtdauer unabhängig von der Anzahl der Peers
- Gesamtdauer kleiner als $2 * T_0$.

Gliederung

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ansatz: Chunked-Swarm

Super-Peer teilt die Daten vor dem Transfer in kleine Teile (Chunks):

- Mindestens so viele Chunks wie Peers
- Peers beantragen disjunkte Chunks vom Super-Peer
- Peers tauschen Chunks untereinander aus
- Vereinfachung: Alle Peers haben die gleiche Uploadbandbreite

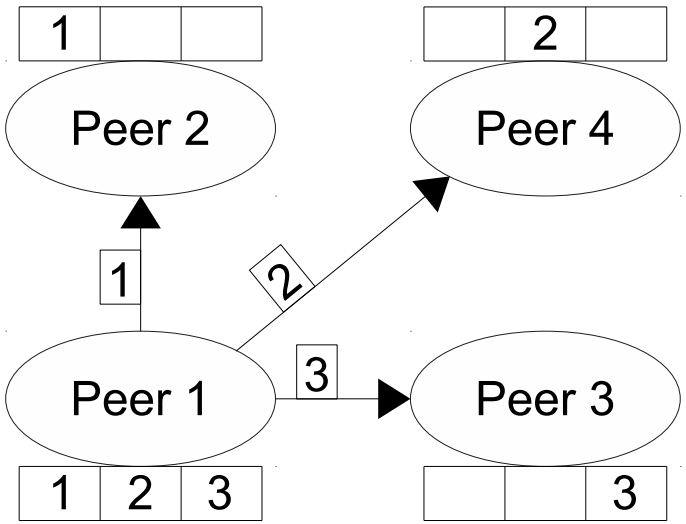
HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

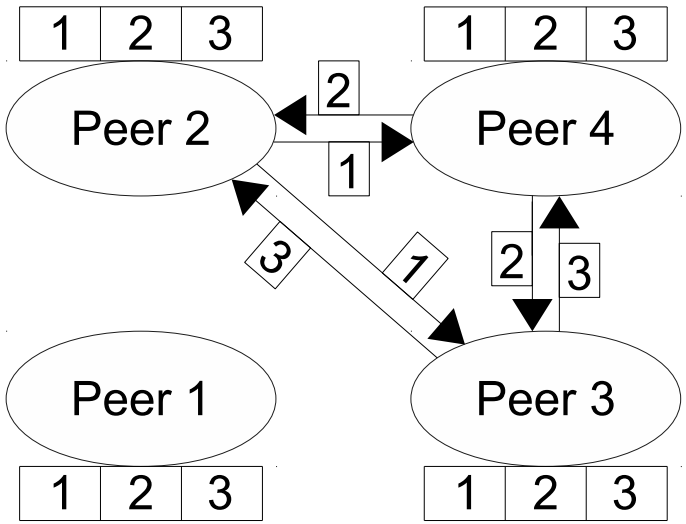
Chunked-Swarm: Ablauf

Beispiel: Genauso viele Chunks wie Peers

Phasen

- Phase 1 (links): Peers bekommen disjunkten Chunk vom Super-Peer
- Phase 2 (rechts): Peers tauschen ihre Chunks untereinander





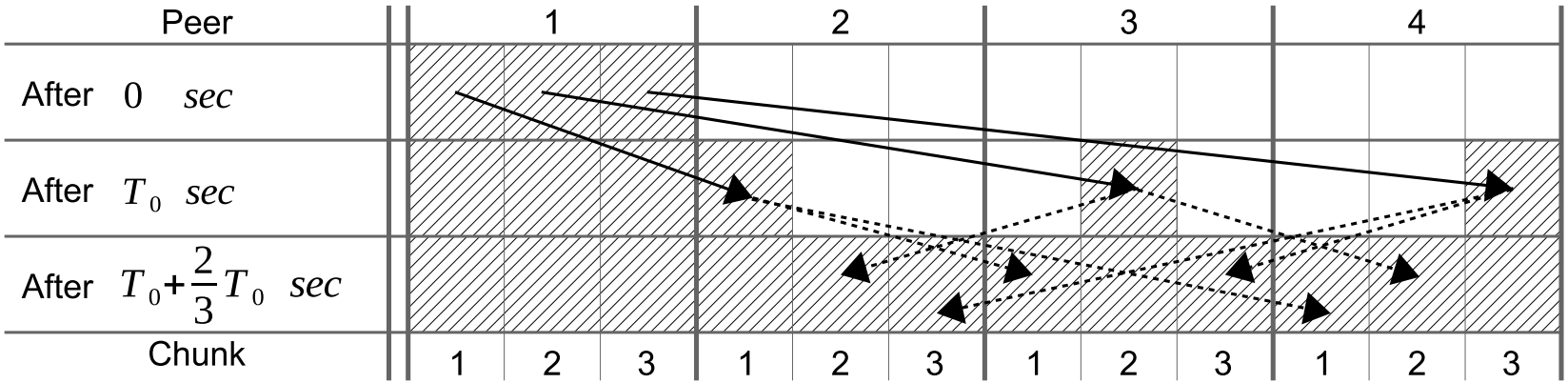
HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Chunked-Swarm: Gesamtdauer

Beispiel: Genauso viele Chunks wie Peers

Zeitlicher Ablauf

- Bis T_0: Super-Peer 1 schickt disjunkten Chunk an jeden Peer
- Ab T_0: Jeder Peer schickt Chunk an die anderen beiden Peers
- Ab $T_0 + \frac{2}{3} * T_0$: Daten wurden vollständig verbreitet



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Chunked-Swarm: Gesamtdauer

Eigenschaften

- Verdoppelung der Chunks halbiert die Zeit zwischen T_0 und Ende
- Gesamtdauer: $T(n,c) = T_0 + \frac{n}{c} * \frac{n-1}{n} * T_0, n \in \mathbb{N}_1, c = n * 2^i, i \in \mathbb{N}_0$
- Real-time: Gesamtdauer immer unter $2 * T_0$

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Implementierung

- Mesh Topologie: Alle Peers sind miteinander verbunden
- Pull-Based: Chunks werden nur auf Wunsch übertragen
- Announcements: Jeder Peer kündigt seine Chunks an
- Automatic (Re-)Connect: Peers finden andere Peers durch Super-Peer

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Implementierung

- Dead Peer Detection: Super-Peer verschickt Chunks erneut
- Traffic-Shaping: Up-/Downloadbandbreite drosselbar
- Implementiert in Java mit Netty5

Gliederung

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Methodik

- Verschiedene Szenarios
- Ein Default-Szenario: Übrige Szenarios sind Abwandlungen
- Jedes Szenario läuft 10 mal
- Plots: Durchschnitt und Konfidenzintervall (Konfidenzniveau 95%)

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Methodik

Besonderheiten

- Verbindungen nur virtuell: Kein TCP
- Keine (De-)Serialisierung von Paketen: Spart CPU und RAM
- Aber: Paketgröße wird simuliert!

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Default Szenario

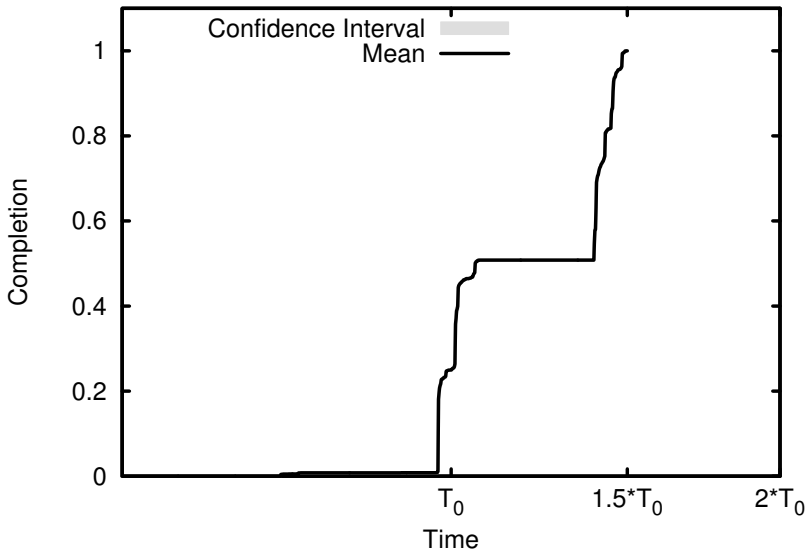
Einstellungen

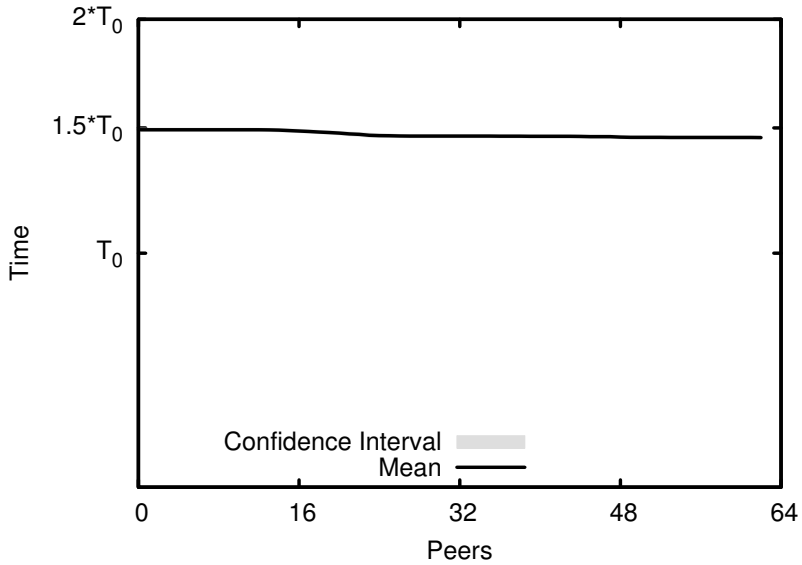
- 1 Super-Peer und 63 Peers
- Doppelt so viele Chunks wie Peers (126 Chunks)
- Gleiche Uploadbandbreite für Super-Peer und Peers
- Datengröße so gewählt, dass $T_0 = 10$ Minuten gilt

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Default Szenario - Completion

- Links: Ablauf des Datentransfers
- Rechts: Peers absteigend sortiert nach Gesamtdauer

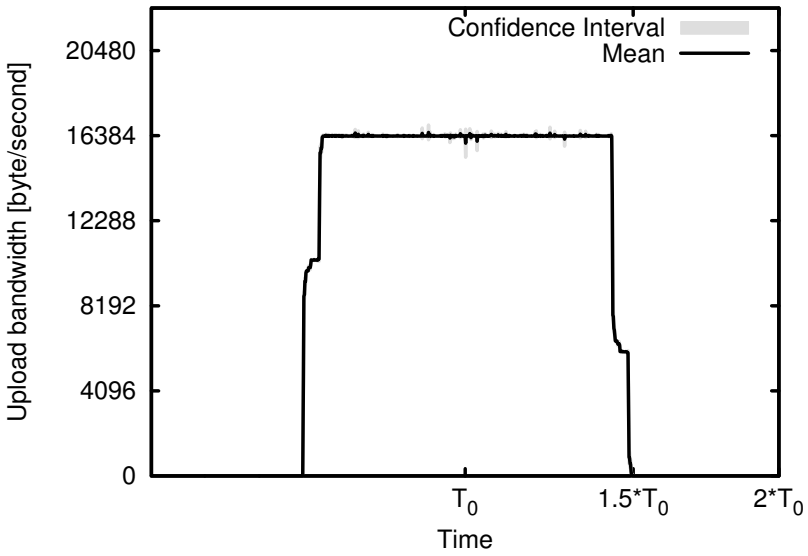


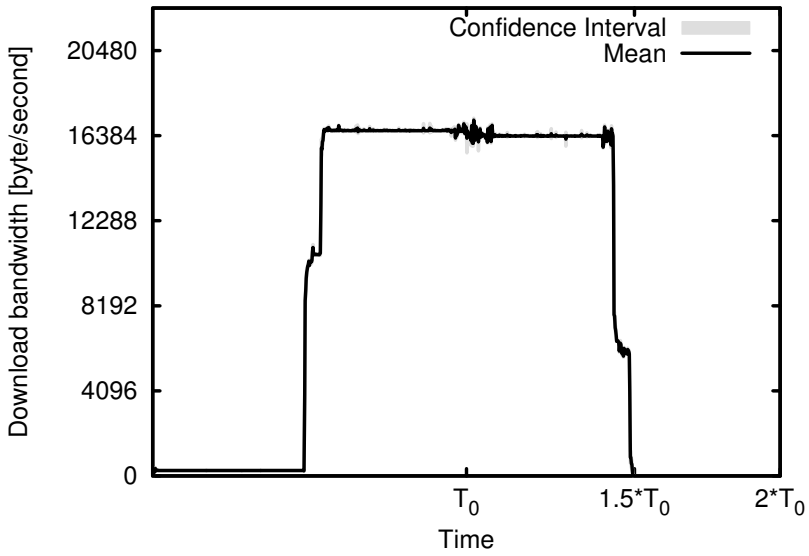


HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Default Szenario - Upload/Download

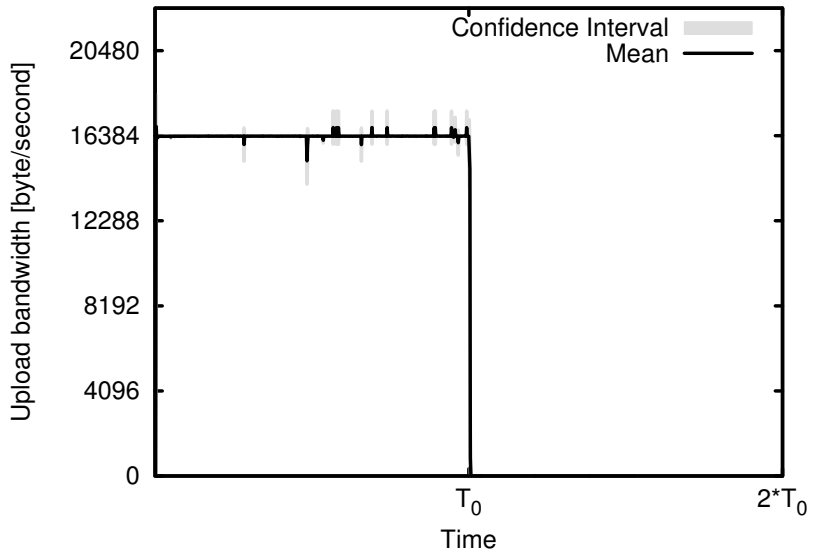
- Links: Uploadbandbreite
- Rechts: Downloadbandbreite





HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

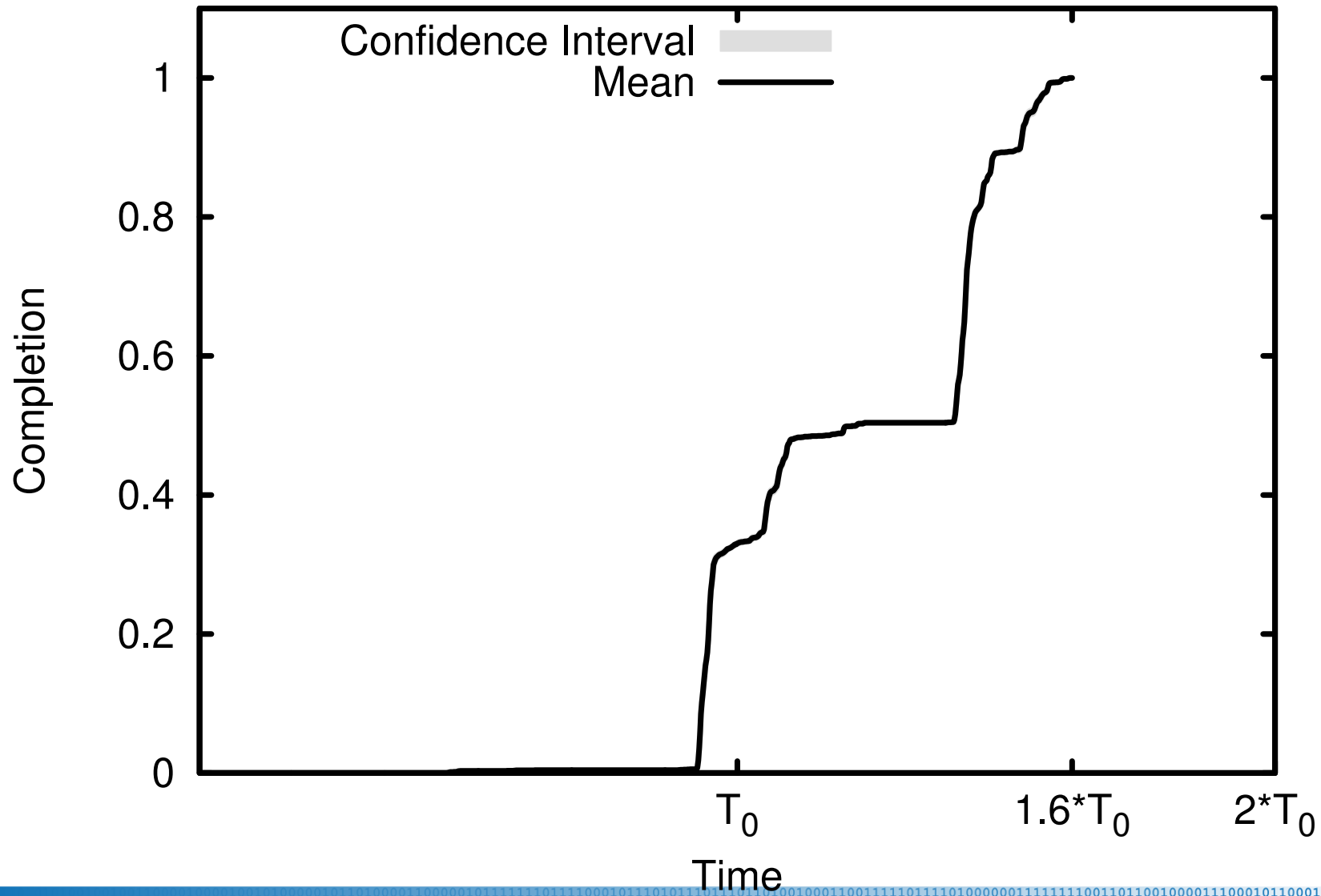
Default Szenario - Super-Peer Upload



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Default Szenario mit 128 Peers

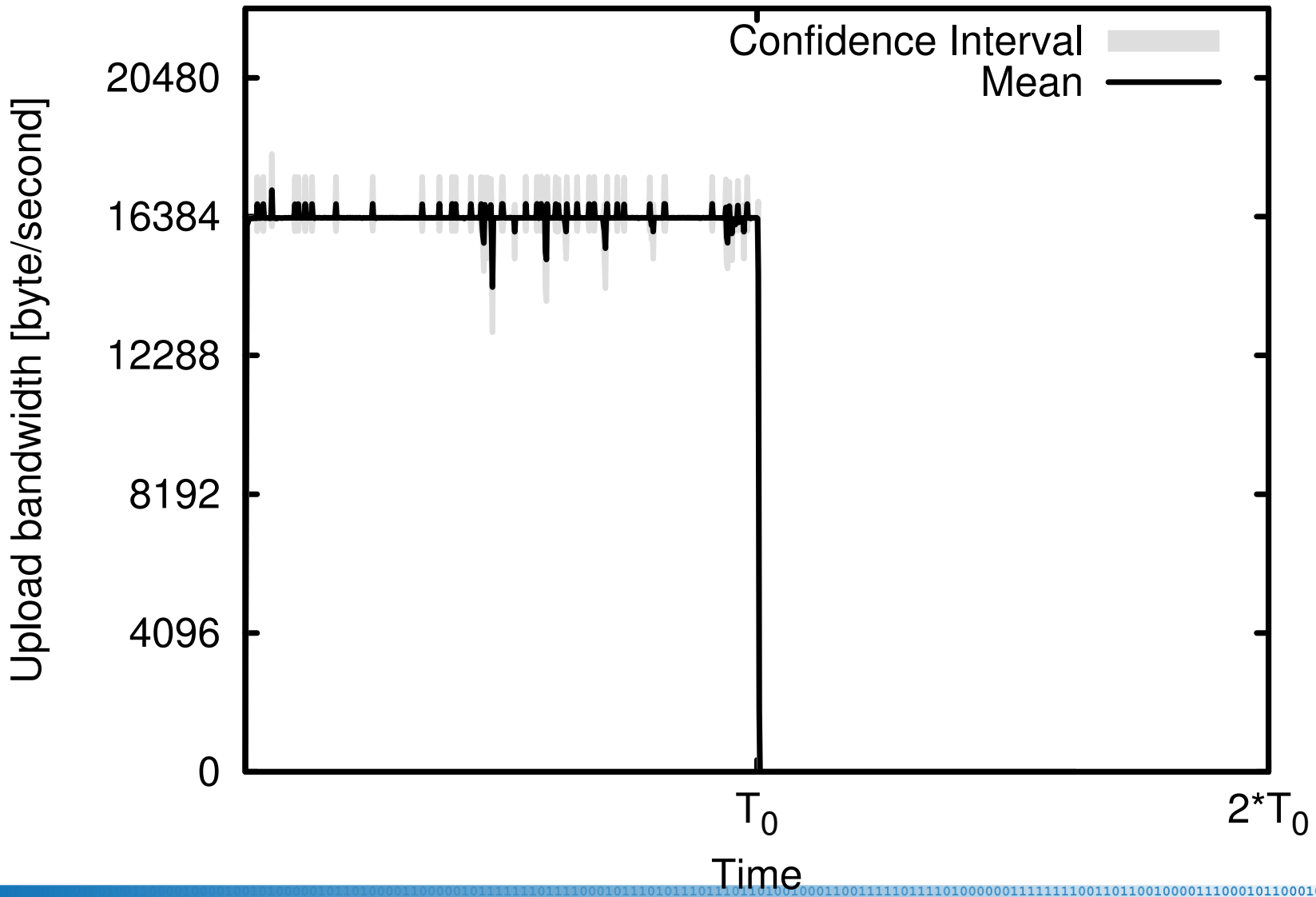
Completion Graph:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Default Szenario mit 128 Peers

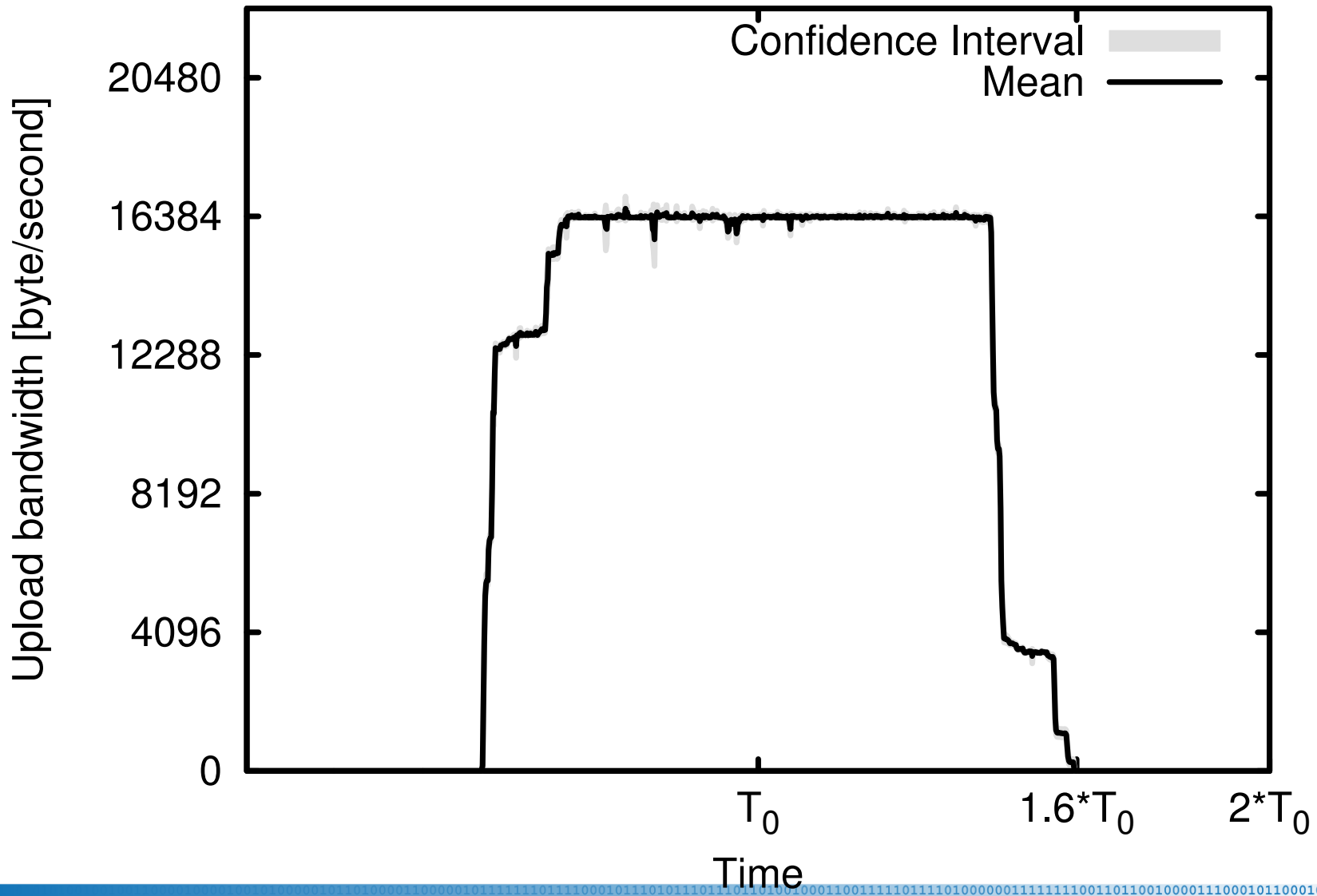
Super-Peer Upload Bandwidth:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Default Szenario mit 128 Peers

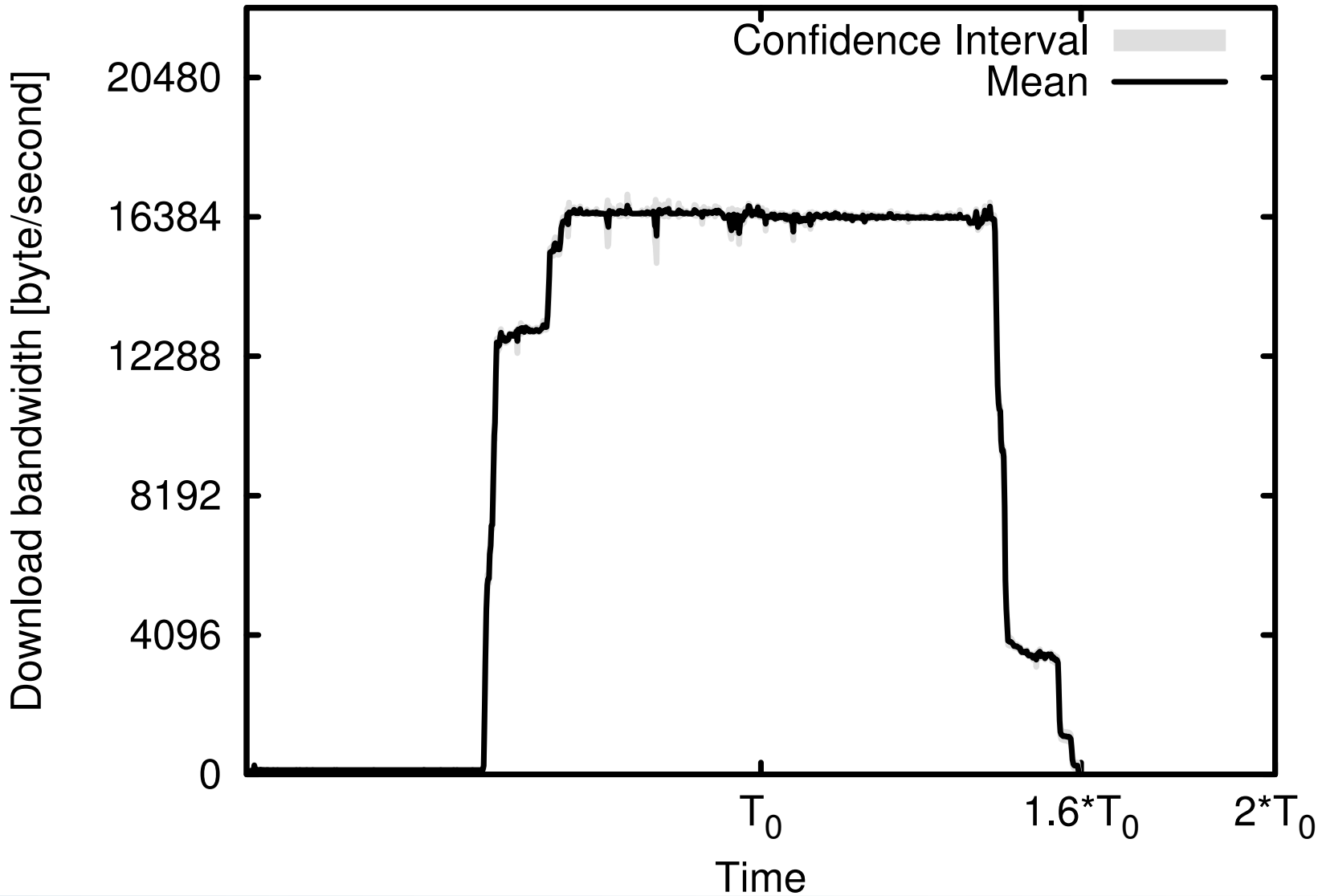
Peer Upload Bandwidth:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Default Szenario mit 128 Peers

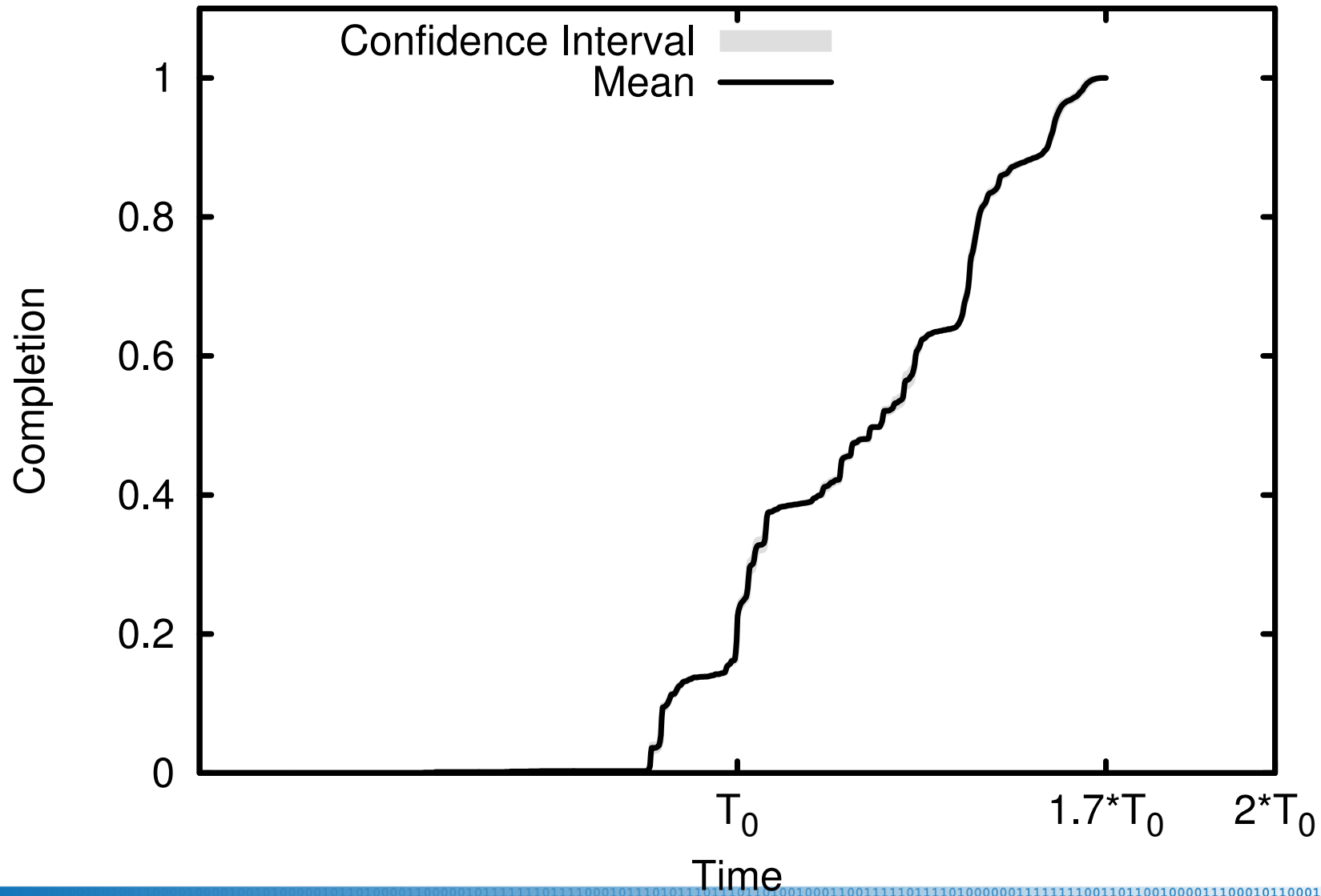
Peer Download Bandwidth:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Default Szenario mit 192 Peers

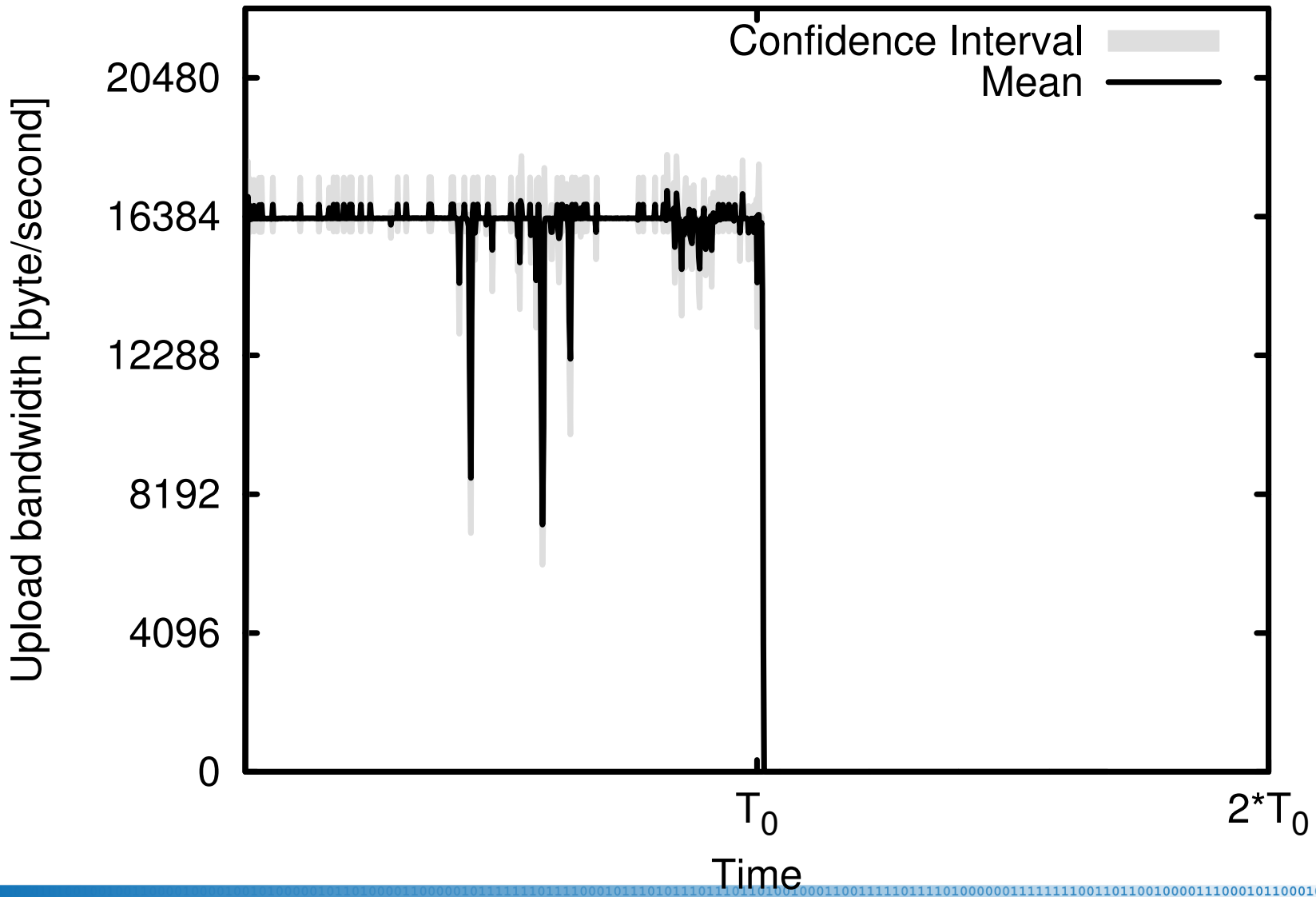
Completion Graph:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Default Szenario mit 192 Peers

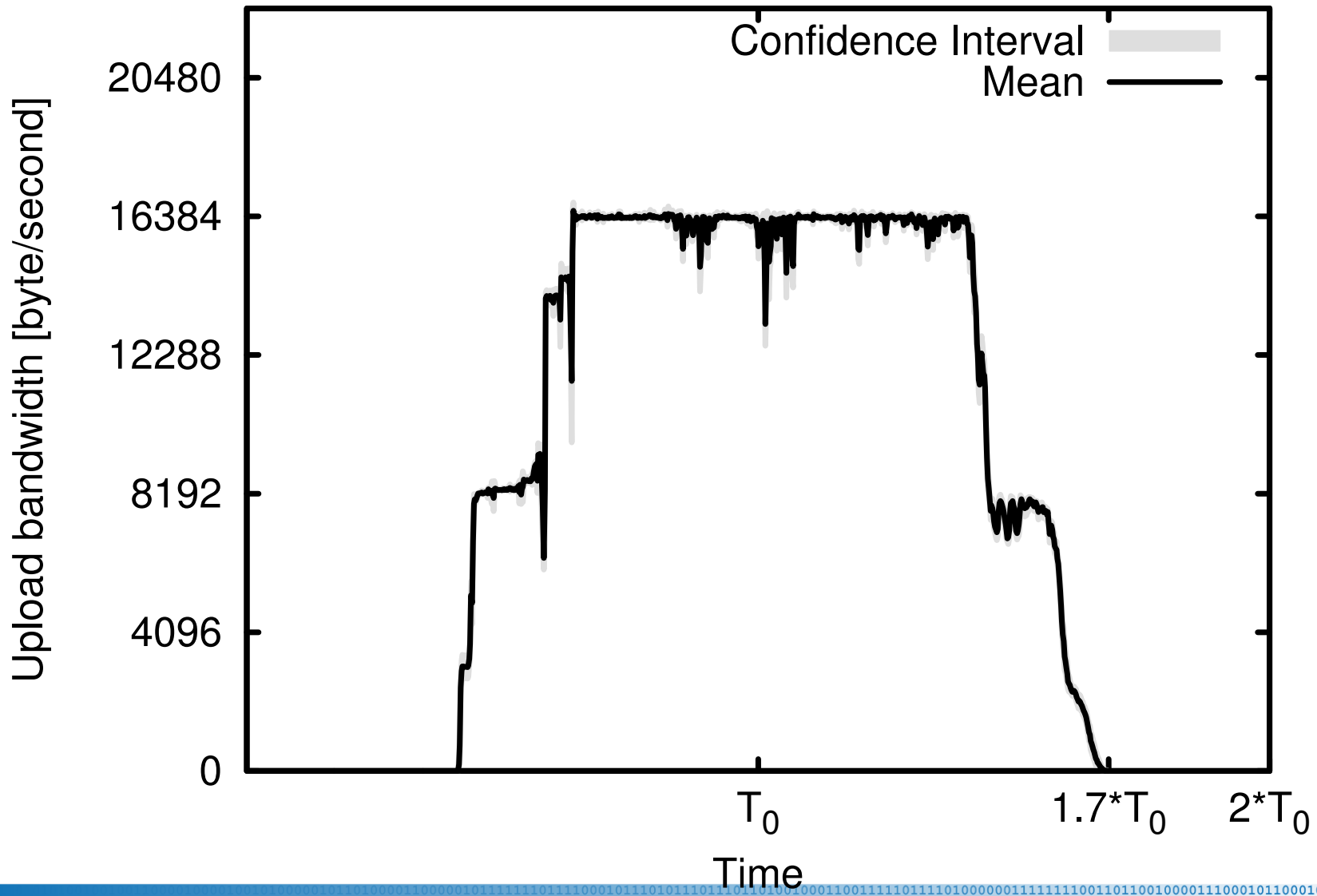
Super-Peer Upload Bandwidth:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Default Szenario mit 192 Peers

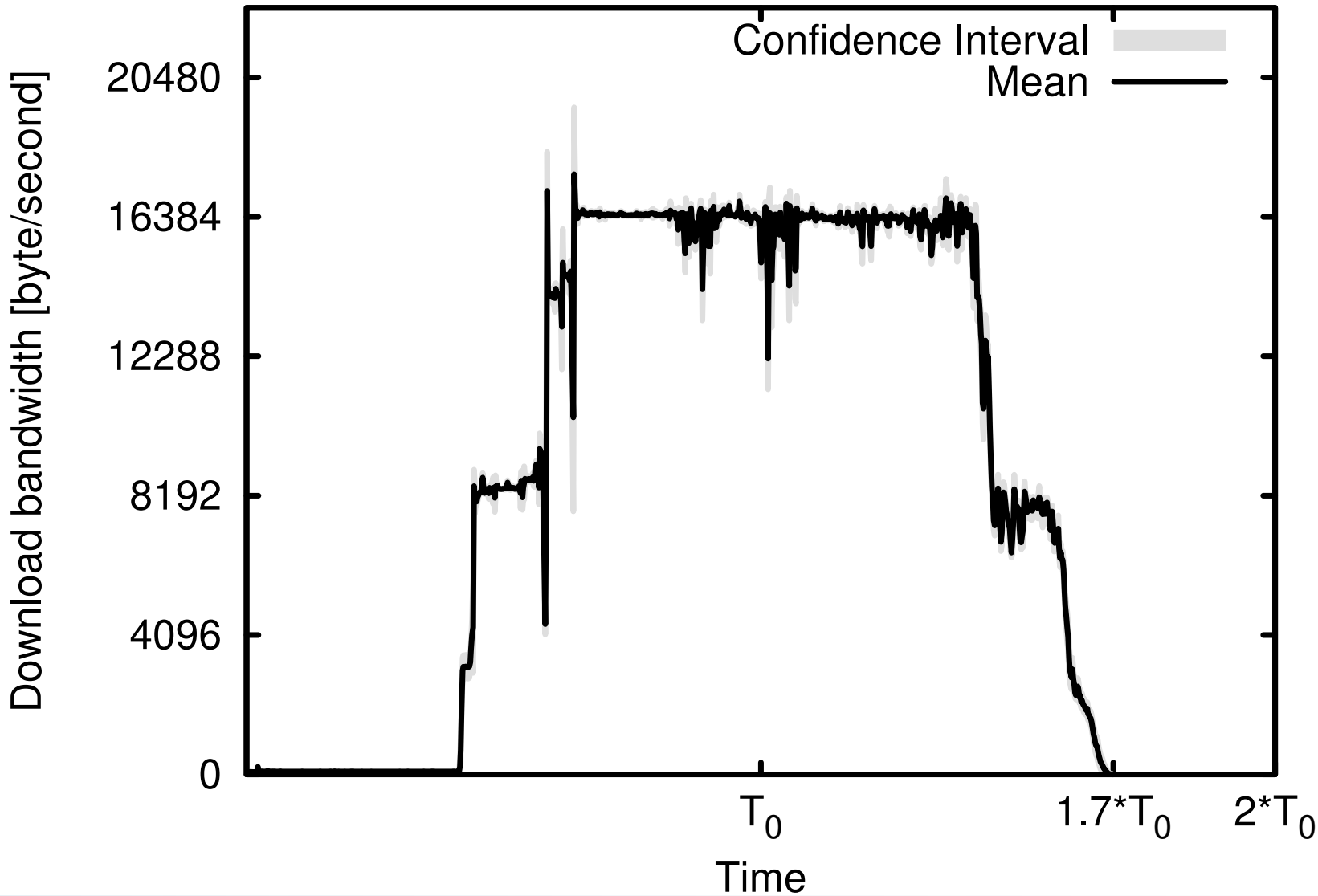
Peer Upload Bandwidth:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Default Szenario mit 192 Peers

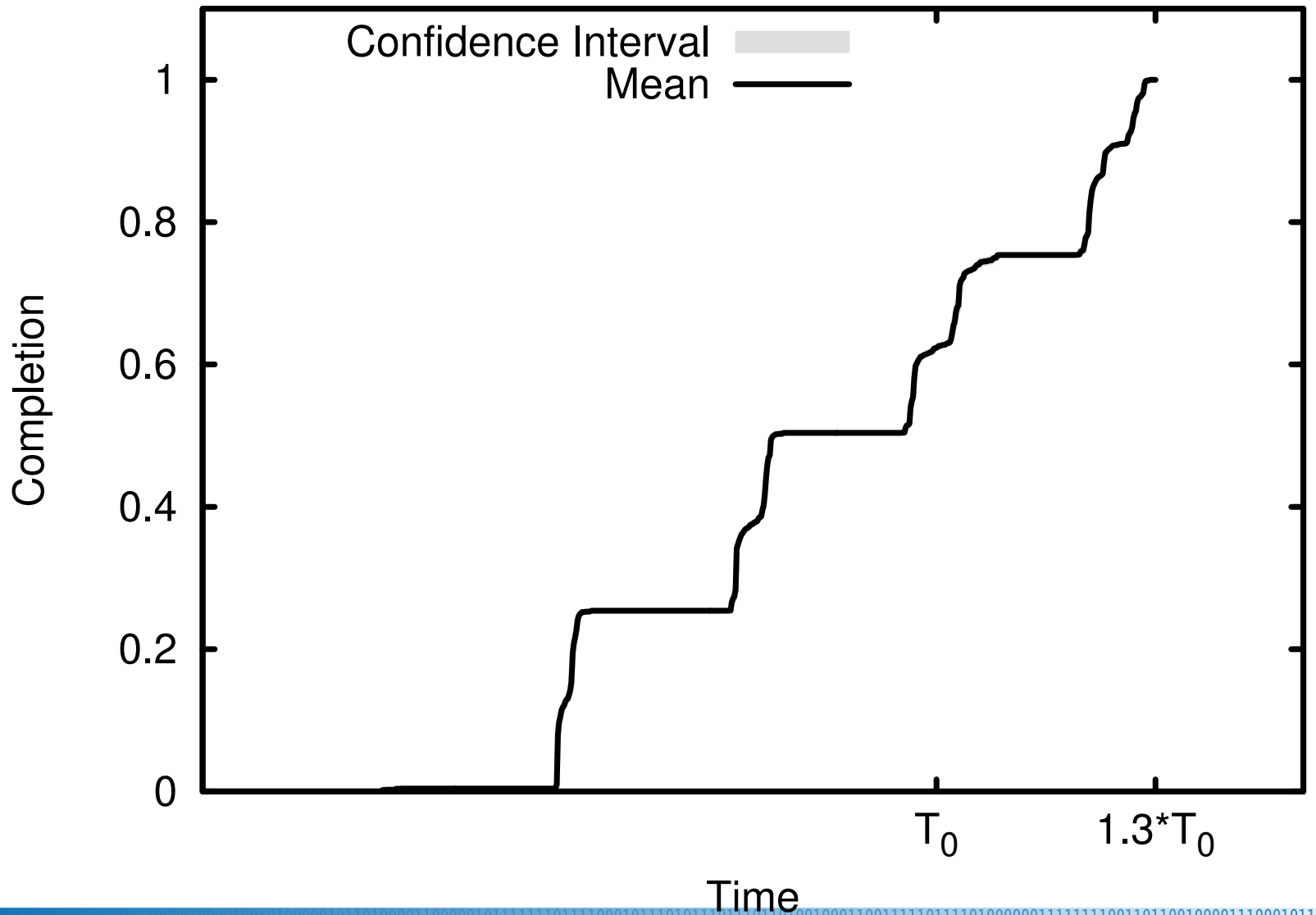
Peer Download Bandwidth:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Default Szenario mit 4x Chunkanzahl

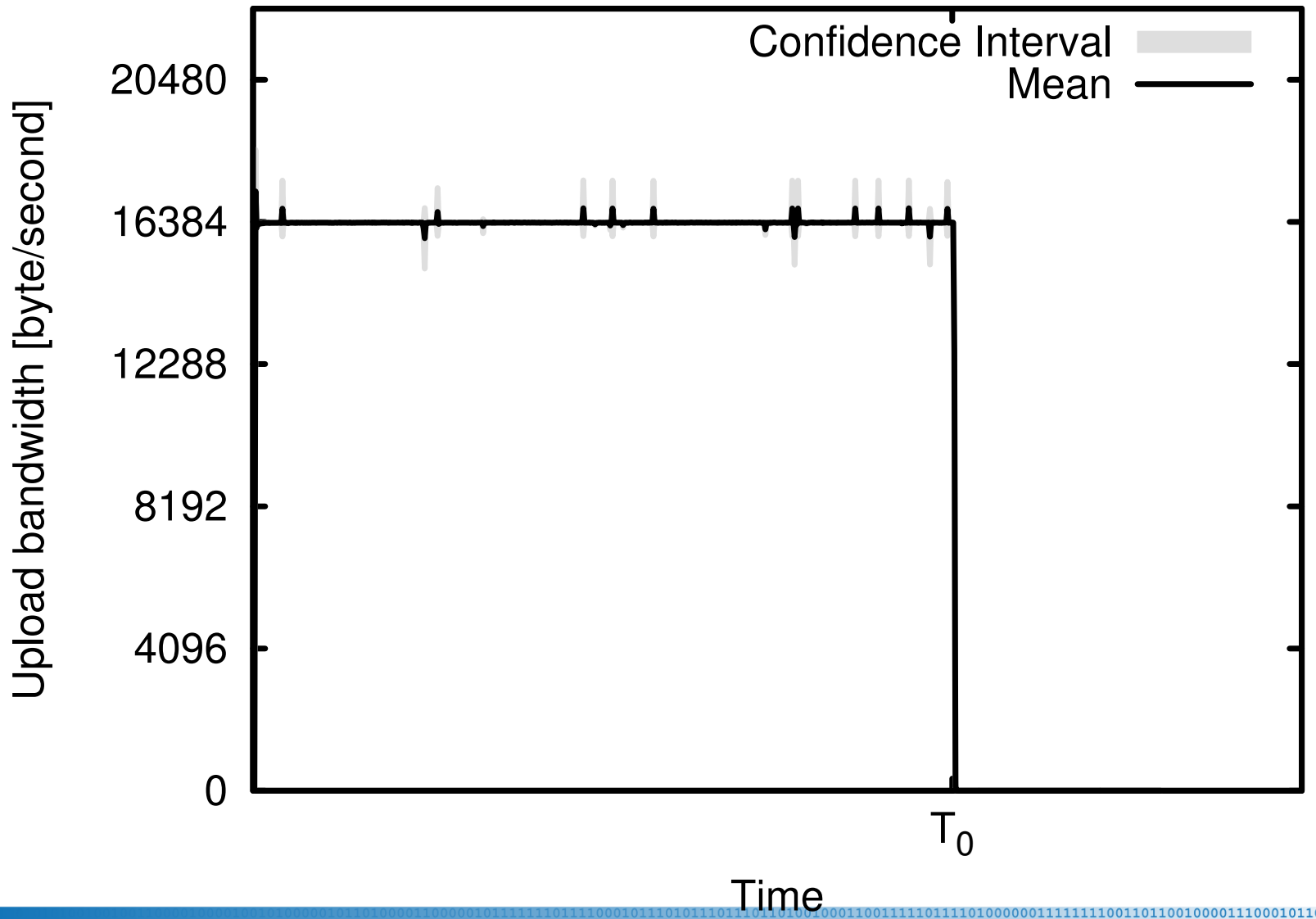
Completion Graph:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Default Szenario mit 4x Chunkanzahl

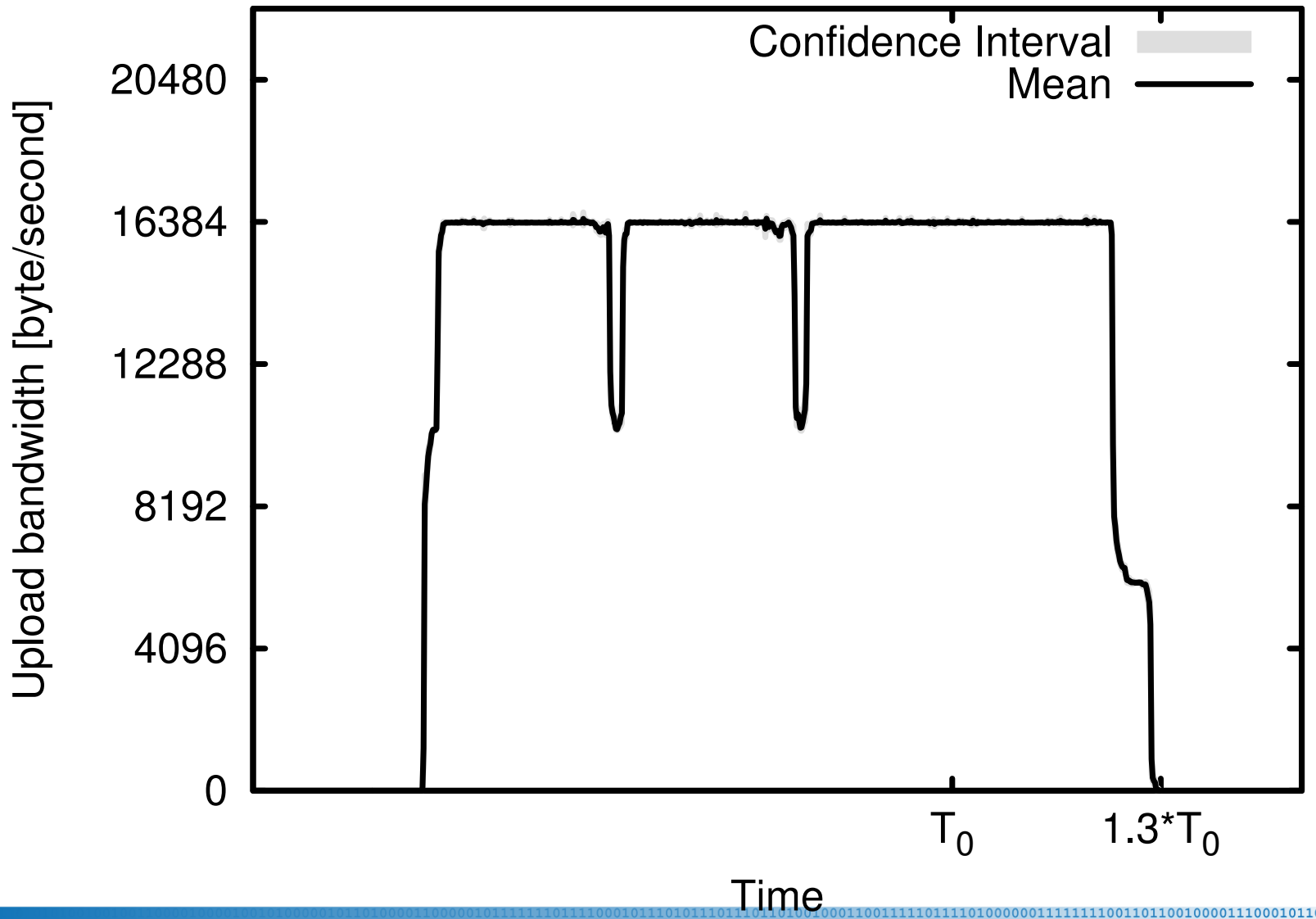
Super-Peer Upload Bandwidth:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Default Szenario mit 4x Chunkanzahl

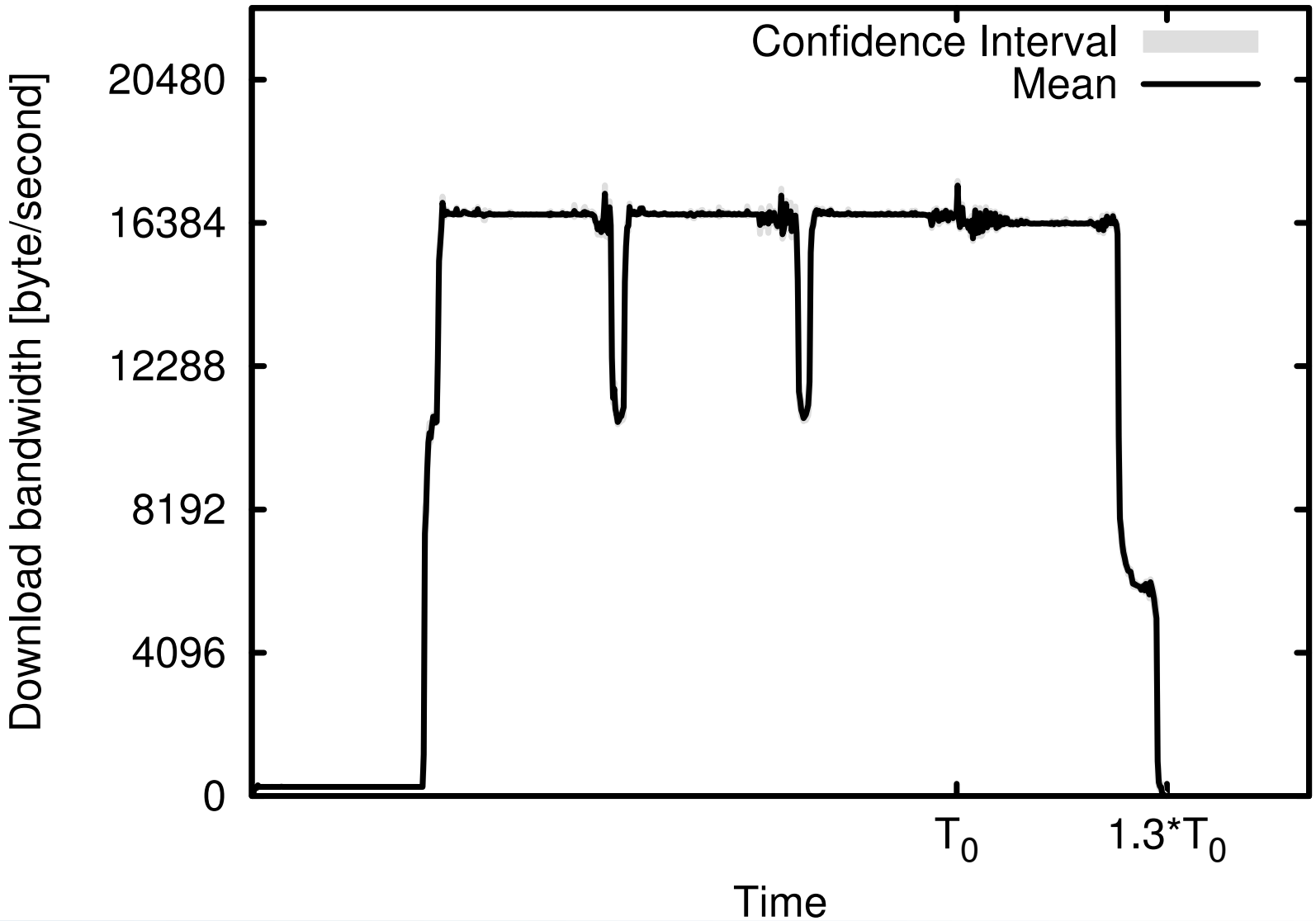
Peer Upload Bandwidth:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Default Szenario mit 4x Chunkanzahl

Peer Download Bandwidth:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Default Szenario mit 20 Datensätzen (Stream)

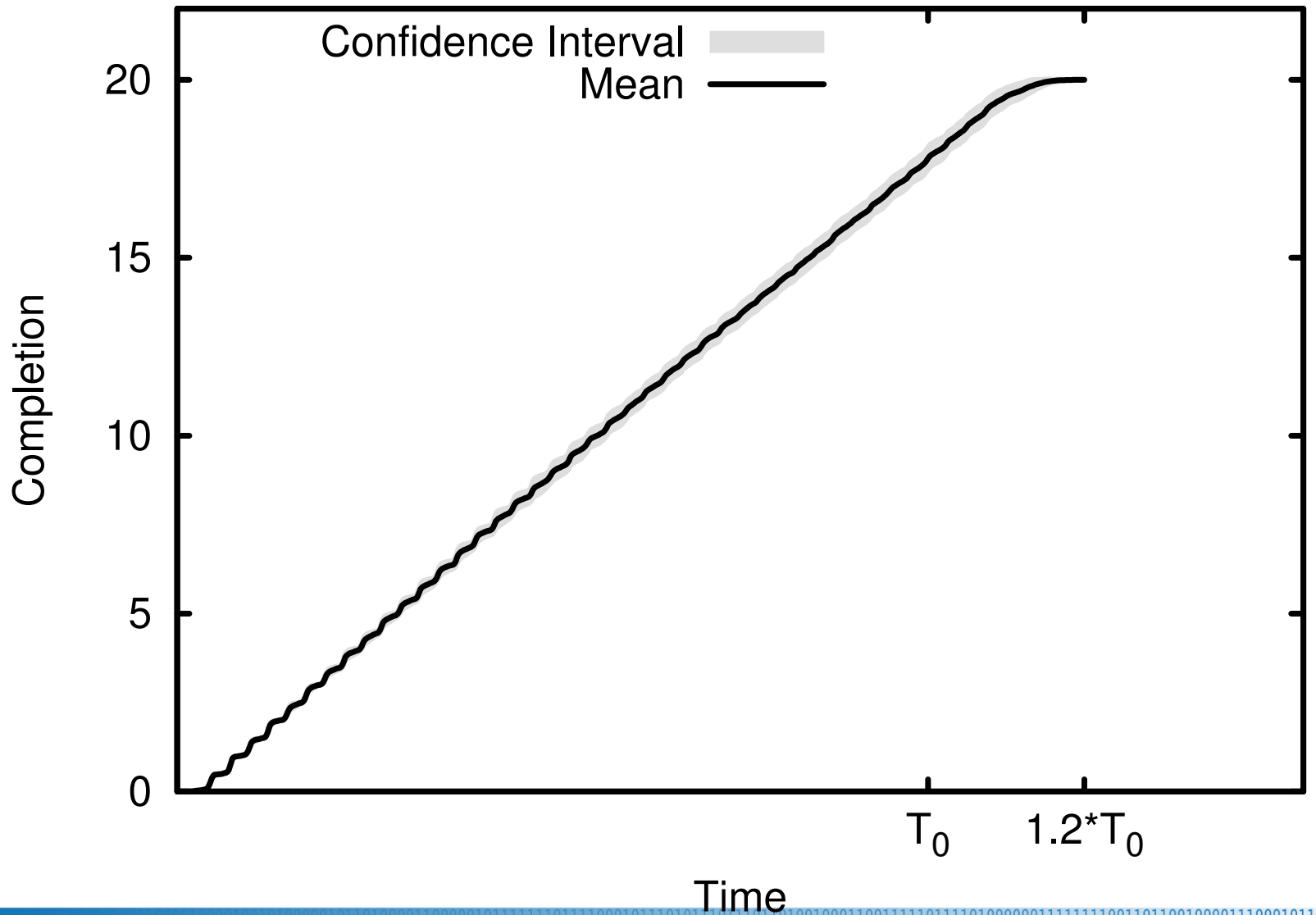
Bislang gab es immer nur einen Datensatz, der verteilt wurde. Nun wird der gesamte Datensatz in 20 weitere Sub-Datensätze geteilt.

- Jeder Sub-Datensatz hat wieder doppelt soviele Chunks wie Peers.
- Die Sub-Datensätze sind durchnummeriert mit IDs.
- Sub-Datensätze mit kleiner ID haben Vorrang.
- Streaming!

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Ergebnisse - Default Szenario mit 20 Datensätzen (Stream)

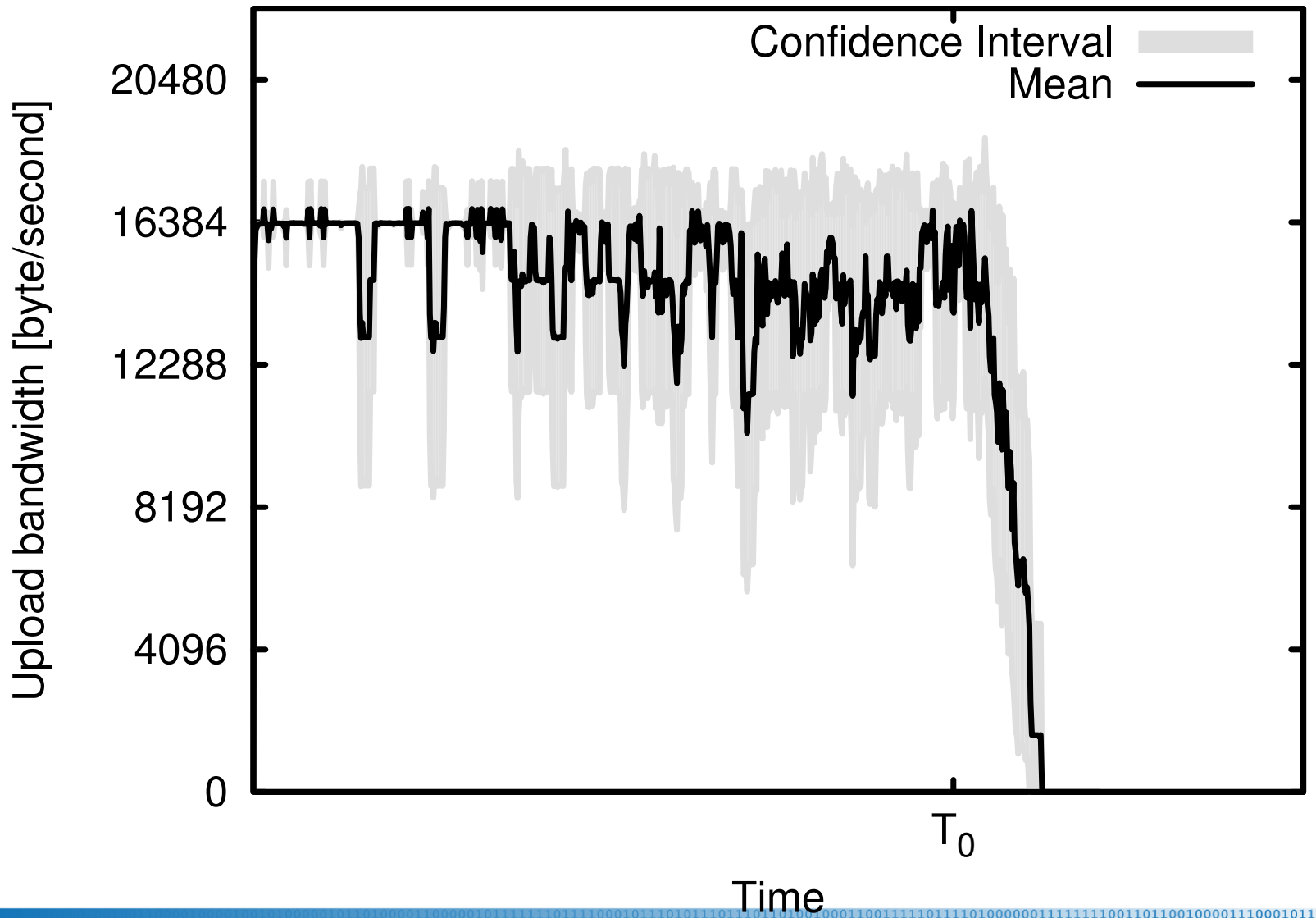
Completion Graph:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Default Szenario mit 20 Datensätzen (Stream)

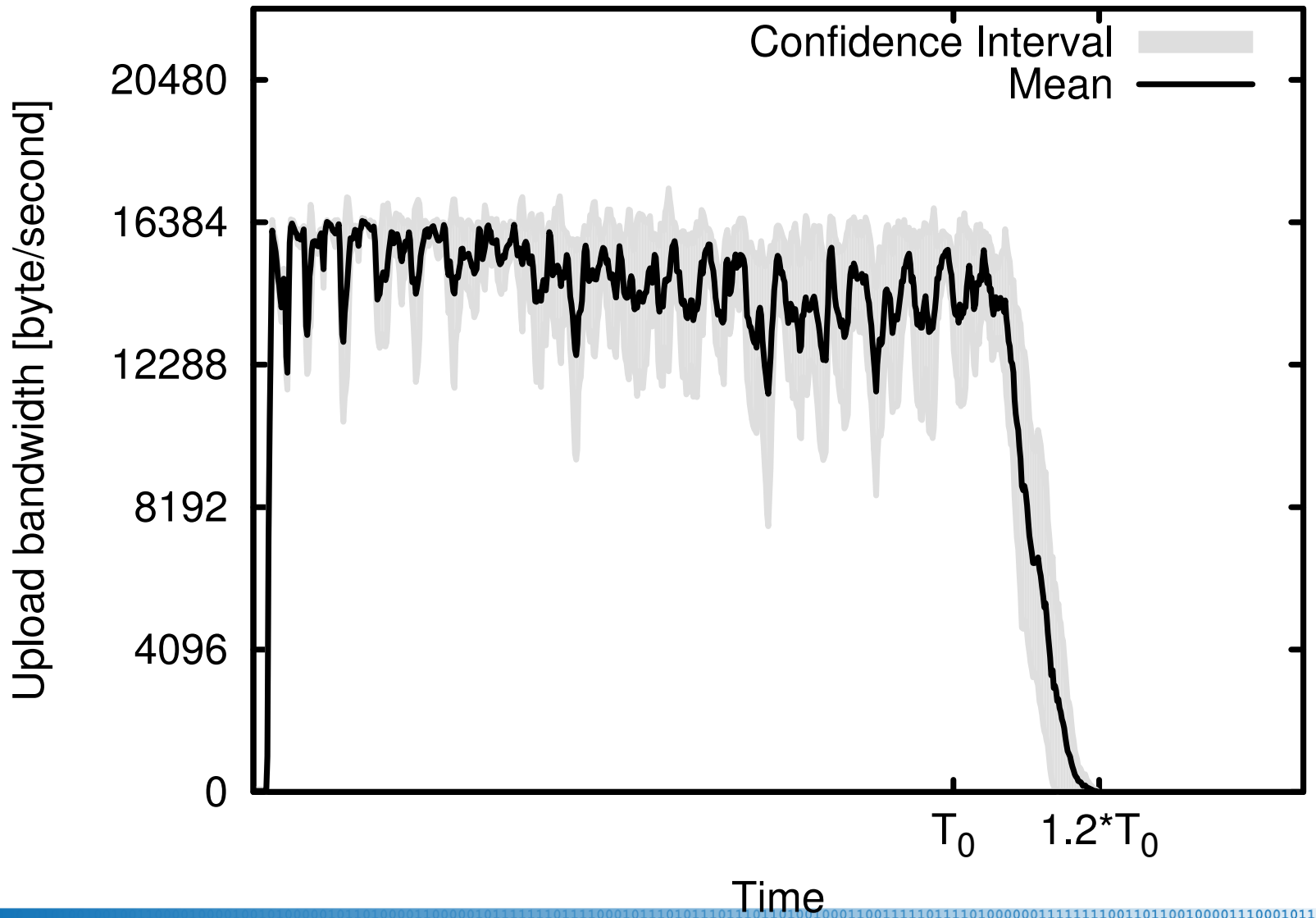
Super-Peer Upload Bandwidth:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Default Szenario mit 20 Datensätzen (Stream)

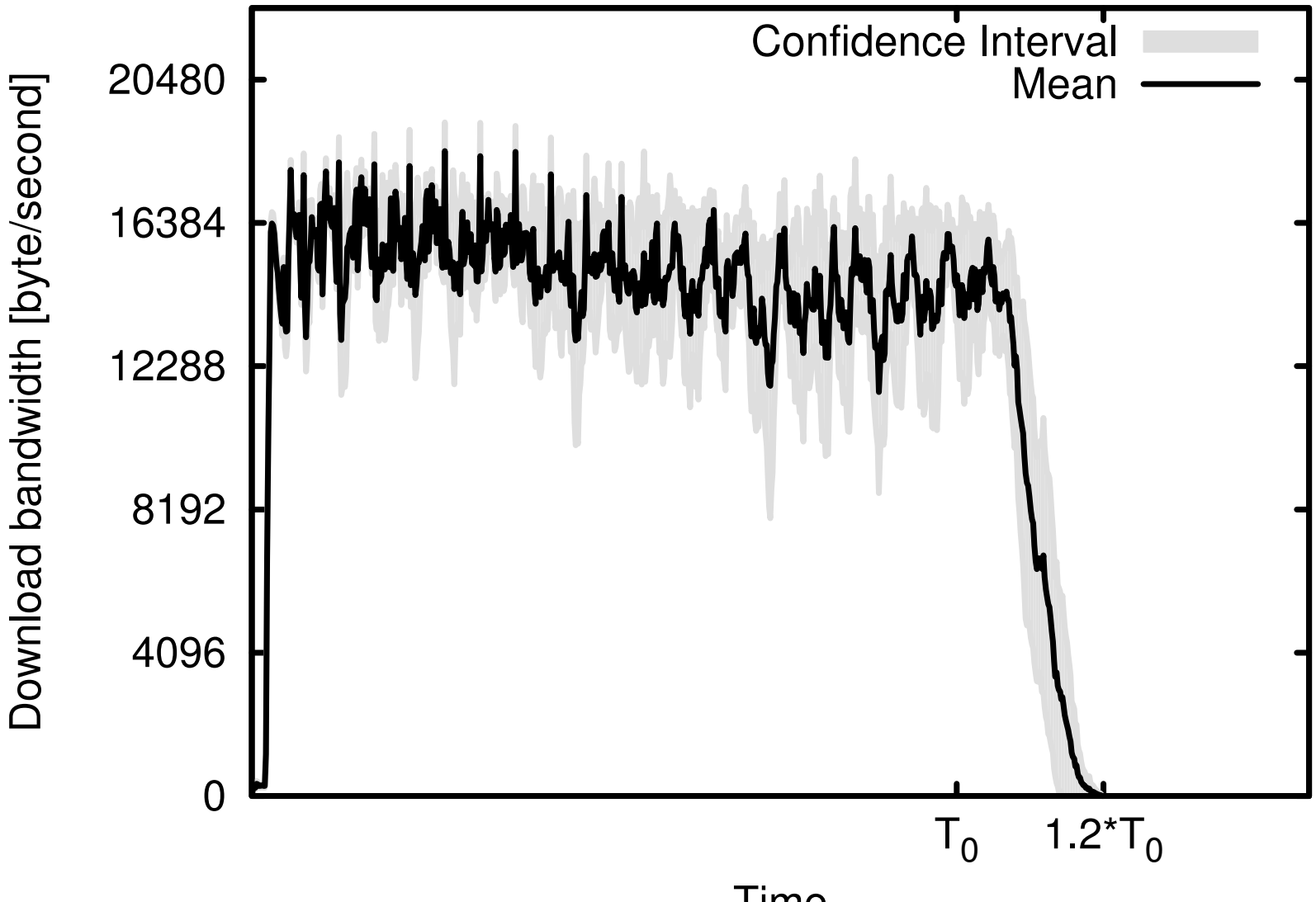
Peer Upload Bandwidth:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Default Szenario mit 20 Datensätzen (Stream)

Peer Download Bandwidth:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Szenario Sequential

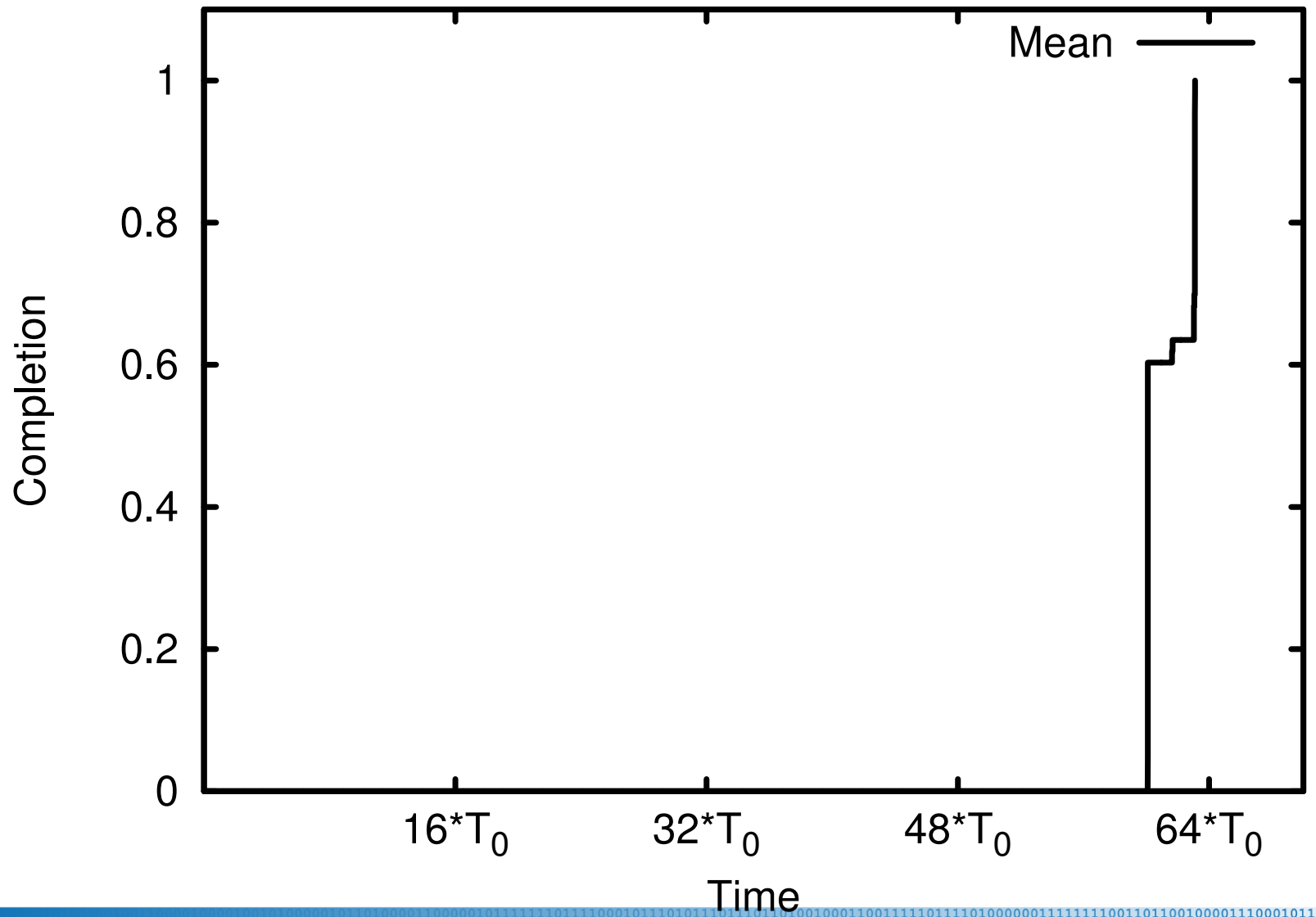
Dieses Szenario implementiert zum Vergleich ein Client / Server System:

- Es gibt einen Super-Peer (Server) und 63 Peers (Clienten).
- Die Peers sind nicht untereinander verbunden.
- Anzahl der Chunks spielt hier keine Rolle.
- Datengröße so gewählt, dass $T_0 = 10$ Minuten gilt.

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Szenario Sequential

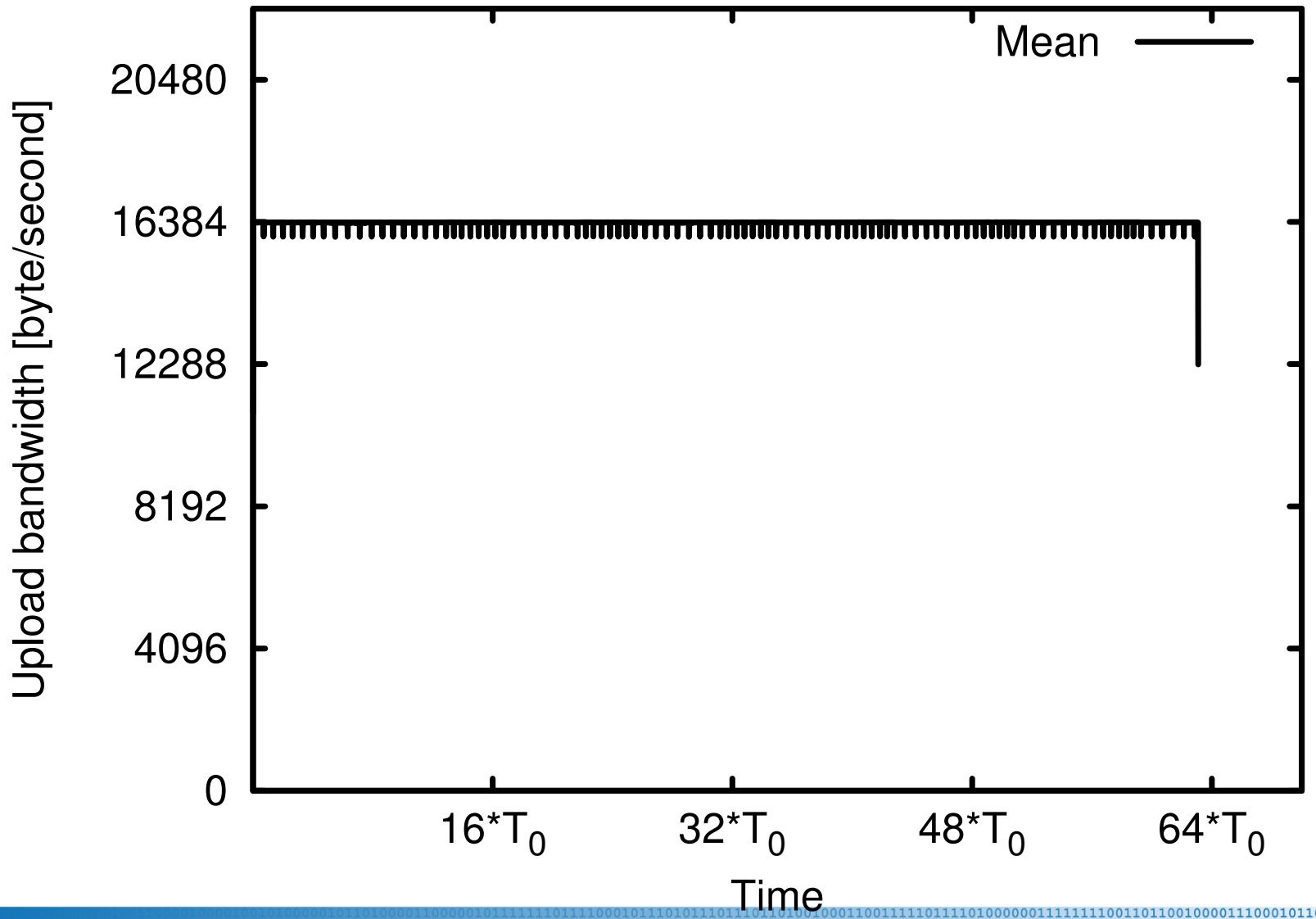
Completion Graph:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Szenario Sequential

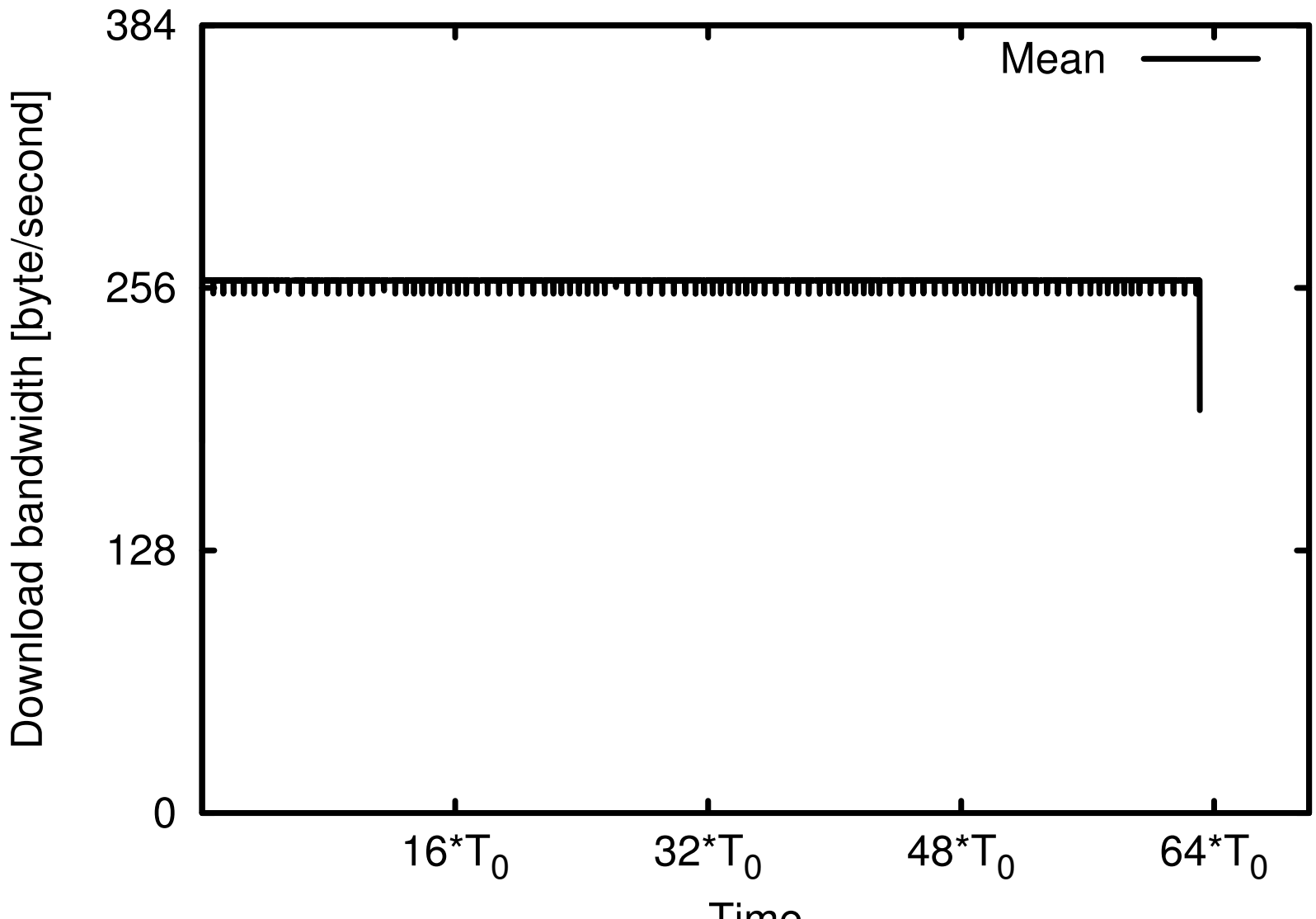
Super-Peer Upload Bandwidth:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Ergebnisse - Szenario Sequential

Peer Download Bandwidth:



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Evaluation - Fazit

- Mit Hilfe des Chunked-Swarm Verfahren kann man im Idealfall Lieferfristen weit unter $2 * T_0$ garantieren, unabhängig von der Anzahl der Peers.
- Die Anzahl der Peers sollte bekannt sein, um eine passende Anzahl an Chunks zu wählen.
- Das System skaliert nicht endlos, da die Anzahl der Verbindungen quadratisch wächst.
- Mit Hilfe von Sub-Datensätzen kann Streaming implementiert werden.

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Future Work

- Einführung einer Simulationszeit, damit bei vielen Peers und sehr hoher CPU Auslastung die Messungen nicht beeinflusst werden.
- Implementierung auf Basis eines Push-Based Ansatzes. Es gäbe keine Notwendigkeit mehr für Announcements.
- Verwendung einer hierarchischen Struktur, um das Problem der quadratisch wachsenden Anzahl an Verbindungen zu verringern.

Gliederung

Sinn des Anhangs

In Diskussionen nach Präsentationen kommt es häufiger vor, dass Nachfragen gestellt werden, die mit dem regulären Material der Präsentation nicht zu beantworten sind.
Daher lohnt es sich, zusätzliche

- Grafiken
- Tabellen mit Detailinformationen
- Erklärungen

aus der schriftlichen Arbeit im Anhang unterzubringen.