

Internet Chat Protokolle

Christoph Ciesla
(ciesla@in.tum.de)

Seminar „Internet Measurement“,
Technische Universität München

SS 2005 (Version vom 24. Juli 2005)

Zusammenfassung

Diese Arbeit behandelt Verfahren zur Analyse von Internet-Chat-Systemen. Dabei wird besonders auf Methoden zur Identifikation von Chat-Verbindungen eingegangen; damit wäre es möglich, die Chat-Verbindungen eingehender zu analysieren und somit Rückschlüsse auf die Chat-Benutzer und ihr Verhalten zu ziehen – ein Aufwand, der sich angesichts der immer größer werdenden Beliebtheit dieser Kommunikationsform sicher lohnen wird. Zunächst werden die charakteristischen Eigenschaften von Internet-Chat-Systemen und Chat-Verbindungen ermittelt und näher untersucht. Mit Hilfe der dadurch gewonnenen Kenntnisse wird dann ein Verfahren entwickelt, das Chat-Verbindungen auf Grund dieser charakteristischen Eigenschaften in einem Verbindungsprotokoll zu finden vermag. Zuletzt werden die Genauigkeit und die Zuverlässigkeit dieses Verfahrens untersucht und einige Testergebnisse vorgestellt.

1 Einleitung

Mit der rasanten Verbreitung des Internets in den letzten Jahren hat auch der Internet-Chat in seinen verschiedenen Ausprägungen rapide an Bedeutung hinzugewonnen und erfreut sich gerade bei jungen Leuten zunehmender Beliebtheit. War es früher nur wenigen vorbehalten, sich online zu unterhalten, so ist es heute für nahezu jedermann kein Problem mehr, sich via Internet mit Freunden, Kollegen, entfernt lebenden Verwandten oder einfach wildfremden Menschen zu unterhalten. Dabei wird es als unschätzbare Vorteil empfunden, dass die Anonymität bei Bedarf gewahrt bleibt. Das geht sogar so weit, dass manche Leute einen Großteil ihrer Freizeit mit Chatten verbringen. Studien belegen, dass Internet-Chat durchaus süchtig machen kann ([Young 97]).

Die Gründe dafür sind folgende: In der heutigen Zeit ist es ohne großen Aufwand möglich, Zugang zu Chats zu erlangen, ohne konkrete Kenntnisse über das jeweilige System zu benötigen. Dies gilt für Web-Chats (z.B. Antenne-Bayern-Chat [Antenne]) ebenso wie für Instant Messaging Systeme wie ICQ [ICQ], AIM [AIM], MSN [MSN] oder Jabber [Jabber]: die Konfiguration der Client-Software ist – falls überhaupt notwendig – oft bereits nach wenigen Mausklicks abgeschlossen. Dagegen sind im Internet Relay Chat (IRC [IRC 93]), einem der ältesten Chat-Systeme im Internet, eher die versierteren Benutzer anzutreffen, die sich ein wenig damit auskennen und wissen, wie die Client-Software zu bedienen ist und wo sie Zugang zu einem IRC-Netz bekommen – Fragen,

mit denen sich die meisten Benutzer von Web-Chats in der heutigen Zeit nicht mehr auseinandersetzen (wollen).

Zunächst stellt sich die Frage, was an Chat-Systemen so interessant sein soll, wo doch eher kleine Datenmengen übertragen werden. Motivierend wirkt jedoch die Tatsache, dass die Gruppe der Chat-Benutzer in den letzten Jahren beträchtlich angewachsen ist; es lohnt sich also durchaus, Chat-Verbindungen zu analysieren, um daraus Rückschlüsse auf die Benutzer und ihr Verhalten zu ziehen.

Diese Arbeit beschäftigt sich vor allem mit der Frage, wie Chat-Verbindungen vom restlichen Netzwerkverkehr herausgefiltert werden können, was angesichts der Vielfalt der verschiedenen Systeme nicht einfach ist. Es gibt zum einen den IRC, dessen Protokoll wohldefiniert und in RFC 1459 ([IRC 93]) klar spezifiziert ist und auf einem eigenen TCP-Port arbeitet. Daneben gibt es sog. Instant Messaging Systeme wie ICQ, AIM, MSN oder Jabber, deren Haupteinsatzzweck der Versand von Kurzmitteilungen ist. Auch diese Systeme arbeiten mit klar spezifizierten Protokollen und auf eigenen Ports.

Bei den bisher angesprochenen Systemen ist es wegen der eigenen Protokolle nicht schwer, den von ihnen hervorgerufenen Netzwerkverkehr vom übrigen Netzwerkverkehr heraus zu filtern. Schwieriger wird es schon bei den verschiedenen Web-Chat-Systemen: Sie arbeiten jeweils mit eigenen oft nur dürftig oder überhaupt nicht dokumentierten Protokollen, von denen es meist keine frei verfügbaren Spezifikationen gibt. Darüber hinaus gibt es Web-Chat-Systeme, die HTTP als Transportprotokoll verwenden; solche Verbindungen aus den übrigen HTTP-Verbindungen herauszufiltern, ist besonders schwierig, da derselbe TCP-Port verwendet wird wie für die Übertragung von Webseiten und über das Chat-Protokoll selbst oft so gut wie nichts bekannt ist.

In [DWF 03] wird ein Verfahren vorgestellt, das von Wissenschaftlern der Universität des Saarlandes und der Technischen Universität München entwickelt worden ist, mit dem Ziel, in einem über eine bestimmte Zeit aufgezeichneten Verbindungsprotokoll die Chat-Verbindungen zu finden. Dabei wurde folgender Ansatz verfolgt: Zunächst wurde der Netzwerkverkehr aufgezeichnet und anhand charakteristischer Merkmale die Paketströme identifiziert, die für Chat in Frage kommen; Paketströme, die nicht in Frage kommen, wurden aussortiert.

Das Verfahren wurde im Dezember 2002 getestet. Über den Zeitraum einer Woche wurde der gesamte Netzwerkverkehr der Universität des Saarlandes aufgezeichnet, und die Chat-Verbindungen wurden extrahiert. Anhand der Ergebnisse wurde dann die Genauigkeit des Verfahrens untersucht, um es später weiter zu optimieren.

2 Die verschiedenen Ausprägungen von Internet-Chat

Das folgende Kapitel beschäftigt sich mit den Eigenschaften der verschiedenen Ausprägungen von Internet-Chat. Zunächst werden anhand von IRC typische Eigenschaften von Chat-Verbindungen ermittelt, die sich auf andere Systeme wie Web-Chat übertragen lassen. Danach wird auf spezielle Eigenschaften von Web-Chats und ihre Eignung zum automatischen Filtern eingegangen. Zuletzt wird ein knapper Überblick über Instant-Messaging-Systeme gegeben.

2.1 Internet Relay Chat (IRC)

Der Internet Relay Chat (IRC, [IRC 93]) ist das wahrscheinlich älteste Chat-System, das heute noch sehr weit verbreitet ist; es existiert seit 1988 und ist damit älter als das WWW, das erst 1989 entstanden ist.

Die Architektur von IRC ist eine typische Client-Server-Architektur (Abbildung 1); mehrere IRC-Server können zu einem Netzwerk zusammengeschaltet werden. Ein Client verbindet sich in der Regel über TCP-Port 6667 mit einem Server und verwendet einen

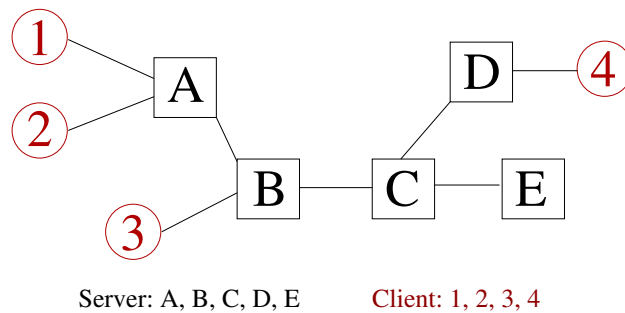


Abbildung 1: Einfaches Beispiel für ein IRC-Netz

eindeutigen Namen (Nickname) zur Identifikation innerhalb des IRC-Netzes. Der Nickname kann geändert werden, muss jedoch IRC-Netz-weit eindeutig sein.

Das IRC Protokoll erlaubt die Eingabe von speziellen Kommandos, um z.B. private Mitteilungen zu versenden. Mehr Informationen dazu sind in der RFC 1459 [IRC 93], in der das IRC-Protokoll spezifiziert ist, zu finden.

Die Kommunikation findet in sog. Discussion Channels statt, von denen es in jedem IRC-Netz sehr viele zu den verschiedensten Themen gibt. In jedem Channel gibt es in der Regel einen (oder mehrere) sog. Channel Operator, einen Benutzer mit besonderen Berechtigungen, wie z.B. Ändern des Themas oder Festlegen von Channel-weit gültigen Richtlinien und Attributen. Es steht jedem Benutzer frei, selbst Channels zu erzeugen und damit Operator dieser Channels zu sein. Neben dem Chatten erlaubt der IRC auch die direkte Übertragung von Dateien zwischen den Clients.

IRC-Chat-Verbindungen zu erkennen ist aufgrund des eigenen Ports relativ einfach. Da der Dateitransfer über einen anderen Port und mit einem eigenen Protokoll abgewickelt wird, ist sichergestellt, dass nur Text- und Kontrollnachrichten erfasst werden.

Eine wichtige Eigenschaft, die IRC-Verbindungen von anderen TCP-Verbindungen unterscheidet, ist die Verteilung der Paketgrößen (siehe dazu Abbildung 7): Es werden größtenteils kleine Pakete übertragen, was nicht weiter wundert, da Chat-Mitteilungen in der Regel maximal ein paar hundert Bytes groß sind. Diese Eigenschaft lässt sich also auch auf Web-Chat-Systeme übertragen.

Ebenfalls typisch für IRC ist der Versand von Keep-Alive-Nachrichten (PINGs und PONGs), die die Zeitdifferenz zwischen zwei Nachrichten (interarrival time) begrenzen.

2.2 Web-Chat

Inzwischen gibt es äußerst populäre Alternativen zum IRC: Web-Chats erlauben es den Benutzern, sich ohne großen Aufwand online zu unterhalten. Es genügt, im Webbrowser eine bestimmte Seite aufzurufen und sich anzumelden, und schon kann es losgehen.

Es gibt jedoch viele verschiedene Web-Chat-Systeme, die mit eigenen Protokollen arbeiten, welche meistens nur wenig oder gar nicht dokumentiert sind. Das macht die korrekte Erkennung der Chat-Verbindungen umso schwieriger. Deshalb ist es nötig, sich genauer mit Web-Chat-Systemen und ihren Eigenschaften zu befassen.

Web-Chats können in drei Klassen eingeteilt werden:

HTML-Web-Chats benutzen den Web-Browser als Oberfläche und werden meist durch server- und clientseitige Skripte realisiert. HTML-Web-Chat-Verbindungen zu erkennen ist sehr schwierig, da HTTP ([HTTP 99]) als Transportprotokoll verwendet wird und die Pakete über Port 80 übertragen werden. Es müssen also Chat-Verbindungen von Webseiten unterschieden werden. Jedoch weisen HTML-Web-Chat-Verbindungen bestimmte charakteristische Eigenschaften auf, die man sich zum Filtern zunutze machen kann:

HTTP erlaubt zum einen das Zwischenspeichern (Caching) von Webseiten. Dies ist für Chat-Verbindungen vollkommen ungeeignet, so dass es durch Cache Control Headers wie *Cache-Control: no-cache* oder *Cache-Control: no-store* unterbunden wird. Weiterhin ist HTTP zustandslos; es müssen also zusätzliche Informationen wie z.B. sog. „Session IDs“ mit übertragen werden. Zum Erkennen von Chat-Verbindungen ist es sinnvoll, Pakete mit diesen Eigenschaften zu suchen. Andererseits weisen Pakete, die nur mit geringer Wahrscheinlichkeit zu Chat-Verbindungen gehören, ebenfalls charakteristische Eigenschaften auf: Da HTML-Web-Chats auf Caching gänzlich verzichten, ist es unwahrscheinlich, dass Cache Control Headers wie *Cache-control: must-revalidate* (siehe [HTTP 99]) in Chat-Verbindungen Verwendung finden.

Applet-Web-Chats benutzen ein Java-Applet, das beim Aufruf automatisch heruntergeladen und gestartet wird, und arbeiten meist mit einem eigenen Protokoll. Statt einer niemals endenden Webseite verwenden Applet-Chats ein Applet-Fenster als Front-End. Zum Aufbau der Verbindung muss der Chat-Teilnehmer also zunächst ein Java-Applet von einem Server laden und starten. Beobachtungen zeigen, dass dabei erstaunlich häufig die Zeichenfolge „chat“ im URL vorkommt. In manchen Fällen kommt das Wort „chat“ sogar in den übertragenen Paketen selbst vor. So ist es ansatzweise möglich, die Chat-Verbindungen ohne konkrete Kenntnisse über das Protokoll zu erkennen.

Applet-IRC-Chats stellen eine Untermenge der Applet-Web-Chats dar: Sie verwenden kein eigenes, sondern das IRC-Protokoll.

2.3 Instant Messaging Protokolle

Neben IRC und Web-Chat-Systemen gibt es noch die sog. Instant-Messaging-Systeme wie ICQ [ICQ], AIM [AIM], MSN [MSN] oder Jabber [Jabber], deren Haupteinsatzzweck der Versand von Kurzmitteilungen ist. Sie bieten im Gegensatz zu Web-Chats und IRC weitere Funktionen an: z.B. können Benutzer eine Liste mit Kontakten anlegen und werden benachrichtigt, sobald jemand aus der Liste sich am System anmeldet.

Leider sind keine offiziellen Dokumentationen zu den Protokollen von ICQ, AIM und MSN frei verfügbar, jedoch ist darüber allgemein so viel bekannt, dass es ohne größere Schwierigkeiten möglich ist, von ihnen hervorgerufenen Netzwerkverkehr zu erfassen und aufzuzeichnen. In [OSCAR] gibt es z.B. eine inoffizielle Dokumentation zu dem Protokoll, das bei ICQ und AIM zum Einsatz kommt. Anders verhält es sich bei Jabber: Das von Jabber verwendete Protokoll (Extensible Messaging and Presence Protocol, XMPP) basiert auf XML und ist in den RFCs 3920-3923 [XMPP 04] spezifiziert.

3 Extrahieren von Chat-Verbindungen

Im folgenden Abschnitt wird ein Verfahren entwickelt, das es ermöglicht, aus den über einen Zeitraum aufgezeichneten Verbindungen die Chat-Verbindungen zu finden.

Zunächst müssen alle Chat-Verbindungen erfasst werden. Es werden also alle Pakete gesammelt, wobei die Ports 0 bis 1023 (außer Port 80 für HTTP) und andere Ports, von bekannt ist, dass sie nicht von Chat-Systemen benutzt werden, ignoriert werden.

Zusammengehörende Pakete werden dann in bidirektionale Paketströme gruppiert. Die Pakete werden einzeln untersucht und abhängig von Startpunkt, Ziel und Ports den einzelnen Verbindungen zugeordnet; dabei werden die Pakete, die zwischen zwei Hosts über dieselben Ports übertragen werden, zusammengefasst ([DWF 03]). Diese Paketströme werden dann anhand der genannten Kriterien gefiltert; dabei ist es sinnvoll, zuerst die Regeln anzuwenden, die erwartungsgemäß sehr viele nicht in Frage kommenden Paketströme verwerfen. Gut dafür geeignet ist z.B. der Paketgrößen-Filter. Chat-Verbindungen werden zwar von eher kleinen Paketen dominiert, das bedeutet jedoch nicht, dass keine größeren vorkommen können. Der Filter muss also die Verteilung der

Paketgrößen berücksichtigen. Beispielsweise können alle Verbindungen markiert werden, von denen mindestens 50% aller Pakete einen festen Grenzwert unterschreiten.

Bevor alle nicht markierten Verbindungen gelöscht werden, werden zunächst alle Paketströme, die in mindestens einem Paket eine HTTP-GET-Anforderung für eine JAR-Datei (Java-Applet) enthalten, als „applet-flows“ markiert. Paketströme, die in mindestens einem Paket das Wort „chat“ enthalten, werden als „chat-word-flows“ markiert. Danach werden alle nicht markierten Verbindungen gelöscht.

Kandidaten für HTML-Web-Chat-Verbindungen lassen sich anhand der Header ermitteln: Verbindungen mit passenden Cache-Control-Headern (siehe dazu 2.2) werden markiert, solche mit ungeeigneten Headern werden aussortiert; weiterhin ist es sinnvoll, sicherzustellen, dass es sich beim angeforderten Objekt um ein HTML-Dokument und nicht etwa um ein Bild handelt.

Ähnlich ist es möglich, Applet-Web-Chat-Verbindungen zu erkennen: ein Paketstrom wird markiert, wenn ein „applet-flow“ oder ein „chat-word-flow“ zwischen demselben Client und Server vorausgegangen ist.

In einem letzten Schritt müssen jetzt nur noch die Verbindungen entfernt werden, von denen bekannt ist, dass sie zu anderen Diensten gehören, die nichts mit Chat zu tun haben. Dazu gehören z.B. SSH-, FTP- oder SMTP-Verbindungen, die nicht auf den jeweiligen Standard-Ports abgewickelt werden. Alle weiteren ungeeigneten Paketströme können mit Hilfe bekannter typischer struktureller Eigenschaften von Chats aussortiert werden; so ist es beispielsweise unwahrscheinlich, dass über Verbindungen, die weniger als 30 Sekunden andauern, gepochet wird – die kurze Zeit reicht für eine Unterhaltung nicht aus. Eine andere wichtige Annahme ist, dass natürlich von beiden Seiten Daten verschickt werden müssen.

Nach Anwendung aller dieser Filterregeln sind am Ende nur noch die Paketströme vorhanden, die mit hoher Wahrscheinlichkeit von Chats her stammen. Wie es um die Präzision dieses Verfahrens bestellt ist, muss im Folgenden untersucht werden.

4 Erprobung des Verfahrens

Der folgende Abschnitt beschäftigt sich mit einem ausführlichen Test des Verfahrens und einer Analyse der Testergebnisse.

4.1 Testläufe

Das besprochene Verfahren wurde 2002 an der Universität des Saarlandes getestet (siehe [DWF 03]). Eine Woche lang wurde der Netzwerkverkehr des Campus aufgezeichnet, gefiltert und analysiert. Zunächst bemühte man sich, die anfallende Datenmenge von 960GB möglichst schon während der Aufzeichnung zu reduzieren und eine vollständige Speicherung der Rohdaten zu vermeiden. Dazu wurden parallel zur Aufzeichnung vereinfachte Versionen des Paketgrößen-Filters sowie der Überprüfung auf „Applet-flows“ und „Chat-word-flows“ angewandt. Nicht markierte Verbindungen wurden gelöscht, um die vollständigen Versionen derselben Filter auf die übrigen Verbindungen anzuwenden und wieder nicht markierte Pakete zu löschen. Die anwendungsabhängigen Filter und die strukturellen Regeln wurden zuletzt offline angewandt.

Nach dem Testlauf am Campus der *Universität des Saarlandes* betrug die Größe des resultierenden Datensatzes, hier einfach *WEBCHAT1* genannt, nur noch 238MB.

4.2 Abschätzung der Vollständigkeitsrate – *RECALL*

RECALL ist definiert als die Wahrscheinlichkeit, dass eine Web-Chat-Verbindung von unserem Verfahren gefunden wird, d.h. als das Verhältnis der gefundenen zu allen im

Verbindungsprotokoll enthaltenen Web-Chat-Verbindungen.

Definition von *RECALL*:

$$r = \frac{|RELEVANT \cap FOUND|}{|RELEVANT|} = 1 - \frac{|MISSED|}{|RELEVANT|}$$

wobei *RELEVANT* für die Anzahl von Web-Chat-Verbindungen, *FOUND* für die der gefundenen und *MISSED* für die der nicht gefundenen Web-Chat-Verbindungen steht.

Um eine untere Schranke für *RECALL* zu bekommen, musste zunächst eine untere Schranke für *RELEVANT* sowie eine obere Schranke für *MISSED* ermittelt werden. Dazu wurden über kurze Zeit zwei weitere Datensätze, *SELCHAT* und *WEBCHAT2* erstellt. Für *SELCHAT* wurden Verbindungen zu bekannten Chat-Servern aufgezeichnet; für *WEBCHAT2* wurden zur gleichen Zeit alle Verbindungen aufgezeichnet und der Filter darauf angewandt. Durch manuelles Überprüfen wurden dann aus *SELCHAT* alle Web-Chat-Verbindungen gezählt, was eine untere Schranke für *RELEVANT* lieferte. Eine obere Schranke für *MISSED* wurde ermittelt, indem alle Web-Chat-Verbindungen gezählt wurden, die in *SELCHAT* enthalten waren und in *WEBCHAT2* nicht erkannt worden waren. Zuletzt zog man noch alle Verbindungen ab, von denen sicher ist, dass sie keine Chat-Verbindungen sind.

Testläufe und Auswertungen in [DWF 03] ergaben eine „Vollständigkeitsrate“ von

$$r = 1 - \frac{|MISSED|}{|RELEVANT|} > 91,7\%$$

was bedeutet, dass im Mittel 91,7% aller Web-Chat-Verbindungen erfasst werden, bzw. dass eine Web-Chat-Verbindung mit 91,7% Wahrscheinlichkeit erkannt wird. Werden verschlüsselte Web-Chat-Verbindungen nicht mitgezählt, so liegt *RECALL* bei 93,7%.

4.3 Abschätzung der Genauigkeit – *PRECISION*

Eine weitere interessante Größe ist die Genauigkeit des Verfahrens. *PRECISION* ist definiert als die Wahrscheinlichkeit, dass eine erfasste Verbindung auch wirklich eine Web-Chat-Verbindung ist.

Definition von *PRECISION*:

$$p = \frac{|RELEVANT \cap FOUND|}{|FOUND|}$$

wobei *RELEVANT* für die Anzahl der Web-Chat-Verbindungen und *FOUND* für die Anzahl der erkannten Web-Chat-Verbindungen steht.

Da die Anzahl der Web-Chat-Verbindungen unbekannt ist und für eine manuelle Überprüfung die Datenmengen viel zu groß sind, bedient man sich der Wahrscheinlichkeitsrechnung und bestimmt den Erwartungswert:

$$E[p] = E\left[\frac{|RELEVANT \cap FOUND|}{|FOUND|}\right] = 1 - E\left[\frac{|WRONG|}{|FOUND|}\right]$$

WRONG ist dabei die Anzahl der irrtümlich erfassten Verbindungen, die jedoch keine Web-Chat-Verbindungen sind.

Da es jedoch verschiedene Arten von Web-Chat-Systemen gibt, deren Verbindungen nach verschiedenen Kriterien gefiltert werden (siehe Abschnitt 2.2), ist es sinnvoll, jede Klasse für sich zu betrachten. Man bestimmt also für alle Kategorie die jeweiligen Erwartungswerte und berechnet aus ihnen einen Durchschnittswert.

Im Rahmen der Tests aus [DWF 03] wurden die Web-Chat-Verbindungen zunächst grob in HTML-Web-Chat- und Applet-Web-Chat-Verbindungen eingeteilt. Diese großen

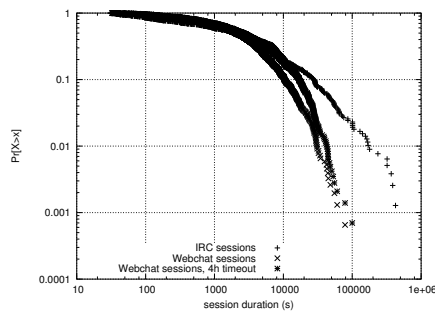


Abbildung 2: CCDF-Diagramm der Sitzungsdauer

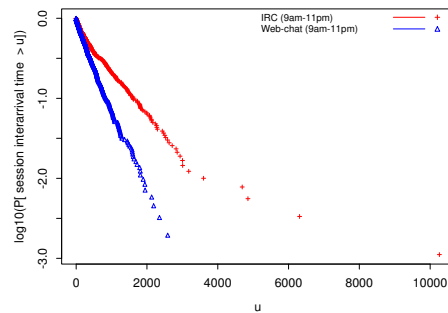


Abbildung 3: CCDF-Diagramm der Session-Interarrival-Zeiten

Gruppen wurden danach noch jeweils feiner untergliedert. Die Gesamtgenauigkeit ergab sich schließlich zu

$$p = \frac{|RELEVANT \cap FOUND|}{|FOUND|} = 93,1\%$$

was bedeutet, dass 93,1% aller erfassten Verbindungen Chat-Verbindungen sind.

4.4 Ergebnisse der Tests

Zuletzt wurden die statistischen Eigenschaften von Internet-Chat-Verbindungen untersucht, mit dem Ziel, die charakteristischen Eigenschaften von IRC-Verbindungen mit denen von Web-Chat-Verbindungen zu vergleichen. Im Gegensatz zu anderen Systemen wie FTP, Web oder Telnet wurden die statistischen Eigenschaften von Internet-Chat bisher kaum untersucht. Genauere Kenntnisse darüber sind jedoch nützlich zur Planung und zum Design von Kommunikationsnetzwerken.

Zunächst muss jedoch klar sein, dass manche Begriffe in Bezug auf Web-Chat eine andere Bedeutung haben als in Bezug auf IRC. Die Dauer einer IRC-Sitzung ist sehr einfach zu definieren als die Zeit vom Herstellen bis zum Trennen der Verbindung. Bei Web-Chats ist das nicht so einfach, da es dort schon einmal passieren kann, dass eine neue TCP-Verbindung z.B. beim Wechsel in einen anderen Channel aufgebaut wird. Eine Web-Chat-Sitzung kann z.B. sinnvoll definiert werden als alle Verbindungen vom selben Client zum selben Server. Auf der Basis dieser Definition ist es nun möglich, IRC-Sitzungen und Web-Chat-Sitzungen miteinander zu vergleichen.

Desweiteren wurde überprüft, ob die statistischen Eigenschaften von Internet-Chat denen von anderen Systemen ähnlich sind; besonderes Augenmerk wurde dabei auf die Verteilung von Sitzungsdauern gelegt, die beispielsweise bei FTP und Telnet heavy-tailed ist. Außerdem wurden die Interarrival-Zeiten von Sitzungen bzw. Verbindungen untersucht, die bei FTP und Telnet exponentialverteilt sind (bei Web jedoch nicht!). Zuletzt wurden die Verteilung der Packet-Interarrival-Zeiten, die Verteilung der Paketgrößen und die Menge der übertragenen Pakete bzw. Bytes näher betrachtet.

4.4.1 Sitzungsdauer

Abbildung 2 zeigt die CCDF¹ der Sitzungsdauer auf einer doppelt logarithmischen Skala. Zunächst fällt dabei auf, dass IRC-Sitzungen allgemein länger zu dauern scheinen als Web-Chat-Sitzungen. Der Mittelwert für beide Systeme liegt bei etwa einer halben

¹CCDF = complementary cumulative distribution function – komplementäre kumulative Verteilungsfunktion

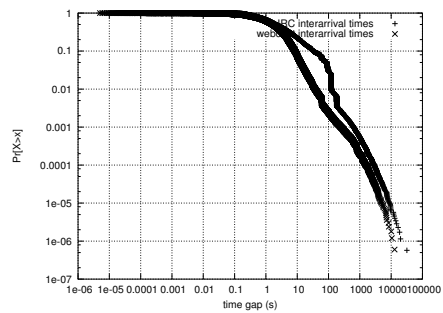
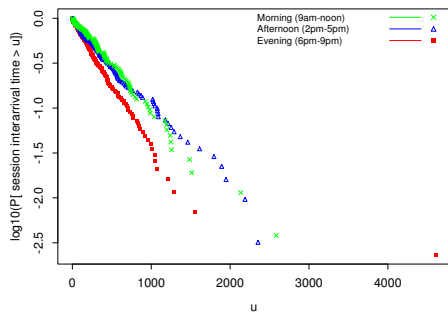


Abbildung 4: CCDF der Session-Interarrival-Zeiten für verschiedene Tageszeiten **Abbildung 5:** CCDF der Packet-Interarrival-Zeiten

Stunde. Lediglich 1% der Web-Chat-Sitzungen dauern länger als acht Stunden, während dies auf 10% aller IRC-Sitzungen zutrifft; 2% der IRC-Sitzungen dauern sogar länger als einen Tag, einige davon bestanden während der gesamten Aufzeichnungsperiode. Dies kann dadurch erklärt werden, dass sich Web-Chat-Benutzer zu einem größeren Teil über Nacht von dem System abmelden, als das bei IRC-Benutzern der Fall ist. Das bedeutet jedoch auch, dass die Verteilung der Sitzungsdauern keineswegs der Länge der Unterhaltungen selbst entspricht; gerade über länger dauernde Sitzungen wird oftmals nur ab und zu wirklich gechattet, sehr häufig sind solche Verbindungen für längere Zeit inaktiv.

4.4.2 Interarrival-Zeiten von Sitzungen

Als nächstes werden die Session-Interarrival-Zeiten untersucht, also die Zeitabständen, nach denen die Benutzer jeweils Verbindungen zu Chat-Servern herstellen; dabei wird jedoch nur der Zeitraum zwischen 9 Uhr und 23 Uhr berücksichtigt (wobei zu beachten ist, dass die Interarrival-Zeiten tagsüber deutlich kürzer sind als nachts). Für jede Verbindung wird ermittelt, wann sie hergestellt worden ist; danach werden für alle folgenden Anmeldungen die Interarrival-Zeiten ermittelt.

Abbildung 3 zeigt die CCDF der Sitzungs-Interarrival-Zeiten für IRC und Web-Chat auf einer logarithmischen Skala. Beide Linien sind mehr oder weniger gerade, was auf eine Exponentialverteilung schließen lässt. Um ein noch genaueres Bild zu bekommen, können die Interarrival-Zeiten für Chat-Sitzungen noch einmal auf mehrere Tageszeiten (Vormittag, Nachmittag, Abend) aufgeteilt werden. Abbildung 4 zeigt die jeweiligen Verteilungen für Web-Chat-Sitzungen. Auch hier ist deutlich erkennbar, dass die Interarrival-Zeiten exponentialverteilt sind.

4.4.3 Interarrival-Zeiten von Chat-Mitteilungen

Eine weitere interessante Frage ist, ob diese Verteilung der Interarrival-Zeiten auch innerhalb von Sitzungen gilt. Deshalb ist es sinnvoll, die Interarrival-Zeiten von Chat-Nachrichten zu untersuchen. Um sie zu ermitteln, müssen zunächst die Pakete mit Chat-Nachrichten von anderen Paketen unterschieden werden können; unter anderem sind Pakete, die keine Daten enthalten (wie z.B. TCP-ACKs), zu ignorieren. Unter der Annahme, dass alle Chat-Nachrichten in je einem Paket übertragen werden können, entsprechen dann die Interarrival-Zeiten der Pakete denen der Chat-Nachrichten.

Mit etwas Vorsicht sind jedoch die periodisch auftretenden Nachrichten zu betrachten. Wie IRC verwenden auch viele Web-Chat-Systeme sog. Keep-Alive-Nachrichten, um Verbindungs-Timeouts zu vermeiden. Auch wenn Keep-Alive-Nachrichten für IRC ignoriert werden, sind bei IRC weitere periodisch auftretende Nachrichten zu beobachten, die unter anderem auf die Benutzung des ISON-Kommandos (zur Überprüfung,

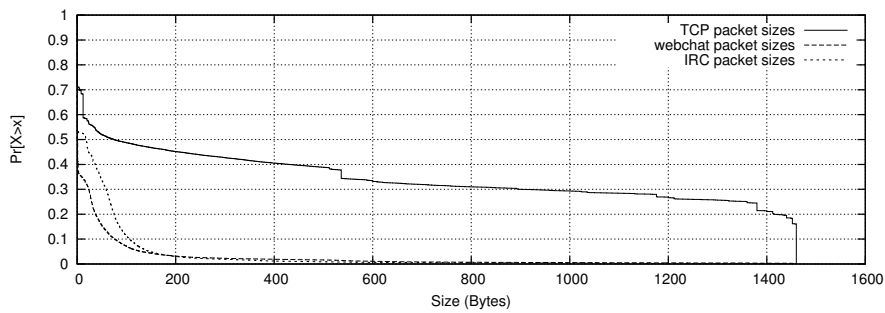


Abbildung 6: Verteilung der Paketgrößen

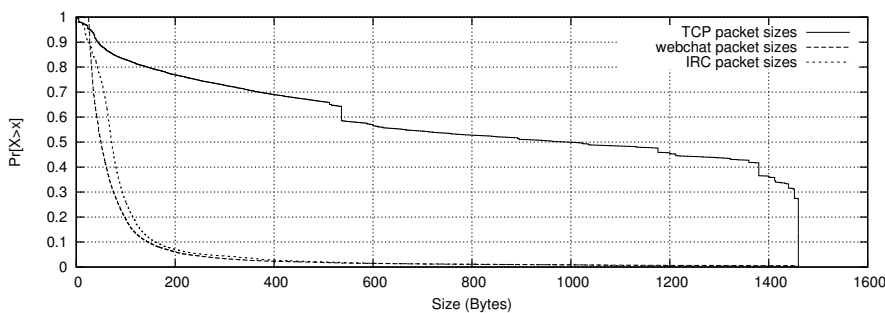


Abbildung 7: Verteilung der Paketgrößen (jeweils ohne Acknowledgements)

ob Benutzer mit bestimmten Nicknamen im IRC-Netz angemeldet sind; siehe [IRC 93]) zurückzuführen sind. Auf Grund der Vielfalt der verschiedenen Web-Chat-Systeme ist es nicht möglich, Keep-Alive-Nachrichten herauszufiltern, weshalb diese bei der Untersuchung mit berücksichtigt wurden.

Die CCDF der Packet-Interarrival-Zeiten ist in Abbildung 5 auf einer doppelt logarithmischen Skala eingetragen. Wie darauf deutlich zu erkennen ist, entspricht die Verteilung der Packet-Interarrival-Zeiten keineswegs einer Exponentialverteilung, sondern vielmehr einer Heavy-tailed-Verteilung. Dass bei Web-Chat, wo Keep-Alive-Nachrichten berücksichtigt worden sind, sehr lange Interarrival-Zeiten (mehrere Minuten bis Stunden) auftreten, ist ein Anzeichen dafür, dass bei weitem nicht alle Web-Chat-Systeme einen Keep-Alive-Mechanismus verwenden, um ihre Verbindungen aufrecht zu erhalten. Wie auf der Abbildung außerdem zu erkennen ist, sind die Interarrival-Zeiten für IRC-Chat deutlich länger als für Web-Chat; das hängt auch damit zusammen, dass IRC-Sitzungen in der Regel länger dauern als Web-Chat-Sitzungen und häufig über längere Zeiträume inaktiv sind. Es bedeutet jedoch nicht, dass in Web-Chat-Systemen mehr Mitteilungen versandt werden als im IRC.

4.4.4 Paketgrößen

Betrachtet man die Verteilung der Paketgrößen für TCP, IRC und Web-Chat getrennt (Abbildung 6), so fällt auf, dass die Pakete bei Chat-Verbindungen im Mittel deutlich kleiner sind als bei anderen Verbindungen. Der Mittelwert der Paketgröße für TCP-Verbindungen beträgt etwa 113 Bytes, wohingegen dieser Mittelwert für IRC- und Web-Chat-Verbindungen wesentlich kleiner ist. Nach Entfernen aller Pakete, die keine Daten enthalten, ist umso deutlicher erkennbar, dass Chat-Verbindungen im Gegensatz

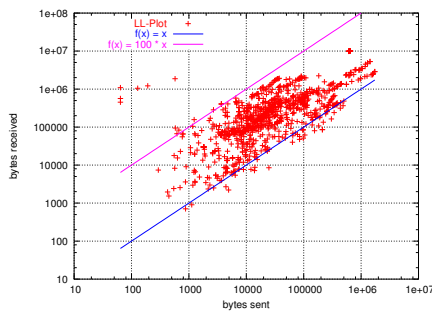


Abbildung 8: Empfangene und gesendete Bytes (IRC)

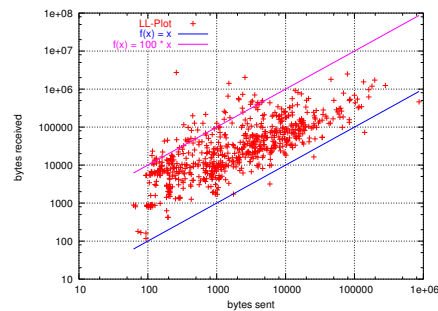


Abbildung 9: Empfangene und gesendete Bytes (Web-Chat)

zu anderen TCP-Verbindungen von kleinen Paketen dominiert werden (Abbildung 7). Während in TCP-Verbindungen 70% aller Pakete größer als 300 Byte sind, trifft dies auf weniger als 8% der Pakete in Chat-Verbindungen zu. Andererseits sind mehr als die Hälfte aller Pakete in Chat-Verbindungen kleiner als 100 Bytes; dieser Anteil ist bei TCP-Verbindungen allgemein deutlich kleiner.

4.4.5 Übertragene Bytes pro Sitzung

Zuletzt wird das Verhältnis der gesendeten zu den empfangenen Bytes pro Chat-Sitzung untersucht. Trägt man die Anzahl gesendeter und empfangener Bytes in ein Diagramm mit doppelt logarithmischer Skala ein, so ergibt sich ein Bild wie in den Abbildungen 8 und 9. Im Mittel empfängt ein Client zehn mal so viel wie er sendet, wobei sich das Verhältnis von gesendeten zu empfangenen Bytes zwischen 1 : 1 und 1 : 100 bewegt.

Dabei gibt es auch einige Ausnahmen: Manche Clients empfangen sehr viel mehr (Faktor $\gg 100$) als sie senden. Andere empfangen deutlich weniger, als sie senden, was zunächst überrascht, angesichts der Tatsache, dass der Server alles, was ein Benutzer sendet, auch wieder zu ihm zurückschickt. Im Fall von Web-Chat ist eine mögliche Erklärung der Aufbau einer neuen HTTP-Verbindung für jede Zeile, die der Benutzer abschickt; wenn ein Benutzer überdurchschnittlich viel schreibt, kann das bereits ausreichen. Für den IRC ist eine mögliche Erklärung dafür exzessiver Versand (weitestgehend unbeantworteter) privater Nachrichten sowie Nutzung der ISON-Funktion ([IRC 93]).

5 Zusammenfassung

In dieser Arbeit wurde ein Verfahren vorgestellt, mit dem Chat-Verbindungen aus dem übrigen Netzwerkverkehr extrahiert werden können, wobei höchstens 8,3% aller Chat-Sitzungen nicht erkannt werden und mindestens 93,1% aller erfassten Verbindungen Chat-Sitzungen entsprechen. In Anbetracht der überwältigenden Vielfalt verschiedener Chat-Systeme ist diese Bilanz nicht schlecht. Des Weiteren ist man nun in der Lage, statistische Aussagen über Internet-Chat zu machen, was Rückschlüsse auf die Chat-Benutzer selbst und ihr Verhalten zulässt.

Inhalt zukünftiger Forschungen muss es nun sein, dieses Verfahren so zu optimieren und zu verfeinern, dass die Größen *RECALL* und *PRECISION* noch günstigere Werte annehmen und das Verfahren dadurch noch trennschärfer wird.

Literatur

- [DWF 03] Christian Dewes, Arne Wichmann, Anja Feldmann: *An analysis of Internet chat systems*, Internet Measurement Conference, Oktober 2003
<http://www.imconf.net/imc-2003/papers/chat11.ps>
- [Young 97] K. S. Young: *What Makes the Internet Addictive: Potential Explanations for Pathological Internet Use*, August 1997
<http://www.netaddiction.com/articles/habitforming.htm>
- [Antenne] *Antenne-Bayern-Chat*
<http://www.antenne.de/antenne/chat/chat.html>
- [IRC 93] J. Oikarinen, D. Reed: *Internet Relay Chat Protocol RFC 1459*, Mai 1993
<http://www.faqs.org/rfcs/rfc1459.html>
- [ICQ] ICQ Inc.: *What is ICQ?*
<http://www.icq.com/products/whatisicq.html>
- [AIM] AOL Inc.: *AOL Instant Messenger*
<http://www.aim.com>
- [MSN] Microsoft Corp.: *MSN Messenger*
<http://www.msn.com>
- [Jabber] Jabber Software Foundation: *Jabber: Open Instant Messaging and a Whole Lot More, Powered by XMPP*
<http://www.jabber.org>
- [XMPP 04] P. Saint-Andre, Jabber Software Foundation:
Extensible Messaging and Presence Protocol (XMPP): Core RFC 3920, Oktober 2004
<http://www.faqs.org/rfcs/rfc3920.html>
Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence RFC 3921, Oktober 2004
<http://www.faqs.org/rfcs/rfc3921.html>
Mapping the Extensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging (CPIM) RFC 3922, Oktober 2004
<http://www.faqs.org/rfcs/rfc3922.html>
End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP) RFC 3923, Oktober 2004
<http://www.faqs.org/rfcs/rfc3923.html>
- [OSCAR] Adam Fritzler: *AIM/OSCAR Protocol Specification*
<http://www.oilcan.org/oscar/>
- [HTTP 99] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee: *Hypertext Transfer Protocol – HTTP/1.1 RFC 2616*, Juni 1999
<http://www.faqs.org/rfcs/rfc2616.html>