
DSLP 3.0 Spezifikation

Peter Tröger

02.05.2020

Inhaltsverzeichnis:

1	Einführung	2
2	Konventionen im Dokument	2
3	Verbindungsaufbau und -abbau	2
4	Struktur einer DSLP-Nachricht	2
4.1	Nachrichtenkopf	2
4.2	Nachrichtenkörper	3
5	Nachrichtentypen	3
5.1	request time	3
5.2	response time	3
5.3	group join	4
5.4	group join ack	4
5.5	group leave	5
5.6	group leave ack	5
5.7	group notify	5
5.8	user join	6
5.9	user join ack	6
5.10	user leave	7
5.11	user leave ack	7
5.12	user text notify	7
5.13	user file notify	8
5.14	error	8
6	Fehlerbehandlung	9
7	Datenschutz	9
8	Versionen dieser Dokuments	9
9	Adresse des Autors	10

1 Einführung

Das *Distributed Systems Learning Protocol (DSLP)* realisiert die Übertragung von Nachrichten über eine bereits etablierte Transportverbindung.

2 Konventionen im Dokument

Großgeschriebene Schlüsselwörter sollen in diesem Dokument wie folgt interpretiert werden:

MUSS / DARF NICHT Dies ist eine zwingende Anforderung der Spezifikation.

SOLLTE / SOLLTE NICHT Es kann begründete Fälle geben, in denen die Spezifikation in diesem Punkt ignoriert werden kann.

KANN Die Erfüllung der Spezifikation ist in diesem Punkt optional.

3 Verbindungsaufbau und -abbau

Der Aufbau und Abbau einer Transportverbindung, beispielsweise mit TCP/IP, wird von dieser Spezifikation nicht geregelt. Im Folgenden wird daher lediglich in Sender und Empfänger einer DSLP-Nachricht unterschieden, welche eine etablierte Transportverbindung nutzen. Es wird davon ausgegangen, dass pro Verbindung nur zwei Teilnehmer existieren, von denen zu jedem Zeitpunkt einer als Sender und einer als Empfänger agiert.

4 Struktur einer DSLP-Nachricht

Eine DSLP-Nachricht besteht aufeinanderfolgend aus einem Nachrichtenkopf, der enthalten sein MUSS, und einem (potentiell leeren) Nachrichtenkörper.

Jede Textzeile in einer DSLP-Nachricht MUSS im UTF-8 Format (ohne BOM) übertragen werden und wird mit den nicht-druckbaren Zeichen ‚\r‘ (0x0D) und ‚\n‘ (0x0A) als Zeilenendemarkierung abgeschlossen. Ein Empfänger KANN die maximale Länge von Textzeilen in einer Nachricht beschränken, MUSS den Sender der Nachricht dann aber über diesen Umstand durch eine `error` - Nachricht informieren.

4.1 Nachrichtenkopf

Der Nachrichtenkopf MUSS als eine Menge von Textzeilen übertragen werden.

Die erste Zeile in jedem DSLP-Nachrichtenkopf, welche dem hier beschriebenen Protokoll folgt, MUSS wie folgt lauten:

<code>dslp-3.0</code>

Die zweite Zeile im Nachrichtenkopf definiert den Typ der Nachricht. Hier MUSS eine der folgenden Textzeilen gesendet werden:

- `request time`
- `response time`
- `group join`
- `group join ack`

- group leave
- group leave ack
- group notify
- user join
- user join ack
- user leave
- user leave ack
- user text notify
- user file notify
- error

Etwaige weiteren Zeilen im Nachrichtenkopf ergeben sich aus der Typangabe.

Der Nachrichtenkopf MUSS mit der folgenden Textzeile enden:

```
dslp-body
```

4.2 Nachrichtenkörper

Der Inhalt des Nachrichtenkörpers ergibt sich aus der Typangabe.

5 Nachrichtentypen

5.1 request time

Der Sender fordert den Empfänger auf, ihm seine aktuelle Uhrzeit zu übermitteln. Der Empfänger SOLLTE mit einer `response time` - Nachricht an den Sender antworten. Die Antwort KANN verzögert erfolgen.

Im Nachrichtenkopf und -körper werden keine zusätzlichen Informationen übertragen.

Beispiel:

```
dslp-3.0
request time
dslp-body
```

5.2 response time

Der Sender liefert seine aktuelle Uhrzeit an den Empfänger. Der Empfänger MUSS zuvor eine `request time` - Nachricht geschickt haben.

Im Nachrichtenkopf werden keine zusätzlichen Informationen übertragen.

Der Nachrichtenkörper MUSS eine einzelne Textzeile mit der Zeitangabe im folgenden Format enthalten:

```
yyyy-MM-dd'T'HH:mm:ssX
```

yyyy: Vierstellige Jahresangabe

MM: Zweistellige Monatsangabe

dd: Zweistellige Tagesangabe

HH: Zweistellige Stundenangabe

mm: Zweistellige Minutenangabe

ss: Zweistellige Sekundenangabe

X: Angabe der Zeitzone im ISO 8601-Format

Beispiel:

```
dslp-3.0
response time
dslp-body
2018-10-15T16:00:23+02:00
```

5.3 group join

Der Sender teilt dem Empfänger mit, dass er einer bestimmten Gruppe beitreten möchte. Die Mitgliedschaft in der Gruppe MUSS durch eine `group join ack` - Nachricht als Antwort vom Empfänger bestätigt werden, um gültig zu sein. Ab diesem Zeitpunkt SOLLTEN dem Sender der `group join` - Nachricht neue `group notify` - Nachrichten für diese Gruppe zugestellt werden. Der Empfänger der `group join` - Nachricht KANN dafür selbst `group notify` - Nachrichten weiterleiten, die er empfangen hat. Diese Weiterleitung KANN für beliebige Zeit verzögert werden.

Der Empfänger MUSS mehrere `group join` - Nachrichten für verschiedene Gruppen akzeptieren, sowohl auf der gleichen als auch auf unterschiedlichen Transportverbindungen.

Im Nachrichtenkopf MUSS zusätzlich eine Textzeile mit dem Namen der Gruppe übermittelt werden. Der Name DARF NICHT mit der Zeichenkette „dslp-“ beginnen, dies MUSS vom Sender sichergestellt werden.

Im Nachrichtenkörper werden keine zusätzlichen Informationen übertragen.

Der Empfänger KANN mit einer `error` - Nachricht antworten, falls der Sender bereits der Gruppe beigetreten ist.

Beispiel:

```
dslp-3.0
group join
Electronic Music Fans
dslp-body
```

5.4 group join ack

Der Sender teilt dem Empfänger mit, dass er ihn als Mitglied einer Gruppe registriert hat und zukünftig unaufgefordert `group notify` - Nachrichten senden wird. Der Empfänger MUSS zuvor eine entsprechende `group join` - Nachricht gesendet haben.

Im Nachrichtenkopf MUSS zusätzlich eine Textzeile mit dem Namen der Gruppe übermittelt werden. Im Nachrichtenkörper werden keine zusätzlichen Informationen übertragen.

Der Empfänger KANN mit einer `error` - Nachricht antworten, falls er dieser Gruppe nie beigetreten ist. Der Status der Gruppenmitgliedschaft ist in diesem Fall undefiniert.

Beispiel:

```
dslp-3.0
group join ack
Electronic Music Fans
dslp-body
```

5.5 group leave

Der Sender teilt dem Empfänger mit, dass er eine Gruppe verlassen möchte. Das Verlassen der Gruppe MUSS durch eine `group leave ack` - Nachricht als Antwort bestätigt werden, bevor es gültig wird. Ab diesem Zeitpunkt SOLLTE der Empfänger der `group leave`- Nachricht dem Sender keine `group notify` - Nachrichten mehr weiterleiten, welche an die angegebene Gruppe gerichtet sind.

Im Nachrichtenkopf MUSS zusätzlich eine Textzeile mit dem Namen der Gruppe übermittelt werden. Im Nachrichtenkörper werden keine zusätzlichen Informationen übertragen.

Der Empfänger KANN mit einer `error` - Nachricht antworten, falls der Sender nicht zuvor als Mitglied der angegebenen Gruppe mit einer `group join` - Nachricht registriert wurde.

Beispiel:

```
dslp-3.0
group leave
Electronic Music Fans
dslp-body
```

5.6 group leave ack

Der Sender teilt dem Empfänger mit, dass er ihn aus einer Gruppe entfernt hat und zukünftig keine `group notify` - Nachrichten für diese Gruppe senden wird. Der Empfänger MUSS zuvor eine entsprechende `group leave` - Nachricht gesendet haben.

Im Nachrichtenkopf MUSS zusätzlich eine Textzeile mit dem Namen der Gruppe übermittelt werden. Im Nachrichtenkörper werden keine zusätzlichen Informationen übertragen.

Der Empfänger KANN mit einer `error` - Nachricht antworten, falls er nie eine entsprechende `group leave` - Nachricht geschickt hat. Der Status der Gruppenmitgliedschaft ist in diesem Fall undefiniert.

Beispiel:

```
dslp-3.0
group leave ack
Electronic Music Fans
dslp-body
```

5.7 group notify

Der Sender teilt dem Empfänger eine Textinformation für eine Gruppe mit.

Im Nachrichtenkörper wird die Information als Menge von Textzeilen übermittelt.

Im Nachrichtenkopf MUSS zusätzlich die folgende Menge an Textzeilen in dieser Reihenfolge übermittelt werden:

- Name der Gruppe
- Anzahl der Textzeilen im Nachrichtenkörper

Der Empfänger KANN mit einer `error` - Nachricht antworten, falls der Sender nicht zuvor als Mitglied der angegebenen Gruppe mit einer `group join` - Nachricht registriert wurde.

Beispiel:

```
dslp-3.0
group notify
Electronic Music Fans
2
dslp-body
Ist das neue Album von Modeselektor schon raus?
Wo kann man das kriegen?
```

5.8 user join

Der Sender teilt dem Empfänger mit, dass ein Nutzer mit einer bestimmten Kennung sich registrieren möchte. Dies MUSS durch eine `user join ack` - Nachricht vom Empfänger als Antwort bestätigt werden, bevor die Registrierung gültig wird. Ab diesem Zeitpunkt SOLLTEN dem Sender `user text notify` und `user file notify` - Nachrichten für diese Nutzerkennung zugestellt werden. Der Empfänger KANN dafür `user text notify` und `user file notify` - Nachrichten weiterleiten, die er selbst empfangen hat. Diese Weiterleitung KANN für beliebige Zeit verzögert werden.

Der Empfänger MUSS mehrere `user join` - Nachrichten für verschiedene Nutzerkennungen akzeptieren, sowohl auf der gleichen als auch auf unterschiedlichen Transportverbindungen.

Im Nachrichtenkopf MUSS zusätzlich eine Textzeile mit der Nutzerkennung übermittelt werden. Die Nutzerkennung DARF NICHT mit der Zeichenkette „dslp-“ beginnen, dies MUSS vom Sender sichergestellt werden.

Im Nachrichtenkörper werden keine zusätzlichen Informationen übertragen.

Der Empfänger MUSS mit einer `error` - Nachricht antworten, falls diese Nutzerkennung bereits bei ihm registriert ist.

Beispiel:

```
dslp-3.0
user join
Heinz
dslp-body
```

5.9 user join ack

Der Sender teilt dem Empfänger mit, dass er ihn mit einer Nutzerkennung registriert hat und zukünftig unaufgefordert `user text notify` und `user file notify` - Nachrichten senden wird. Der Empfänger MUSS zuvor eine entsprechende `user join` - Nachricht gesendet haben.

Im Nachrichtenkopf MUSS zusätzlich eine Textzeile mit der Nutzerkennung übermittelt werden. Im Nachrichtenkörper werden keine zusätzlichen Informationen übertragen.

Der Empfänger KANN mit einer `error` - Nachricht antworten, falls er diese Nutzerkennung nie registriert hat. Der Status der Registrierung ist in diesem Fall undefiniert.

Beispiel:

```
dslp-3.0
user join ack
```

(Fortsetzung auf der nächsten Seite)

```
Heinz
dslp-body
```

5.10 user leave

Der Sender teilt dem Empfänger mit, dass er die Registrierung einer bestimmten Nutzerkennung aufheben möchte. Dies MUSS durch eine `user leave ack` - Nachricht als Antwort bestätigt werden, bevor die Aufhebung gültig wird. Ab diesem Zeitpunkt SOLLTEN dem Sender keine `user text notify` oder `user file notify` - Nachrichten mehr vom Empfänger, für diese Kennung, zugestellt werden.

Im Nachrichtenkopf MUSS zusätzlich eine Textzeile mit der Nutzerkennung übermittelt werden.

Im Nachrichtenkörper werden keine zusätzlichen Informationen übertragen.

Der Empfänger KANN mit einer `error` - Nachricht antworten, falls der Sender nicht zuvor die angegebene Kennung registriert hat. Der Status der Registrierung ist in diesem Fall undefiniert.

Beispiel:

```
dslp-3.0
user leave
Heinz
dslp-body
```

5.11 user leave ack

Der Sender teilt dem Empfänger mit, dass er seine Registrierung einer bestimmten Nutzerkennung aufgehoben hat. Der Empfänger MUSS zuvor eine entsprechende `user leave` - Nachricht gesendet haben.

Im Nachrichtenkopf MUSS zusätzlich eine Textzeile mit der Nutzerkennung übermittelt werden. Im Nachrichtenkörper werden keine zusätzlichen Informationen übertragen.

Der Empfänger KANN mit einer `error` - Nachricht antworten, falls er nie eine entsprechende `user leave` - Nachricht geschickt hat. Der Status der Registrierung ist in diesem Fall undefiniert.

Beispiel:

```
dslp-3.0
user leave ack
Heinz
dslp-body
```

5.12 user text notify

Der Sender teilt dem Empfänger eine Textinformation für eine bestimmte Nutzerkennung mit.

Im Nachrichtenkopf MUSS zusätzlich die folgende Menge an Textzeilen in dieser Reihenfolge übermittelt werden:

- Autorenkennung: Kennung des Nutzers, der die Nachricht verfasst hat
- Leserkennung: Kennung des Nutzers, der die Nachricht lesen soll
- Anzahl der Textzeilen im Nachrichtenkörper

Im Nachrichtenkörper wird die Information als Menge von Textzeilen übermittelt.

Der Empfänger MUSS mit einer `error` - Nachricht antworten, falls die Autorekennung nicht zuvor durch den Sender mit `user join` registriert wurde. In diesem Fall soll keine Weiterleitung stattfinden.

Der Empfänger KANN mit einer `error` - Nachricht antworten, falls die Leserkennung unbekannt ist.

Beispiel:

```
dslp-3.0
user text notify
Heinz
Gregor
2
dslp-body
Essen gehen?
Ich habe Hunger.
```

5.13 user file notify

Der Sender übermittelt dem Empfänger eine Datei, die für eine bestimmte Nutzerkennung gedacht ist.

Im Nachrichtenkopf MUSS zusätzlich die folgende Menge an Textzeilen in dieser Reihenfolge übermittelt werden:

- Autorekennung: Kennung des Nutzers, der die Datei zur Verfügung stellt
- Leserkennung: Kennung des Nutzers, der die Datei erhalten soll
- Name der Datei, ohne Pfadangaben
- MIME-Typ der Datei
- Anzahl der Bytes im Nachrichtenkörper

Im Nachrichtenkörper wird die Datei als Bytefolge übermittelt.

Der Empfänger MUSS mit einer `error` - Nachricht antworten, falls die Autorekennung nicht zuvor durch den Sender mit `user join` registriert wurde. In diesem Fall soll keine Weiterleitung stattfinden.

Der Empfänger KANN mit einer `error` - Nachricht antworten, falls die Leserkennung unbekannt ist.

Beispiel:

```
dslp-3.0
user file notify
Heinz
Gregor
formular.pdf
application/pdf
452162
dslp-body
<... 452162 Bytes binäre Daten ...>
```

5.14 error

Der Sender / Empfänger teilt dem Empfänger / Sender ein Problem mit.

Bei dieser Nachricht wird eine menschenlesbare Fehlermeldung als Textzeilen im Nachrichtenkörper übertragen.

Der Nachrichtenkopf enthält zusätzlich die Anzahl der Zeilen im Nachrichtenkörper.

Beispiel:

```
dslp-3.0
error
2
dslp-body
I don't know what the message type 'foobar' means.
Please consult the DSLP 3.0 specification for a list of valid message types.
```

6 Fehlerbehandlung

Der Empfänger einer Nachricht KANN die darunterliegende Transportverbindung jederzeit beenden. Er KANN eingehende ungültige Nachrichten wahlweise ignorieren oder eine `error` - Nachricht als Antwort senden, falls der konkrete Fehlerfall durch dieses Dokument nicht näher geregelt ist.

Der Versand einer `error` - Nachricht SOLLTE die anschließende Beendigung der Verbindung durch den Sender zur Folge haben.

7 Datenschutz

Das Protokoll enthält keine Sicherheitsvorkehrungen. Alle Daten werden unverschlüsselt übertragen, die Kommunikationspartner werden nicht authentifiziert.

8 Versionen dieser Dokuments

2.5.2020 (Version 3.0)

- Neue Trennzeilen für Nachrichtenbestandteile: `dslp-3.0` und `dslp-body`
- Neue Nachrichtentypen: `user join ack`, `user join leave`, `group join ack`, `group join leave`
- Mehr Details zum Verhalten bei fehlenden oder mehrfachen Nachrichten

2.5.2019 (Version 2.0)

- Trennung in Kopf und Körper durch `dslp/body`
- Längenangabe für Körper ersetzt Endemarkierung
- Entfernung von `peer notify`, `peer join` und `peer leave`
- Neue Nachrichtentypen: `user join`, `user leave`, `user file notify`, `user text notify`
- Mehr Details zum Verhalten in Fehlerfällen

12.11.2018 (Version 1.2)

- Neuer Nachrichtentyp: `peer notify`

02.11.2018 (Version 1.1)

- Verfeinerung der Beschreibung für Fehlerbehandlung
- Korrektur von Rechtschreibfehlern

29.10.2018 (Version 1.1)

- Neue Nachrichtentypen: `group join`, `group leave`, `group notify`
- Genauere Beschreibung der Fehlerbehandlung beim Empfänger
- Beschreibung des Sonderfalls, dass eine Endemarkierung als Datenzeile übertragen werden soll

15.10.2018 (Version 1.0)

- Initiale Version

9 Adresse des Autors

Peter Tröger

Beuth Hochschule für Technik Berlin

Berlin, Deutschland

Email: peter.troeger@beuth-hochschule.de