

Digital Fountain

Seminararbeit

vorgelegt am

Lehrstuhl für Praktische Informatik IV

Prof. Dr. W. Effelsberg

Universität Mannheim

im

Mai 2002

von

Katrin E. Piotrowski

aus München

Inhaltsverzeichnis

| | Seite |
|--|-------|
| Abkürzungsverzeichnis..... | II |
| Abbildungsverzeichnis..... | III |
| Tabellenverzeichnis..... | IV |
| | |
| 1 Einleitung | 1 |
| 2 Vergleich FTP mit Digital Fountain | 3 |
| 3 Digital Fountain-Ansatz | 4 |
| 3.1 Digital Fountain-Architektur | 4 |
| 3.2 Meta-Content™ Technologie | 9 |
| 4 Leistungsparameter | 15 |
| 5 Kritische Würdigung..... | 18 |
| | |
| Literaturverzeichnis..... | V |

Abkürzungsverzeichnis

| | |
|-----------------|--------------------------------|
| CPU: | Central Processing Unit |
| FLID: | Fair Layered Increase Decrease |
| FTP: | File Transfer Protocol |
| GB: | Gigabyte |
| I/O-Ressourcen: | Input-/ Output-Ressourcen |
| IP: | Internet Protocol |
| LTC: | Luby Transform Codes |
| MB: | Megabyte |
| PC: | Personal Computer |
| RSC: | Reed-Solomon Code |
| RTT: | Round Trip Time |
| TCP: | Transmission Control Protocol |
| UDP: | User Datagram Protocol |
| WWW: | World Wide Web |
| XOR: | Exclusive-Or |

III

Abbildungsverzeichnis

| | | |
|--------------|---|----|
| Abbildung 1: | Architektur eines konventionellen Servers | 5 |
| Abbildung 2: | Architektur des Digital Fountain-Servers | 6 |
| Abbildung 3: | Verschlüsselung der Originaldatei..... | 7 |
| Abbildung 4: | Rekonstruktion der Originaldatei | 8 |
| Abbildung 5: | Umwandlungsmechanismus..... | 8 |
| Abbildung 6: | Datenübertragung mittels Meta-Content-Technologie | 10 |
| Abbildung 7: | Tornado Code | 12 |
| Abbildung 8: | Funktionsweise von FLID..... | 14 |
| Abbildung 9: | Vergleich von Digital Fountain mit FTP | 16 |

Tabellenverzeichnis

| | | |
|------------|---|----|
| Tabelle 1: | Vergleich von Tornado Code mit Reed-Solomon Code..... | 13 |
| Tabelle 2: | Übertragungszeit mit einer 1.5 Mbps Verbindung..... | 17 |

1 Einleitung

Eines der wichtigsten Probleme auf dem Weg zur effizienten Nutzung des Internets ist die Übertragung großer Datenmengen an viele Benutzer. Beispiele hierfür sind z.B. Programm-Updates und Video-On-Demand. Hierbei sind Datenmengen nicht nur groß, sondern auch heterogen, weshalb klassische Verfahren wie das File Transfer Protocol (FTP) versagen.¹

Das Transmission Control Protocol/ Internet Protocol (TCP/IP) ist eines der meistgenutzten Protokolle zum Transfer von Daten. FTP benutzt TCP. Hierbei werden für n Empfänger n Verbindungen aufgebaut, wobei das Sliding Window Protokoll zum Versenden der Datenpakete verwendet wird. Der Server sendet innerhalb einer festgelegten Zeit eine bestimmte Anzahl von Datenpaketen und wartet auf Bestätigungen seitens des Clients, bevor er weitere Pakete versendet. Somit kann der Server überprüfen, welche Pakete erfolgreich empfangen wurden. Falls Bestätigungsantworten ausbleiben, erkennt der Server, welche Pakete auf dem Weg zum Client verloren gegangen sind und verschickt diese nochmalig. Die Datenpakete sind mit Sequenznummern versehen, so dass der Client überprüfen kann, welche Pakete er erhalten hat und in welcher Reihenfolge diese dekodiert werden müssen.²

Digital Fountain³ verfolgt den Ansatz, ein neues Multi- und Broadcast-Protokoll zu entwickeln, das eine Rückmeldung der Clients völlig überflüssig macht. Dadurch kann ein Server potenziell unendlich viele Clients bedienen. Dabei wurde das Ziel hoher Effizienz, Ausfallsicherheit, Toleranz und Leistungsfähigkeit nach Anforderungsbedarf verfolgt.⁴

Im Gegensatz zu den Standardprodukten verwendet Digital Fountain ihre eigens entwickelte Meta-ContentTM-Technologie. Die kodierten Datenpakete bestehen aus einer beliebigen Menge von Gleichungen, die aus der gesamten Datei generiert werden. Diese Technologie macht nochmaliges Versenden und Empfangen von Bestätigungen obsolet, wodurch Datenverlust bei der Datenübertragung als irrelevant gilt.⁵

¹ Vgl. Byers, J.W.; Luby, M.; Mitzenmacher, M.; Rege, A. (1998): S. 1.

² Vgl. Black, U. (1992): S. 146-150.

³ Das Unternehmen Digital Fountain wurde im Jahr 1998 gegründet und arbeitet mit Cisco Systems und Sony zusammen. Weitere Informationen zu dem Unternehmen: <http://www.digitalfountain.com/company/index.htm>.

⁴ Vgl. Byers, J.W.; Luby, M.; Mitzenmacher, M.; Rege, A. (1998): S. 3.

⁵ Vgl. Byers, J.W.; Luby, M.; Mitzenmacher, M.; Rege, A. (1998): S. 1.

Ziel der Arbeit ist ein vollständiges Herausarbeiten der Besonderheiten der Digital Fountain-Technologie. Hierbei werden zu Beginn die Unterschiede zu FTP aufgezeigt. Im dritten Abschnitt erfolgt eine ausführliche Betrachtung des Digital Fountain-Ansatzes: neben der Beschreibung der Digital Fountain-Architektur wird auf die eigentliche Technologie und ihre Besonderheiten im Unterschied zu herkömmlichen Codierungsalgorithmen näher eingegangen. Dabei liegt der Fokus auf dem Mechanismus der Datenübertragung und des hausinternen Codierungsalgorithmuses, dem Luby Transform Code. Aspekte der Performance und Leistungsfähigkeit machen den innovativen Ansatz zur Datenübertragung im Vergleich zu FTP deutlich. Eine kritische Würdigung dieser Technologie rundet die Arbeit ab.

2 Vergleich FTP mit Digital Fountain

Standardmäßig wird im Internet FTP zur Datenübertragung verwendet. Mittels FTP können Daten jeden Typs wie Text-, Bild-, Ton-, Video- und Programmdateien übertragen werden. Mittlerweile ist FTP soweit in das World Wide Web (WWW) integriert, dass es den meisten Nutzern nicht mehr auffällt, wenn sie FTP nutzen.

FTP setzt auf TCP auf. Zur Durchführung eines Dateitransfers benötigt FTP zwei gleichzeitig aktive TCP-Verbindungen. Zunächst baut der FTP-Client eine Managementverbindung auf. Die eigentliche Datenübertragung erfolgt dann über die zweite Verbindung.⁶

Ein Hauptmerkmal von TCP ist die Gewährleistung einer zuverlässigen Datenübertragung, wobei TCP zu den verbindungsorientierten im Gegensatz zu den verbindungslosen Protokollen gehört. Für die Übertragung sind die zu versendenden Pakete mit Sequenznummern versehen, die, in der richtigen Reihenfolge dekodiert, zur Wiederherstellung der Originaldatei führen. Die Adressierung erfolgt über IP-Adressen. Um Ausfallsicherheit zu gewährleisten, verwendet TCP positive Quittungen (Acknowledgements) nach Empfang von Informationen und wiederholte Übertragungen beim Ausbleiben von Bestätigungen. Der Server versendet, nachdem eine bestimmte Zeit überschritten ist, die zuletzt gesendeten Pakete nochmalig. Diese Zeitüberschreitung wird ‚Timeout‘ genannt.

Wiederholtes Versenden führt dazu, dass Pakete doppelt in Empfang genommen werden; mithilfe von Sequenznummern wird überprüft, welche Pakete bereits eingegangen sind. Somit kann die vollständige und korrekte Dekodierung gewährleistet werden. TCP benutzt das Sliding Window Protokoll. Die Pakete werden in der richtigen Reihenfolge der Sequenznummern dekodiert und führen so zur Wiederherstellung der Originaldatei. Ferner passt die Flusskontrolle die Übertragungsrates der Datenpakete der Empfangskapazität an.

Digital Fountain verwendet das User Datagram Protocol (UDP).⁷ UDP ist ein unzuverlässiges, verbindungsloses Protokoll und als einfache Erweiterung des IPs zu sehen; dabei ist es im Gegensatz zu TCP multicast-fähig. Große Vorteile bietet UDP

⁶ Vgl. Black, U. (1992): S. 151.

⁷ Vgl. Digital Fountain (2001a): S. 9.

beim geringen Bedarf an Speicherplatz und Central Processing Unit (CPU)-Zeit. Neben keinem explizitem Verbindungsaufbaus ist die einfache Implementierung zu erwähnen.⁸

Digital Fountain muss, da es UDP benutzt, ihre eigenen Mechanismen zur Verfügung stellen, die für die Paketierung der Daten und die Flusskontrolle zur Vermeidung von Verstopfungen sorgen. UDP gewährleistet zwar eine zuverlässige Übertragung, verfügt jedoch über keine Fluss- und Überlastkontrolle. Digital Fountain implementiert einen effizienten Algorithmus, den Luby Transform Code, der ein wiederholtes Versenden von Datenpaketen und Versenden von Bestätigungen vermeidet. Für die Flusskontrolle⁹ setzt Digital Fountain Fair Layered Increase Decrease (FLID) ein.¹⁰

3 Digital Fountain-Ansatz

3.1 Digital Fountain-Architektur

FTP stößt an seine Grenzen, sobald dieselben Dateninhalte an eine große Menge von Empfänger-Clients übertragen werden sollen: FTP baut für jeden Client eine eigene Verbindung auf, so dass großer Bedarf an Speicher, CPU und Input-/ Output-Ressourcen (I/O-Ressourcen) notwendig ist. Selbst wenn alle Empfänger dieselben Daten erhalten, sind die Übertragungen unabhängig voneinander; somit wird die Serverkapazität zur Übertragung derselben Dateninhalte durch die verfügbaren Ressourcen des Servers begrenzt. Nachfolgende Abbildung¹¹ verdeutlicht die Architektur eines herkömmlichen Servers:

⁸ Vgl. Effelsberg (2001): 7.7-7.9.

⁹ Siehe Abschnitt 3.2.

¹⁰ Vgl. Digital Fountain (2001b): S. 6-12.

¹¹ Vgl. Abbildung 1.

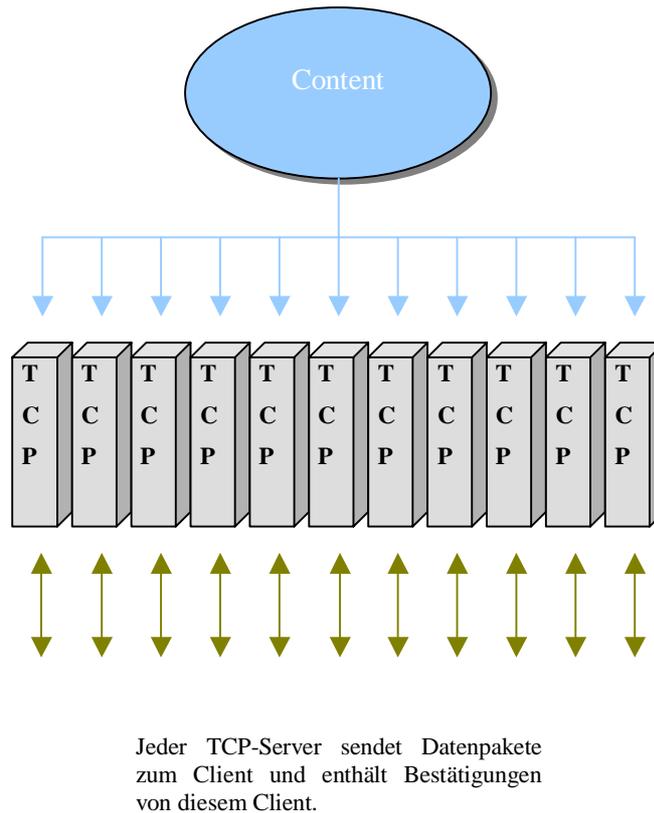


Abbildung 1: Architektur eines konventionellen Servers¹²

Die Digital Fountain-Technologie unterscheidet sich in folgender Art und Weise: zum einen werden nicht einzelne Verbindungen zu den Clients aufgebaut, sondern jeder Client bekommt die kodierten Datenpakete von einer ausgehenden „Verbindung“ und zum anderen ist die Durchsatzrate höher, da durch die patentierte Meta-Content™-Technologie keine Bestätigungen bei erfolgreicher Übertragung oder erneutes Versenden der Datenpakete notwendig sind.

Die Architektur von Digital Fountain besteht aus zwei Komponenten: auf der einen Seite findet man den hardwaretechnisch realisierten Digital Fountain Server und auf der anderen Seite die Digital Fountain Client Software.¹³

¹² In Anlehnung an: Digital Fountain (2001b): S. 5.

¹³ Vgl. Digital Fountain (2001b): S. 6-11.

- Digital Fountain Server

Dieser Server besteht aus den Komponenten Meta-Content-Maschine und dem Replication System.

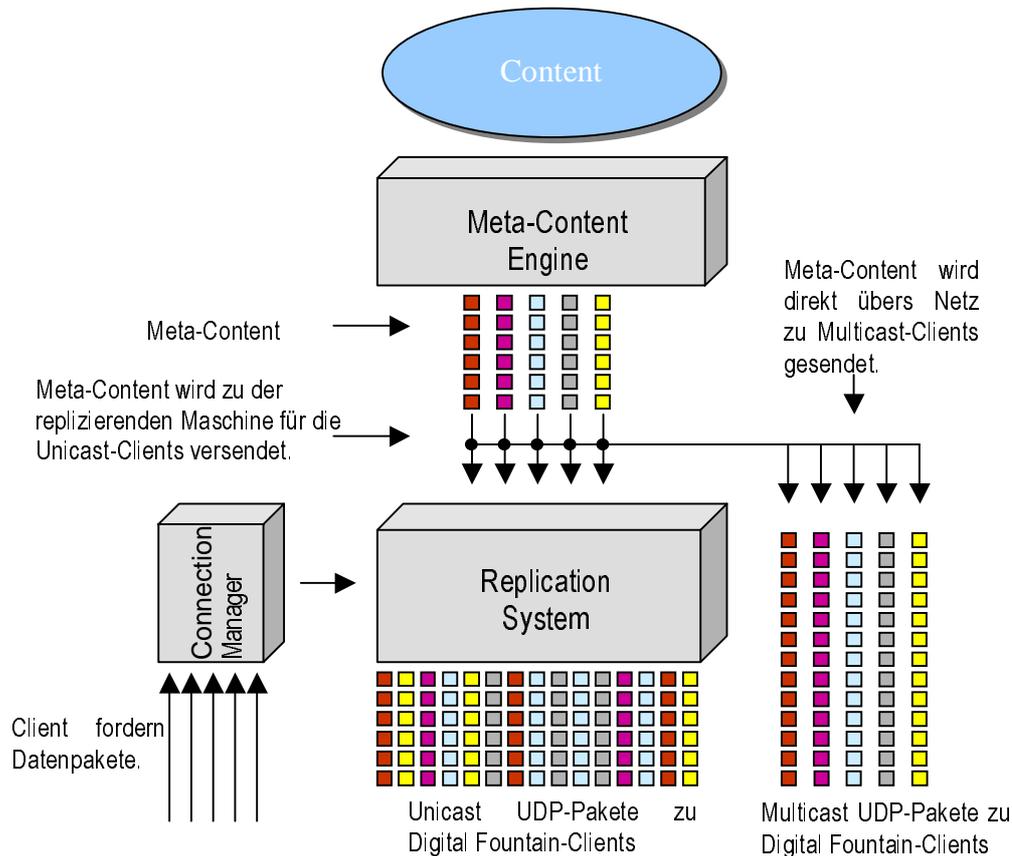


Abbildung 2: Architektur des Digital Fountain-Servers¹⁴

Die Originaldatei wird zunächst in einzelne Pakete aufgeteilt. Die Inhalte dieser Pakete werden zufällig in Meta-Content-Pakete zusammengefasst und kodiert versendet. Zur ‚Kodierung‘ wird der eigens von Digital Fountain entwickelte Code verwendet.¹⁵

Die Meta-Content-Maschine generiert aus jedem Paket der Originaldatei einen bis mehrere Ströme von Meta-Content-Paketen. Falls der Digital Fountain Client direkt an einem Broadcast- oder Multicast-fähigen Netzwerk angebunden ist, können die Meta-

¹⁴ In Anlehnung an: Digital Fountain (2001b): S. 7.

¹⁵ Siehe Abschnitt 3.2.

Content-Pakete direkt empfangen werden. Ansonsten dient das Replication System dazu, die Meta-Content-Pakete transformiert als Unicast-UDP-Pakete weiterzuschicken. Somit kann der Digital Fountain-Ansatz sowohl in uni- als auch in broad- und multicast-fähigen Netzwerken zum Einsatz kommen.

Die Meta-Content-Maschine generiert aus jedem einzelnen Datenstück der zu übertragenden Datei Ströme von Meta-Content-Paketen. Realisiert wird diese Umwandlung mittels des speziell von Michael Luby¹⁶ entwickelten Transformierungscode: Luby Transform Codes (LTC). Dieser Code beinhaltet einen Algorithmus, der die Pakete so generiert, dass allein die Tatsache ausreicht, genügend Pakete zu empfangen, ohne auf die Reihenfolge zu achten; somit kann die Originaldatei lediglich durch Kombination irgendwelcher Pakete rekonstruiert werden.

Die Meta-Content-Pakete sind durch die Verknüpfung der Originalpakete durch Exclusive-Or (XOR)-Operationen erzeugt.¹⁷ Alle Daten finden sich in irgendeiner Weise in den Meta-Daten wieder. Somit können die Pakete, die nicht empfangen wurden, durch bit-weises XOR rekonstruiert werden. Dabei ist es lediglich von großer Bedeutung, dass die Anzahl der Meta-Content-Pakete der Anzahl der ursprünglichen Pakete entspricht.

Nachfolgende Abbildungen veranschaulichen die Transformierung der Pakete in Meta-ContentTM-Pakete:

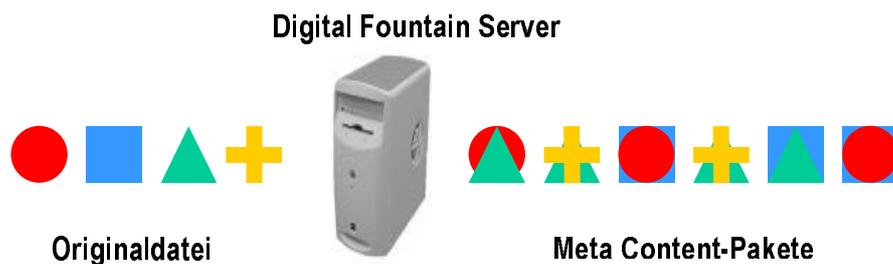


Abbildung 3: Verschlüsselung der Originaldatei¹⁸

¹⁶ Vgl. <http://www.digitalfountain.com/company/team.htm>

¹⁷ Vgl. Digital Fountain (2001b): S. 8.

¹⁸ Vgl. o.V. (2002a).

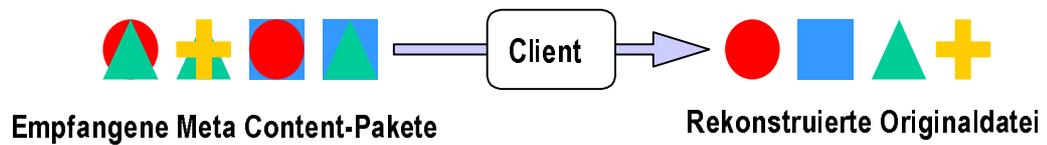


Abbildung 4: Rekonstruktion der Originaldatei¹⁹

Zuerst wird der Dateninhalt in gleichgroße Stücke logisch aufgeteilt. Diese Pakete werden mittels Kombinatorik in Meta-Pakete verpackt, so dass die Originaldatei mittels Kombinierens von Meta-Content-Paketen wiederhergestellt werden kann. Folgende Abbildung verdeutlicht die Technik des Umwandeln und Kombinierens innerhalb der Meta-Content-Pakete:

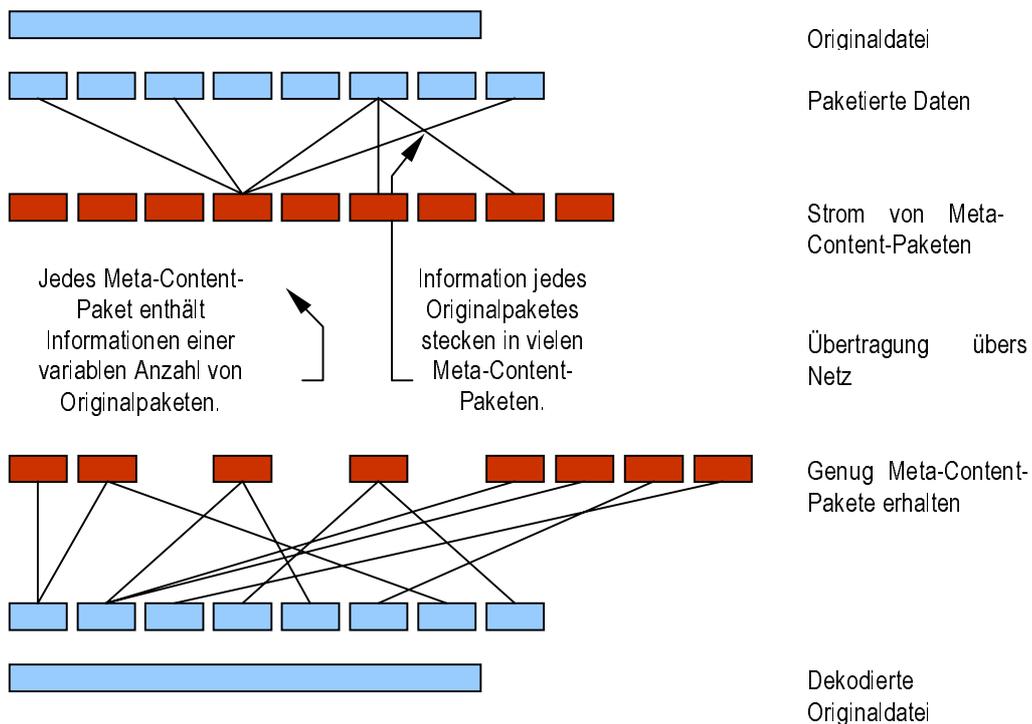


Abbildung 5: Umwandlungsmechanismus²⁰

¹⁹ Vgl. o.V. (2002a).

²⁰ In Anlehnung an: Digital Fountain (2001b): S. 10.

- Digital Fountain Client Software

Der Digital Fountain Client empfängt die Meta-Content-Pakete und rekonstruiert die Originaldatei. Digital Fountain Client Software ist durch das schnelle Wiederherstellen der Originaldatei gekennzeichnet, so dass wenig Speicherplatz notwendig ist; selbst wenn Gigabyte(GB)-große Daten entschlüsselt werden, übersteigt der Speicherbedarf nicht die Megabyte(MB)-Grenze.²¹ Falls die Multicast-Verbindung missglückt, bezieht der Client seine Daten über eine Unicast-Verbindung.

3.2 Meta-Content™ Technologie

Die Meta-Content™-Technologie ist der Schlüsselfaktor für das eigentliche Erfolgsrezept von Digital Fountain. Sie ermöglicht Ausfallsicherheit, hohe Performance und Skalierbarkeit in der Datenübertragung;²² diese Parameter werden von FTP wegen der Unicast-Fähigkeit und der Mechanismen zur zuverlässigen Datenübertragung auf diese Art nicht gewährleistet.

Der Digital Fountain Server verschlüsselt mittels des LTC die originären Daten und verschickt sie als Meta-Daten-Pakete übers Netz. Dabei ist es vernachlässigbar, in welcher Reihenfolge sie beim Empfänger ankommen und entschlüsselt werden; bei dieser Übertragungsmethode ist es ausschließlich von Bedeutung, dass genügend viele Datenpakete, d.h. genau so viele Datenpakete empfangen werden, wie originäre Datenpakete versendet wurden, damit die Entschlüsselung gewährleistet werden kann. Zusammengefasst hat jeder Meta-Content-Strom folgende Eigenschaften:²³

- Meta-Content-Pakete werden unabhängig von der Originaldatei erzeugt.
- Ein Teil der Originaldatei kann aus jedem empfangenen Meta-Content-Paket wiederhergestellt werden; dabei ist es lediglich wichtig, dass die Meta-Content-Pakete in der Gesamtsumme so viel Inhalt wie den aus den Originaldaten ergeben.

²¹ Vgl. Digital Fountain (2001b): S. 11.

²² Vgl. Digital Fountain (2001b): S. 6.

²³ Vgl. Digital Fountain (2001b): S. 6.

Aus diesen Eigenschaften wird nicht ersichtlich, dass die Reihenfolge der Meta-Content-Pakete zum Entschlüsseln von keinem Interesse sind. Die verschlüsselten Pakete sind gegenseitig austauschbar, es müssen lediglich eine genügend große Anzahl an Paketen empfangen werden. Die Gefahr, wichtige Datenpakete zu verlieren, ist damit gebannt. Dadurch, dass keine Bestätigungen versendet werden müssen, wird auch die Durchsatzrate erhöht. Die endgültige Zeit bis zum Wiederherstellen der Daten hängt lediglich von den Parametern „Netzwerkgeschwindigkeit“, „Verlustrate“ und der eigentlichen Größe der Originaldatei ab.²⁴

Nachfolgende Abbildung veranschaulicht das Verschlüsseln der Originaldaten in Meta-Daten-Pakete und das Entschlüsseln dieser Pakete:

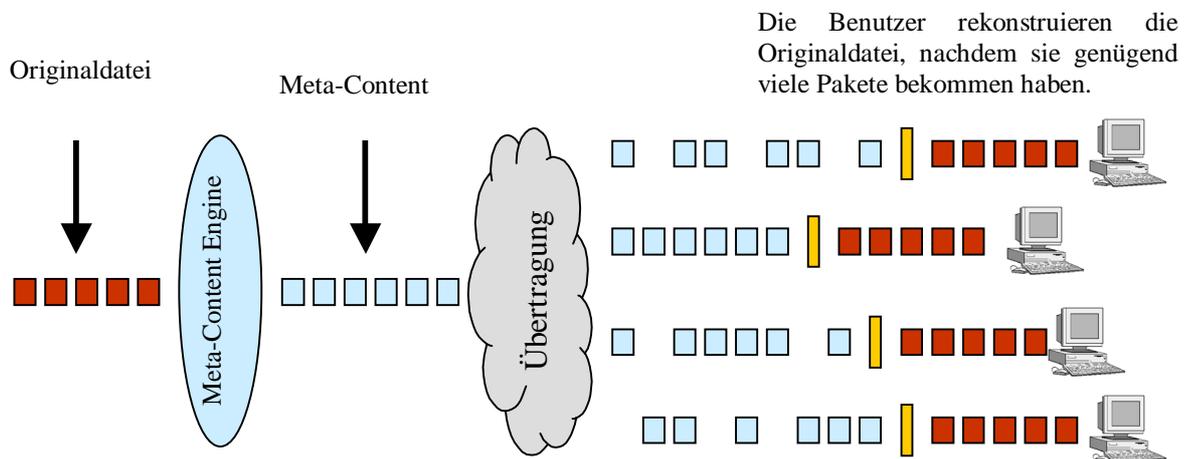


Abbildung 6: Datenübertragung mittels Meta-Content-Technologie²⁵

Im folgenden werden die wichtigsten Errungenschaften innerhalb dieser Technologie erläutert:

²⁴ Vgl. Digital Fountain (2001b): S. 6.

²⁵ In Anlehnung an: Digital Fountain (2001b): S. 6.

- Luby Transform Codes (LTC)

Digital Fountain setzten sich zum Ziel, eine Möglichkeit zur Datenübertragung zu generieren, die folgende Eigenschaften gewährleistet:²⁶

- Flexibilität
Jeder Client kann, wann immer er möchte, an der Datenübertragung teilnehmen.
- Flusskontrolle
Je nach Bandbreite der Verbindung zum Digital Fountain Server nehmen die Clients mit unterschiedlicher Auslastung an der Datenübertragung teil.
- Ausfallsicherheit
Die Datenpakete können jederzeit dekodiert werden, wenn genügend Datenpakete vorhanden sind.
- Skalierbarkeit
Zum einen werden keine negativen und positiven Quittungen für die Flusskontrolle benutzt. Pakete werden nur an die Clients versendet, die diese haben möchten.

Daran lässt sich erkennen, dass zum einen die Datenübertragung multicast-fähig sein muss und zum anderen die Verwendung eines bereits bestehenden Error-correcting Codes nicht möglich ist.²⁷ Nachfolgende Abschnitte erläutern die Funktionsweise und die Unterschiede herkömmlicher Codierungsalgorithmen mit dem LTC.

Die meisten Codierungsalgorithmen benutzen den Reed-Solomon Code (RSC). Hierbei rekonstruiert der Client die Originaldatei aus k erfolgreich übertragenden Paketen. Die Datei wird in Blöcke aufgeteilt, wobei jeder Block in k Teile segmentiert ist. An jeden Block werden $n-k$ verschlüsselte Segmente angehängt, die den Reed-Solomon Encoder benutzen. Somit besteht die zu versendende Datei aus n Elementen. Die Originaldatei, die eigentlich aus k Segmenten besteht, wird um $n-k$ ($=l$) Elemente erweitert, so dass l

²⁶ Vgl. Byers, J.W.; Luby, M.; Mitzenmacher, M.; Rege, A. (1998): S. 1-4.

²⁷ Vgl. Byers, J.W.; Luby, M.; Mitzenmacher, M.; Rege, A. (1998): S. 3.

Pakete während der Datenübertragung verloren gehen können, ohne dass die Qualität bei der Wiederherstellung beeinträchtigt wird. Die Größe n , die bei RSC klein gewählt wird, ist die Gesamtanzahl an Segmenten. Jedes Paket beinhaltet entweder eine Datei oder ein verschlüsseltes Segment eines Blockes, wobei der Encoding Index des Paketes den genauen Block bzw. Segment spezifiziert.²⁸

Ein anderer Codierungsansatz wird im Tornado Code realisiert. Hierbei besteht jede Datei aus k Segmenten, an die $n-k$ verschlüsselte Segmente angehängt werden, die die Tornado-Verschlüsselung benutzen. Der Unterschied zu dem RSC liegt in der Größe von k und n : k kann im Gegensatz zu RSC groß sein, wobei n um einen kleinen konstanten Faktor größer als k ist.²⁹

Im Beispiel ist $n = k + 1$, wobei alle Pakete zusammen eine festgelegte Länge von P haben. Folgende Abbildung zeigt, wie der Tornado Code funktioniert, wobei die zusätzlichen Pakete durch XOR-Kombination der Originalpakete generiert werden.

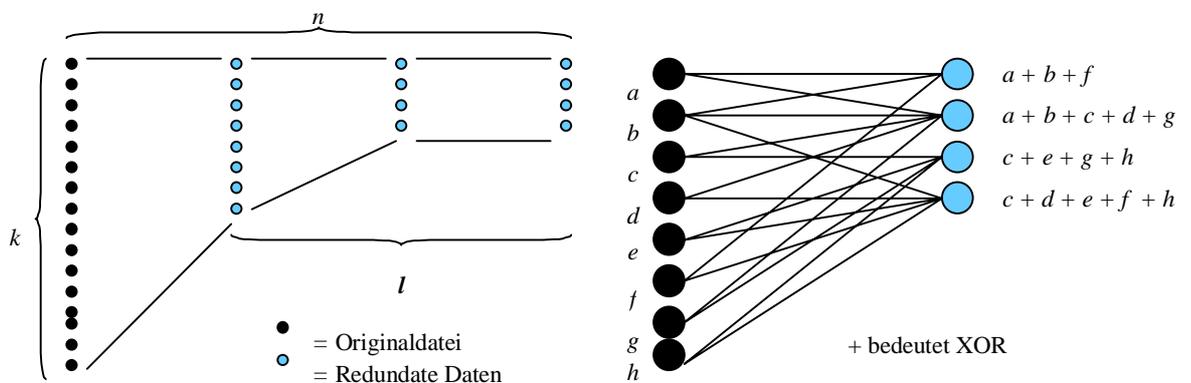


Abbildung 7: Tornado Code³⁰

Die folgende Tabelle zeigt die Eigenschaften der oben genannten Codes anhand ihrer Ver- und Entschlüsselungszeiten:

²⁸ Vgl. Byers, J.W.; Luby, M.; Mitzenmacher, M.; Rege, A. (1998): S. 3.

²⁹ Vgl. Byers, J.W.; Luby, M.; Mitzenmacher, M.; Rege, A. (1998): S. 4-7.

³⁰ In Anlehnung an Byers, J.W.; Luby, M.; Mitzenmacher, M.; Rege, A. (1998): S. 5.

| | Tornado Codes | Reed-Solomon Codes |
|--------------------------|---------------------------|---------------------------|
| Empfangs-Overhead | $\epsilon > 0$ required | 0 |
| Kodierungszeit | $(k + 1)\ln(1/\epsilon)P$ | $k(1 + l)P$ |
| Dekodierungszeit | $(k + 1)\ln(1/\epsilon)P$ | $k(1 + x)P$ |
| Operationen | XOR-Operation | Komplexe Feldoperationen |

Tabelle 1: Vergleich von Tornado Code mit Reed-Solomon Code³¹

Digital Fountain hat aus dem Tornado Code seinen eigenen LTC weiterentwickelt:³² eine Datei besteht wiederum aus k Segmenten. LTC generiert eine unendliche Anzahl von Meta-Content-Paketen, von denen eine beliebige Menge der Größe k zur Ursprungsdatei zusammengesetzt werden können. Jedes Paket enthält somit eine beliebige Menge verschiedener Informationen aus den ursprünglichen Datenpaketen. Die Größe k kann genauso wie bei Tornado Code groß gewählt werden.

Die Besonderheit bei diesem Code liegt darin, dass keine redundanten Pakete erzeugt werden; somit wird hier die Anzahl k gleich n gesetzt. Mithilfe dieses Codes sind alle Eigenschaften erfüllt. Die Ausfallsicherheit wird durch das kursierende Versenden der Pakete gewährleistet, so dass auch keine Bestätigungen seitens der Clients notwendig sind.

Digital Fountain hat als Flusskontrolle folgendes Protokoll programmiert:

- Fair Layered Increase Decrease (FLID)

Ein großer Vorteil bei dem Digital Fountain-Ansatz zur Datenübertragung ist ihre selbst programmierte und entwickelte Flusskontrolle. Der Digital Fountain Server verschickt die Meta-Content-Pakete über Paketströme zu den Clients. Diese Ströme werden „Layer“ genannt. Je nachdem, wie die zu empfangene Rate beim Client z.B. aufgrund von Paketverlust ausfällt, erhöht der Client den Empfang beim Datenstrom oder verringert ihn. Der Client tritt also so vielen „Layers“ bei, wie er verarbeiten kann.

³¹ In Anlehnung an Byers, J.W.; Luby, M.; Mitzenmacher, M.; Rege, A. (1998): S. 5.

Nachfolgende Abbildung verdeutlicht, wie der Server mit den Clients zusammenspielt und inwiefern das FLID Protokoll arbeitet:

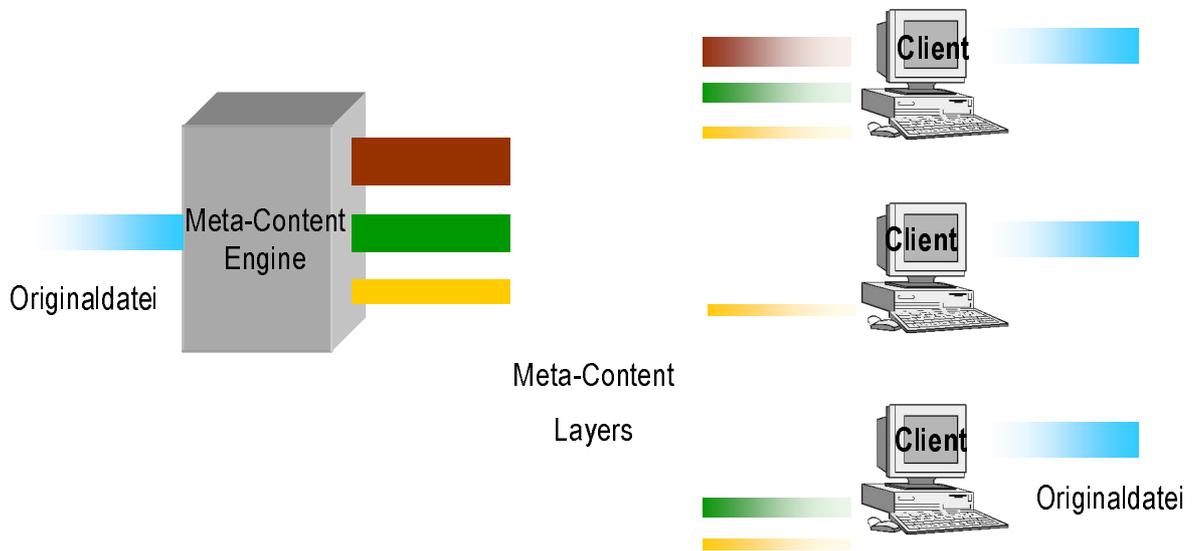


Abbildung 8: Funktionsweise von FLID³³

FLID ermöglicht ein Netzwerkmanagement, das der Client eigenständig steuert. Die Besonderheit liegt darin, dass diese Kontrolle gewährleistet, dass der von Digital Fountain generierte Datenverkehr sich fair zu jedem anderen Datenverkehr im Netzwerk verhält.³⁴ Der Verbraucher bestimmt, wann diese Kontrollfunktion ein- oder ausgeschaltet ist, so dass jeder andere Datenstrom ohne Einschränkung fließen kann. Somit können Datenströme, die mit FTP versendet werden, nicht behindert werden.

Ferner wird der Digital Fountain Server nicht beeinflusst, wenn die Fountain Clients in eine Broadcast- oder Multicast-Umgebung wechseln. FLID ermöglicht jedem Client, unabhängig voneinander ihre Datenrate, die an das jeweilige Netzwerk angepasst ist, zu korrigieren.³⁵ Somit arbeitet jeder Client unabhängig von den anderen und nutzt seine Ressourcen. Es ist vor allem für viele Nutzer entwickelt worden und unterstützt die Anforderungen an die von Digital Fountain geforderte Übertragungsrate.

³² Vgl. o.V. (2002b).

³³ In Anlehnung an Digital Fountain (2001b): S. 13.

³⁴ Vgl. Byers, J.W. u.a. (2000): S. 1-5.

³⁵ Vgl. Digital Fountain (2001b): S. 12.

4 Leistungsparameter

Zusammengefasst sind Merkmale wie „Ausfallsicherheit“, „Geschwindigkeit“ und „Skalierbarkeit“ bei diesem Ansatz optimal gelöst. Die Ausfallsicherheit wird durch den Meta-Content-Ansatz gelöst: der Client muss lediglich genügend Pakete erhalten; dabei ist es vernachlässigbar, in welcher Reihenfolge und welche Pakete ankommen. Im Gegensatz zu FTP mit TCP beeinflusst die Paketverlustrate somit weder die Ausfallsicherheit noch die Geschwindigkeit. *„The Digital Fountain solution combines the speed of UDP with the reliability of TCP.“*³⁶

Bei TCP wird die Übertragungsrate hauptsächlich von den geforderten Bestätigungen über den Erhalt der Pakete und durch nochmaliges Versenden bestimmter Pakete verringert, so dass die Bandbreite ausgelastet ist. Mit dem Ansatz von Digital Fountain wird eine hohe Übertragungsrate sichergestellt, indem die Meta-Content-Pakete lediglich solange im Umlauf sein müssen, bis genügend Pakete zum Wiederherstellen beim Client ankommen. Paketverlusten oder Verzögerungen spielen keine Rolle. *„The key benefit is that the administrator is in control of the rate of transfer-not the network conditions or the intrinsic limitations of a transfer protocol.“*³⁷

Als weiteres Merkmal ist die hohe Skalierbarkeit zu erwähnen, die durch seine Multicast-Fähigkeit gewährleistet wird: bei Digital Fountain wird kein „Feedback Channel“ verwendet. Somit ist genügend Speicherplatz vorhanden, um das Versenden von Meta-Datenpaketen an eine Menge von Clients nach oben unbegrenzt zu halten.

Nachfolgende Abbildung zeigt einen Vergleich zwischen Digital Fountain und FTP. Dabei ist eine verfügbare Bandbreite von 2 Mbps und eine Verlustrate von 2% gewählt:

³⁶ Digital Fountain (2001a): S. 3.

³⁷ Digital Fountain (2001b): S. 16.

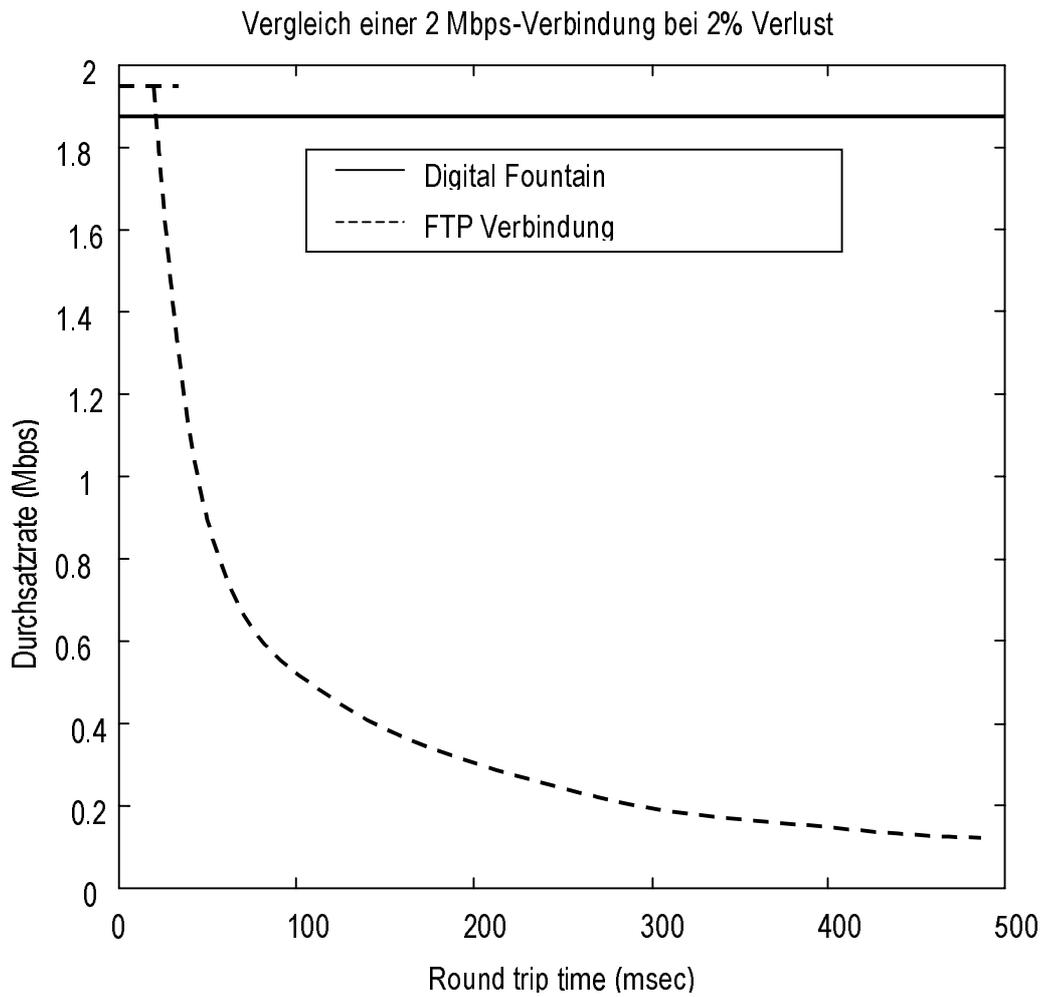


Abbildung 9: Vergleich von Digital Fountain mit FTP³⁸

Falls die Round Trip Time (RTT) bei 300msec liegt, erreicht Digital Fountain fast eine zehnmal größere Durchsatzrate als FTP.

Angenommen man hätte eine Verbindung mit 1,5 Mbps, dann würde die Übertragung folgendermaßen aussehen:

³⁸ Digital Fountain (2001a): S. 13.

| Dateigröße | FTP mit 300 Kbps* | FTP mit 460 Kbps** | Digital Fountain |
|-------------------|--------------------------|---------------------------|-------------------------|
| 200 MB | 1,5 h | 58 min | 18 min |
| 1 GB | 7,4 h | 4,8 h | 1,5 h |
| 5 GB | 37,0 h | 24,2 h | 7,6 h |

* Maximale Übertragungsrate bei 2% Verlustrate und 200 msec RTT.

** Maximale Übertragungsrate bei 1% Verlustrate and 200 msec RTT.

Tabelle 2: Übertragungszeit mit einer 1.5 Mbps Verbindung³⁹

³⁹ Digital Fountain (2001a): S. 14.

5 Kritische Würdigung

Abschließend kann erwähnt werden, dass die Technologie von Digital Fountain nur Vorteile aufweist, sofern die Übertragung über eine Multicast-Verbindung aufgebaut wird. Mittels Digital Fountain können Video- und Audio-Daten so effizient verpackt und übers Netz versendet werden, wie es gängige Server nicht gewährleisten.

Die Technologie verwendet zur Übertragung und Verschlüsselung Meta-Content™-Pakete. Damit ist es einem einzigen Server ermöglicht, je nach Bedarf große Datenmengen an eine unbegrenzte Anzahl von Clients zu senden. Sobald eine Multicast-Verbindung existiert, wird das Potenzial von Digital Fountain noch mehr ausgeschöpft.

Die Clients müssen sich weder um die Reihenfolge der eingehenden Pakete noch um Paketverlust kümmern. Sobald sie genügend Meta-Content™-Pakete erhalten haben, können sie mit der Entschlüsselung beginnen. Die eigentliche Datenübertragung wird durch nichts gestört, außer der Server drosselt die Übertragungsrate aufgrund einer Datenübertragung mit höherer Priorität.

So kann ein einziger Digital Fountain Server ein Video mit DVD-Qualität gleichzeitig an 4000 Benutzer übertragen; normalerweise sind für eine solche Datenübertragungen bis zu 30 herkömmliche Server notwendig.⁴⁰

⁴⁰ Vgl. Ferreira, A. (2001).

Literaturverzeichnis

- Black, U.D. (1992): TCP/IP and Related Protocols, McGraw-Hill Series on Computer Communications, 1992.
- Byers, J.W.; Luby, M.; Mitzenmacher, M.; Rege, A. (1998): A Digital Fountain Approach to Reliable Distribution of Bulk Data, <<http://citeseer.nj.nec.com/36766.html>>, 30.04.2002.
- Byers, J.W. u.a. (2000): FLID-DL: Congestion Control for Layered Multicast, <<http://citeseer.nj.nec.com/349388.html>>, 13.05.2002.
- Digital Fountain (2001a): Comparison of Digital Fountain File Transfer to FTP – Breaking the WAN Speed Barrier, A Digital Fountain White Paper, <<http://www.digitalfountain.com/technology/index.htm>>, 30.04.2002.
- Digital Fountain (2001b): Digital Fountain's Meta-Content™ Technology – A Quantum Leap in Content delivery, A Digital Fountain White Paper, <<http://www.digitalfountain.com/technology/index.htm>>, 30.04.2002.
- Effelsberg, W. (2001): Skript zur Vorlesung Rechnernetze im Sommersemester 2001, <<http://www.informatik.uni-mannheim.de/informatik/pi4/>>, 15.05.02.
- Ferreira, A. (2001): U.S.-Digital Fountain Streams Content, <<http://www.europemedia.net/shownews.asp?ArticleID=2718>>, 1.05.2002.
- Gemmell, J. u.a. (1999): Fcast Scalable Multicast File Distribution: Caching And Parameter Optimizations, <<http://citeseer.nj.nec.com/gemmell99fcast.html>>, 15.05.2002.
- Handley, M. u.a. (2000): The Reliable Multicast Design Space for Bulk Data Transfer, <<http://www.ietf.org/rfc/rfc2887.txt?number=2887>>, 14.05.2002.
- Luby, M. u.a. (2002): Asynchronous Layered Coding protocol instantiation, <<http://www.ietf.org/internet-drafts/draft-ietf-rmt-pi-alc-08.txt>>, 9.05.2002.

- Mahanti, A.; Eager, D.; Vernon, M.; Sundaram-Stukel, D. (2001): Scalable On-Demand Media Streaming with Packet Loss Recovery, <<http://citeseer.nj.nec.com/mahanti01scalable.html>>, 13.05.2002.
- o.V. (2002a): Digital Fountain – Core Technology, <<http://www.digitalfountain.com/technology/coreTechnology.htm>>, 1.05.2002.
- o.V. (2002b): Digital Fountain Receives Patent for Groundbreaking “Meta-Content™” Technology - Unique Technology Promises to Change the Way Data is Distributed, <<http://www.digitalfountain.com/news/pressReleaseItem.htm?uid=22>>, 1.05.2002.
- Rizzo, L.; Vicisano, L. (1997): A Reliable Multicast data Distribution Protocol based on software FEC techniques, <<http://citeseer.nj.nec.com/96037.html>>, 1.05.2002.