Eine Übersicht von Verfahren und Anwendungen zur Klassifizierung von Netzwerkverkehr

Yaroslaw Medyany und Ruben Niclasen
Hochschule Bonn-Rhein-Sieg
Studiengang Informatik
Grantham-Allee 20
53757 Sankt Augustin
Seminar SoSe 2015

***Zusammenfassung*—Für die Betreiber eines Telekommunikationsnetzwerks ist es wichtig, dass das Netzwerk die Performanz-Anforderungen der Kunden erfüllt. Mit Hilfe der Internet-Traffic-Analyse kann das aktuelle Nutzerverhalten in einem Netzwerk untersucht und der Netzwerkverkehr klassifiziert werden, um in der Folge das Netzwerk für das gegebene Nutzerverhalten zu optimieren. Die Klassifizierung der Verkehrsdaten ist dabei eine nichttriviale Aufgabe. In dieser Arbeit werden Anforderungen an Tools zur Klassifizierung von Echtzeit- und Nicht-Echtzeit-Netzwerkverkehr vorgestellt und eine Auswahl von Tools anhand dieser Anforderungen untersucht.**

I. Einleitung

Die Internet-Traffic-Analyse befasst sich unter anderem mit der Frage, welche Anwendungen den beobachteten Traffic erzeugt haben. Diese Klassifizierung des Traffics ist besonders für Telekommunikationsnetzbetreiber interessant, da so das Netzdesign für die genutzten Anwendungen optimiert werden kann oder die Netzwerkpakete bestimmter Anwendungen priorisiert werden können. Ein weiterer Anwendungsfall können gesetzliche Vorgaben sein, die das Herausfiltern von Traffic mit illegalem Inhalt erfordern. Auch in der Informationssicherheit kann eine solche Klassifizierung zur Detektion von möglicherweise schädlichem oder anderweitig ungewolltem Traffic eingesetzt werden [1].

Für das Problem der Klassifizierung wurden mit der Zeit verschiedene Lösungsansätze entwickelt. So war es in der Anfangszeit des Internets üblich, die Port-basierte Klassifizierung zu nutzen, die den Verkehr anhand der von den Paketen der Transportschicht genutzten Ports und der Liste der Standardports der Internet Assigned Numbers Authority (IANA) klassifizierte. Diese Methode ist inzwischen allerdings sehr unzuverlässig, da viele Anwendungen zufällige Ports oder Ports, die normalerweise für andere Protokolle vorgesehen sind, nutzen, um die Kontrollmechanismen der Netzbetreiber zu umgehen. Ein anderer Ansatz ist die Payload-basierte Klassifizierung, die sich grob in die Kategorien Deep Packet Inspection (DPI) und Stochastic Packet Inspection (SPI) unterteilen lässt. Hier wird der konkrete Inhalt eines Pakets untersucht. Den nächsten Ansatz bildet die statistische Klassifizierung. Hier werden statistische Merkmale – genannt Features – von Netzwerk-Flows beobachtet und mit Hilfe von Machine-Learning-Algorithmen eingeordnet. Der letzte Ansatz ist die verhaltensbasierte Klassifizierung. Diese untersucht die Verbindungen zwischen den Kommunikationspartnern im Traffic. Diese Verfahren teilen Traffic nicht in einzelne Anwendungen ein, sondern in Anwendungsklassen, wie z.B. P2P oder HTTP [1].

Eine abstraktere Klassifizierung, die ähnlich zu den vorhergehenden Klassifizierungsarten, Rückschlüsse auf die von der Applikation bzw. vom Anwender geforderte Performanz an das Netzwerk darlegt, ist die Klassifizierung nach Echtzeit- und Nicht-Echtzeit-Verkehrsklassen. In dieser Arbeit wird untersucht, ob eine solche Klassifizierung mit aktuellen Tools möglich ist.

Es werden zunächst die unterschiedlichen Klassifizierungsmethoden und -Problematiken im Detail vorgestellt. Danach werden die zu klassifizierenden Verkehrsklassen definiert und konzeptionell modelliert. Im Anschluss wird eine Auswahl an Tools auf die Erfüllung von zuvor erstellen Anforderungen untersucht.

II. Aktuelle Methoden zur Klassifizierung

In diesem Kapitel werden verschiedene Lösungsansätze zur Klassifizierung vorgestellt. Diese haben je nach Anwendungsfall Vor- und Nachteile. Muss ein passendes Verfahren zur Klassifizierung gewählt werden, so lohnt es sich, die Verfahren auf die in [1] definierten Eigenschaften Granularität, Zeitlichkeit und Rechenaufwand zu untersuchen. Die Granularität kann dabei fein oder grob sein, je nachdem, ob der Traffic in einzelne Anwendungen oder eher in Anwendungsgruppen klassifiziert werden kann. Die Zeitlichkeit beschreibt, wie schnell die Klassifizierung erfolgen kann. Dies kann nach einigen Sekunden sein, nach einer bestimmten Anzahl von Paketen oder sogar erst nach Beendigung einer Kommunikationsverbindung. Der benötigte Rechenaufwand ist vor allem je nach Größe des untersuchten Netzwerks ein wichtiger Faktor [1].

A. Port-basierte Verfahren

Port-basierte Verfahren zur Traffic-Klassifizierung stellen die simpelsten Methoden dar. Solche Implementierungen betrachten die Transmission Control Protocol (TCP)- oder User Datagram Protocol (UDP)-Header der Pakete im zu untersuchenden Traffic und klassifizieren diesen mit Hilfe der „well-known ports“-Liste der IANA anhand der verwendeten Ports. Ein Vorteil dieser Methode ist die Einfachheit, da die verwendeten Ports effizient ausgelesen werden können und somit eine schnelle Klassifizierung möglich ist. Port-basierte

Verfahren sind in der heutigen Zeit aber nicht mehr zuverlässig, da Anwendungen entweder häufig unbekannte oder zufällig gewählte Ports verwenden oder die eigenen Pakete hinter Ports anderer Anwendungen/Zwecke verbergen. So nutzen zum Beispiel Peer-to-Peer (P2P)-Protokolle häufig zufällige Ports und die Voice-over-IP (VoIP)-Anwendung Skype nutzt, falls über zufällig gewählte UDP-Ports größer 1024 keine Verbindung aufgebaut werden kann, die TCP-Ports 443 und 80, die normalerweise für die Protokolle HTTPS und HTTP verwendet werden [2]. Laut [3] können durchschnittlich lediglich 30%-70% von Traffic mit Hilfe der Port-Analyse korrekt klassifiziert werden. Zusätzlich funktioniert dieser Ansatz nicht, sobald Network Address Port Translation (NAPT) genutzt wird [3][1].

B. Payload-basierte Verfahren

Um das Problem der Portverschleierung zu umgehen, können Payload-basierte Verfahren verwendet werden. Hier werden nicht nur die Paket-Header untersucht, sondern auch der Paketinhalt, die sogenannte Payload. Es wird zwischen zwei Verfahren unterschieden – der Deep Packet Inspection (DPI) und der Stochastic Packet Inspection (SPI) [1].

Bei der DPI wird in der Payload nach bekannten Mustern, Schlüsselwörtern oder anhand von regulären Ausdrücken gesucht. Die Informationen aus der Payload werden mit einer Signatur-Datenbank verglichen. Bei einem positiven Ergebnis können demnach aus den Payload-Informationen Rückschlüsse auf die zugrunde liegende Anwendung gezogen werden. DPI bietet einerseits eine sehr hohe Genauigkeit, erfordert gleichzeitig allerdings viele Ressourcen, da der Inhalt jedes einzelnen Pakets untersucht und Vergleiche erstellt werden müssen. Weitere Nachteile sind, dass bei verschlüsselten Paketen DPI nicht angewendet werden kann und dass die genutzten Schlüsselwörter oder regulären Ausdrücke für jede Anwendung manuell abgeleitet oder angepasst werden müssen [1]. Bekannte DPI-Tools sind z.B. OpenDPI, das darauf basierende nDPI, HiPPIE und L7-filter [3].

Die SPI versucht dagegen anhand von statistischen Eigenschaften der Payloads die Pakete einzuteilen. So existieren Verfahren, die nur eine bestimmte Anzahl der ersten Payload-Bytes als Features für Machine-Learning-Algorithmen nutzen. SPI erfordert dabei etwas weniger Ressourcen als DPI, da die auf die Payload angewendeten, statistischen Operationen günstiger sind, als pures Pattern-Matching [1]. Im Gegensatz zu DPI gibt es bei SPI Verfahren, die bei teilverschlüsselten Paketen angewendet werden können, wie z.B. das in [4] vorgestellte Verfahren, bei dem mit einem Chi-Quadrat-Test die Zufälligkeit des ersten Payload-Bytes untersucht wird, um für jedes Anwendungsprotokoll ein eigenes Syntaxmodell zu erstellen [1]. Der Chi-Quadrat-Test untersucht dabei die Anpassungsgüte von aufgezeichneten Testdaten gegenüber einer Chi-Quadrat-verteilten Zufallsvariable [4]. In [5] wird dagegen ein Verfahren vorgestellt, das anhand der Entropie der ersten Bytes der Payload eines Paketes zumindest eine grobe Klassifizierung erreichen kann – so spricht beispielsweise eine niedrige Entropie für reinen Text, eine mittelhohe Entropie entspricht Binärinformationen und eine hohe Entropie deutet auf eine verschlüsselte Payload hin. Obwohl dadurch eine nur sehr grobe Klassifizierung erreicht werden kann und Anwendungen mit einer hohen Wahrscheinlichkeit Payloads aller drei Arten (Text, Binär, Verschlüsselung) verwenden, können die gewonnenen Informationen dennoch beispielsweise für die Priorisierung von Paketen verwendet werden [1]. Als Beispiel wird in [1] das Priorisieren von Anwendungsaktualisierungen, die in binär-codierten Paketen verschickt werden und der Fehlerbehebung dienen, genannt.

C. Statistische Verfahren

Statistische Verfahren untersuchen den Paketfluss, den sogenannten Flow, auf bestimmte signifikante statistische Eigenschaften. Zu einem Flow werden dabei alle Pakete mit dem gleichen 5-Tupel, bestehend aus den Attributen Transportschicht-Protokoll, Quell-IP-Adresse und -Port und Ziel-IP-Adresse und -Port, gezählt [6]. In Verbindung mit Data Mining-Verfahren und Machine Learning-Algorithmen werden durch diese Verfahren Statistiken über Flows erstellt, untersucht und der Traffic wird klassifiziert. Zu den untersuchten Eigenschaften gehören beispielsweise die Größe und die Häufigkeit verschickter Pakete. Gegenüber den Payload-basierten Verfahren benötigen statistische Verfahren weniger Ressourcen, da hier die Pakete nicht im Detail betrachtet werden müssen, liefern aber in der Regel auch weniger genaue Ergebnisse [1]. Aufgrund der Irrelevanz der Payloads entfällt die Verschlüsselungsproblematik. Die statistischen Verfahren lassen sich in supervised- und unsupervised-Verfahren unterteilen: Erstere Verfahren benötigen eine Lernphase, in der mit Hilfe von vorklassifizierten Trainingsdaten die Features bestimmt werden, letztere Verfahren funktionieren auch ohne eine solche Phase, jedoch muss zusätzlich ein Labeling der automatisch generierten Klassen durchgeführt werden [6] [1].

D. Verhaltensbasierte Verfahren

Verhaltensbasierte Verfahren untersuchen den Netzwerkverkehr eines Hosts oder eines Endpunktes in einem Netzwerk auf unterschiedliche Muster. Dabei wird versucht, Anwendungen durch bestimmte Verhaltensmuster in dem von ihnen erzeugten Flow zu identifizieren. Zu den betrachteten Verhaltenseigenschaften zählen unter anderem die Anzahl der kontaktierten Hosts, die Anzahl der verwendeten Ports oder das auf der Transportschicht verwendete Protokoll. Unter Betrachtung dieser Eigenschaften wird beispielsweise deutlich, dass ein P2P-Host im Gegensatz zu einem Webserver eine Reihe von Hosts mit jeweils unterschiedlichen Ports anspricht, wogegen der Webserver von unterschiedlichen Clients zeitgleich kontaktiert wird, sodass sich die verschiedenen Anwendungen identifizieren lassen. Ähnlich wie für die statistischen Verfahren ist die Payload auch für die verhaltensbasierten Verfahren irrelevant, sodass letztere auch bei verschlüsselten Payloads funktionieren und effizienter als DPI- oder SPI-Verfahren arbeiten [1].

III. Problematiken der Klassifizierung

Sobald Netzwerkverkehr verschleiert wird, wird die Klassifizierung — unabhängig von der Art und Weise der Verschleierung — zum Problem. Im Folgenden werden zwei zentrale Problematiken, das Port-Wrapping und die Verschlüsselung und mögliche Lösungsansätze erläutert.

A. Problematiken

Eine zentrale Problematik ergibt sich durch die Nutzung eines Ports für verschiedene Anwendungen und Zwecke. Ein prominentes Beispiel hierfür ist der Messenger Skype, der die TCP-Ports 80 und 443 als Ausweichlösung gegen das Blocken der eigenen Pakete durch restriktive Firewalls nutzt [7]. Die genannten TCP-Ports werden in der Regel für HTTP- bzw. HTTPS-Verkehr genutzt. Diese Problematik wird im Folgenden Port-Wrapping genannt. Durch das Port-Wrapping lassen sich Port-basierte Klassifizierungsverfahren nicht mehr anwenden, da hier die Ergebnisse stark verfälscht werden [8].

Ein weiterer Störfaktor für die Klassifizierung ist die Verschlüsselung der Pakete. Durch die Verschlüsselung von Payloads können Methoden, welche die Paketinhalte analysieren (beispielsweise DPI-Methoden) nicht mehr verwendet werden [9]. Es muss allerdings zwischen verschiedenen Verschlüsselungsansätzen unterschieden werden. So gibt es zunächst die Verschlüsselung nach einem Handshake, wie z.B. im Falle von STARTTLS. Hier werden zunächst Klartext-Nachrichten ausgetauscht, bevor der Traffic verschlüsselt wird. Dies kann eine Klassifizierung anhand der Klartext-Nachrichten ermöglichen. Als nächstes existiert der Ansatz, nur die Payload zu verschlüsseln, wie zum Beispiel beim P2P-Protokoll BitTorrent. Eine Klassifizierung ist bei solchen Protokollen anhand des Headers möglich. Ein weiterer Ansatz ist das Verschlüsseln der Payload und zusätzlich von Teilen des Headers, wie z.B. bei dem Extensible Messaging and Presence Protocol (XMPP). Der letzte Ansatz ist das Verschlüsseln des kompletten Flows, wie z.B. bei Transport Layer Security (TLS) oder bei Virtual Private Network (VPN)-Protokollen wie dem Internet Protocol Security (IPSec) [3].

B. Lösungsansätze

Um das Problem des Port-Wrappings zu lösen, bedarf es komplexerer Methoden, als der Port-basierten Klassifizierung. Die bereits vorgestellte DPI-Methode funktioniert lediglich, falls der Netzwerkverkehr nicht verschlüsselt übertragen wird und falls es möglich ist, die Pakete tiefgehend zu betrachten [3]. Der Einsatz von statistischen Methoden ist dagegen auch bei verschlüsselten Paketen möglich. Allerdings sind statistische Methoden insbesondere verglichen mit der DPI ungenau [10]. Denkbar wäre eine Kombination aus unterschiedlichen Ansätzen, um auch verschleierten Traffic mit einer hohen Genauigkeit zu analysieren und zu klassifizieren.

Als Beispiel für solch einen hybriden Ansatz wird in [11] der sogenannte Signature Statistic Port Classifier (SSPC) vorgestellt. SSPC besteht aus einem Port-basierten, einem statistischen und einem Signatur-basierten Modul. Alle drei Module arbeiten unabhängig und analysieren den eingespeisten Netzverkehr auf ihre eigene Art und Weise. Das Port-basierte Modul vergleicht die vorhandenen Ports mit einer Port-Datenbank und speichert das Ergebnis. Das statistische Modul entscheidet nach vorher definierten Regeln über die Klassifizierung. Das Signatur-basierte Modul vergleicht den vorhandenen Flow mit einer Signatur-Datenbank.

Der SSPC-Algorithmus berechnet die finale Klassifikation nach unterschiedlich priorisierten Fällen:

1) Das Signatur-basierte Modul liefert ein Ergebnis (erste Priorität).
2) Keines der drei Module liefert ein Ergebnis – der Netzverkehr gilt als nicht-klassifizierbar (zweite Priorität).
3) Das statistische und das Port-basierte Modul liefern das selbe Ergebnis, das Signatur-basierte Modul liefert kein Ergebnis (dritte Priorität).
4) Nur das statistische Modul liefert ein Ergebnis (vierte Priorität).
5) Nur das Port-basierte Modul liefert ein Ergebnis (fünfte Priorität).

SSPC funktioniert sowohl mit sogenannten Offline-, als auch mit Online-Paketen. Offline-Pakete werden zur Analyse zwischengespeichert, während Online-Pakete während der Netzwerkübertragung analysiert werden. Laut [11] arbeitet SSPC mit einer Genauigkeit zwischen 82% und 92%.

IV. Definition von Verkehrsklassen

Die Art der Klassifizierung und Kategorisierung des im Netzwerk vorhandenen Verkehrs ist von dem Zweck der Traffic-Analyse abhängig. Um Analysen zur qualitativen Nutzung des Netzwerkes durchzuführen bietet sich eine Kategorisierung nach unterschiedlichen Applikationen an. In dieser Arbeit soll der Netzverkehr allerdings quantitativ und damit grobgranular, insbesondere zur Optimierung des Quality of Service analysiert und klassifiziert werden.

Das ATM-Forum definiert in [12] fünf unterschiedliche Service-Klassen, welche sich zunächst in Echtzeit- und Nicht-Echtzeit-Services unterteilen. Echtzeit-Services sind dabei solche Anwendungen, die möglichst geringe Verluste (cell loss) und Verzögerungen (cell delay) während der Übertragung benötigen.

Für Echtzeit-Services werden zwei Kategorien definiert. Die Constant Bit Rate (CBR) erfordert eine kontinuierlich hohe, verlustfreie Bandbreite für die gesamte Dauer der Verbindung. Die Real-Time Variable Bit Rate (rt-VBR) kommt bei Anwendungen zum Einsatz, die eine variable und schwankend starke Bandbreite benötigen. Die Bandbreite wird für rt-VBR durch die Parameter Peak Cell Rate (PCR), Sustainable Cell Rate (SCR) und Maximum Burst Size (MBS) beschrieben. PCR beschreibt einen maximalen Wert für die Zellenrate einer Übertragung, SCR beschreibt dafür einen mittleren Wert. MBS definiert die höchste Anzahl an Bits, die während einer Übertragung gesendet werden [12].

Für Nicht-Echtzeit-Services werden drei Kategorien definiert. Die Non-Real-Time Variable Bit Rate (nrt-VBR) klassifiziert, ähnlich wie die rt-VBR, variable und schwankend starke Bandbreiten für Nicht-Echtzeit-Anwendungen. Ebenso werden auch für die nrt-VBR-Klasse Werte für PCR, SCR und MBS bestimmt. Die Available Bit Rate (ABR) klassifiziert den Traffic für Anwendungen, die eine minimale und eine maximale Bandbreite spezifizieren. Diese Werte werden durch die PCR und die Minimum Cell Rate (MCR), welche einen minimalen Wert für die Zellenrate einer Übertragung angibt, bestimmt. Die minimale Bandbreite darf dabei durch das Netzwerk nicht unterschritten werden. Die letzte Service-Kategorie ist die Unspecified Bit Rate (UBR), die einen Best-Effort-Service darstellt. Für die UBR-Kategorie existieren keine Verlust- oder Verzögerungsanforderungen. Unter diese

Kategorie fallen beispielsweise Dateiübertragungen oder E-Mail-Verkehr [12].

Die Kategorien des ATM-Forums sind auf Asynchronous Transfer Mode (ATM)-Netze ausgelegt. Über ATM-Netze werden Zellen mit einer festen Größe von 53 Bytes (48 Bytes für die Payload, 5 Bytes für den Header) übertragen. In dieser Arbeit werden allerdings Ethernet-Netze betrachtet. Über diese werden Pakete unterschiedlicher Größen versendet und es entfallen ATM-spezifische Werte, wie beispielsweise die PCR oder SCR [13]. Aufgrund dessen werden auf Basis der ATM-Spezifikation zum Zwecke der Verkehrs-Klassifizierung die drei Klassen Real-Time Constant Packet Length (rt-CPL), Real-Time Variable Packet Length (rt-VPL) und Non-Real-Time Unspecified Packet Length (nrt-UPL) definiert. Diese werden in Tabelle I beschrieben. Die Definitionen von Real-Time und Non-Real-Time entsprechen dabei denen des ATM-Forums, wobei Real-Time-Anwendungen minimale Verzögerungs- und Verlustquoten benötigen und Non-Real-Time-Anwendungen ohne solche Anforderungen arbeiten [12].

rt-VPL	Der Verkehr wird von einer Echtzeit-Anwendung generiert und besteht aus Paketen mit einer variablen Paketlänge auf der Transportschicht, welche durch eine untere und eine obere Schranke begrenzt wird.
rt-CPL	Der Verkehr wird von einer Echtzeit-Anwendung generiert und besteht aus Paketen mit einer konstanten Paketlänge auf der Transportschicht.
nrt-UPL	Der Verkehr wird von einer Nicht-Echtzeit-Anwendung generiert und besteht aus Paketen mit einer variablen Paketlänge ohne Schranken auf der Transportschicht.

Tabelle I. DEFINITION VON VERKEHRSKLASSEN

Ein Beispiel für Verkehr, den die Klasse rt-VPL enthält, sind Echtzeitübertragungen von Audio- oder Videodaten unter der Verwendung mehrerer Codecs, die beispielsweise je nach verfügbarer Bandbreite adaptiv verwendet werden. Ein prominenter Vertreter von Anwendungen, die diese adaptive Technik nutzen und solchen Traffic generieren, ist Skype [14]. Je nach genutzten Codecs können auch andere VoIP-Programme, die das Realtime Transport Protocol (RTP) nutzen, z.B. Session Initiation Protocol (SIP)-Programme [15], zu dieser Klasse gehören. Ein anderes Beispiel für Anwendungen mit Traffic von unterschiedlichen Paketlängen und Anspruch auf geringe Paketverluste und -Verzögerungen sind Online-Videospiele, wie z.B. Counter-Strike [16].

Im Gegensatz zu Verkehr aus rt-VPL haben die Pakete des rt-CPL-Verkehrs eine konstante Paketlänge. Dies kann beispielsweise Verkehr sein, der von Audiostream- oder Videostreamanwendungen ohne eine adaptive Funktion generiert wird. Auch hier kann je nach Codec RTP verwendet werden – allerdings unter Verwendung von nur einem Codec, durch den Pakete mit einer konstanten Paketlänge erzeugt werden [17]. In [18] wird zur Analyse der Quality of Service (QoS)-Unterstützung von Real-Time-Übertragungen in IEEE 802.1e-basierten Netzwerken unter anderem Real-Time-Verkehr erzeugt, der aus Audio- und Videoübertragungen mit konstanten Bitraten und Paketlängen besteht. Für die Audioübertragungen wurde dabei eine Bitrate von 65 kbit/s verwendet, was eine Paketlänge von 160 Bytes erzeugte. Für die Videoübertragungen wurde der Verkehr einer HDTV-Anwendung mit einer Bitrate von 20 Mbit/s und einer Paketlänge von 1024 Bytes erzeugt [18]. Ein solcher Traffic entspricht der Klasse rt-CPL.

Typische Anwendungen, die Verkehr für die Klasse nrt-UPL erzeugen sind unter anderem Web-Browsing, File Transfer Protocol (FTP)-Datenübertragungen oder das Übertragen von Videostreams auf Streaming-Plattformen wie YouTube oder Netflix [19]. Insgesamt enthält diese Klasse alle Anwendungen, die nicht zeitkritisch sind.

V. MODELLIERUNG VON VERKEHRSKLASSEN

Die in Kapitel IV definierten Verkehrsklassen können als Basis für die Modellierung von Netzwerkverkehr verwendet werden [20]. Für eine korrekte Modellierung der Verkehrsklassen werden Features benötigt, welche den zu analysierenden Verkehr für eine Zuordnung in die drei Klassen rt-VPL, rt-CPL und nrt-UPL detailliert beschreiben. Aus der Literatur wird im Folgenden eine Auswahl von Features abgeleitet und anhand von Beispielanwendungen in Verbindung zu den drei Verkehrsklassen aus Kapitel IV gebracht. Die Auswahl von Features ist maßgeblich für die Effektivität der Verkehrsklassifizierung, da irrelevante oder redundante Features oftmals die Genauigkeit der Ergebnisse verschlechtern [6].

A. Features

In der Literatur finden sich in vorgestellten Verfahren viele unterschiedliche Features, siehe [6] [20] [21] [22] [23] [24] [25]. Um die drei definierten Verkehrsklassen rt-CPL, rt-VPL und nrt-UPL modellieren zu können, wurden folgende Features ausgewählt:

- Paketlänge [6] [20] [21] [23] [24]
- Packet Interarrival Time [6] [20] [23] [24] [25]
- Übertragungsrichtung [20] [24]

Die Paketlänge ist dabei die komplette Länge eines TCP- oder UDP-Pakets in Bytes. Hier gibt es verschiedene Möglichkeiten der Betrachtung, einige davon sind: Minimale, maximale oder durchschnittliche Paketlänge des Flows, Paketlänge in Abhängigkeit zum vorherigen oder nachfolgenden Paket oder die Standardabweichung der Paketlängen [6].

Bei der Packet Interarrival Time handelt es sich um die Zeit, die zwischen der Ankunft zweier aufeinanderfolgender Pakete des gleichen Flows vergeht. Mögliche statistische Eigenschaften sind hier unter anderem die durchschnittliche Packet Interarrival Time, die Varianz in Vor- und Rückwärts-Richtung oder die Fourier-Transformation [6].

Die Übertragungsrichtung (engl. „packet direction“) definiert bei einer bidirektionalen Kommunikation Pakete, die entweder von einem Client zu einem Server oder von einem Server zu einem Client verschickt werden. Diese Information wird zusammen mit der Paketgröße relevant, da so erkennbar ist, wie viele Informationen aus der einen in die andere Richtung verschickt werden [20].

Diese drei Features genügen nach [6], [20] und [23] in gemeinsamer Kombination zur Darstellung der Real-Time- und Non-Real-Time-Klassen, da sich diese im Wesentlichen in der Paketlänge und in der durchschnittlichen Ankunftszeit verschickter Pakete unterscheiden [12].

B. Modellierung der definierten Verkehrsklassen

1) rt-VPL: Für die Klasse rt-VPL kann, wie bereits in Kapitel IV für die Definition der Verkehrsklasse vorgestellt, die VoIP-Anwendung Skype als Beispiel herangezogen werden. In [23] wurden unterschiedliche Ansätze zur Klassifizierung von Skype-Verkehr vorgestellt. Unter den vorgestellten Verfahren findet sich eine auf dem naiven Bayes-Klassifikator basierende Methode, für die zwei Features definiert wurden: Die Paketlänge und die sogenannte „Inter-Packet Gap“. Das letztere Feature beschreibt im klassischen Telekommunikationsumfeld den minimalen zeitlichen Abstand zwischen verschickten Paketen. In dieser Arbeit kann der Wert sinngemäß durch die Packet Interarrival Time ersetzt werden. Nach [23] eignet sich der naive Bayes-Klassifikator in Verbindung mit ausgewählten stochastischen Charakteristiken gut zur Klassifizierung von nicht-Skype-spezifischem Video-Übertragungs- und VoIP-Netzwerkverkehr mit entsprechenden Eigenschaften, da Skype sowohl Audio- als auch Video-Übertragungen anbietet und beide Übertragungsarten Netzwerkverkehr nach dem selben Schema erzeugen [23].

Ein weiteres Anwendungsbeispiel sind Online-Spiele. In [24] wurde der Verkehr des Massive Multiplayer Online Role-Playing Game (MMORPG) „ShenZhou Online“untersucht. Zur Darstellung des Verkehrs wurden die Features Paketlänge, Packet Interarrival Time und Übertragungsrichtung untersucht. Die Übertragungsrichtung wird dabei bei [24] ähnlich wie in [20] im Zusammenhang mit der Paketlänge und der Packet Interarrival Time betrachtet.

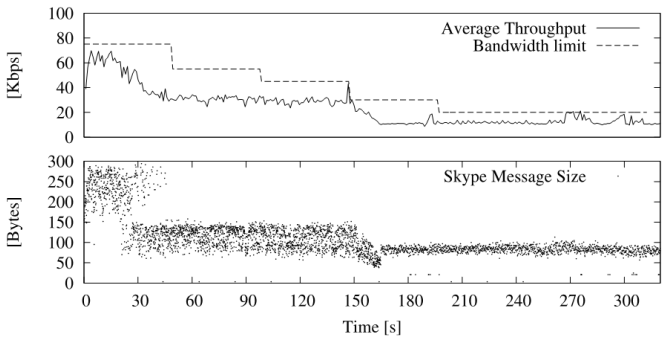

Abbildung 1. Durchschnittliche Bitrate und Nachrichtengröße während eines Skype-Anrufes [23]

Die Paketlänge ist in der Klasse rt-VPL variabel und wird durch nominale Parameter charakterisiert, im Falle von Skype ist dies die Codec-Rate. Skype wählt abhängig von der zur Verfügung stehenden Bandbreite Codecs, die wiederum unterschiedliche Paketgrößen erzeugen. Durch eine minimale und eine maximale Codec-Bitrate werden eine untere und eine obere Schranke für die Paketgröße definiert [23]. Abbildung 1 zeigt die durchschnittliche Bitrate eines Skype-Videoanrufes und die Gesamtgröße der verschickten Nachrichten. Es wird deutlich, wie die Paketlänge mit geringerer, verfügbarer Bandbreite abnimmt [23]. Im Falle des in [24] untersuchten MMORPG unterscheiden sich die Paketgröße zwischen den vom Client oder vom Server verschickten Paketen. 98% der „Client-zu-Server“-Pakete waren kleiner als 32 Bytes, wobei 52% der Pakete 27 Bytes und 36% der Pakete 23 Bytes klein waren. Die „Server-zu-Client“-Pakete waren im Durschnitt 114 Bytes groß [24]. Zwar wurde in [24] im Detail nur ein MMORPG untersucht, allerdings ergeben sich für andere Online-Spiele ähnliche Werte [24] [16]. Die Paketgröße wird durch die Gaußverteilung repräsentiert [23].

Die Packet Interarrival Time wird durch die Gaußverteilung modelliert. Bei der Packet Interarrival Time entsteht keine Varianz, da die Zeit zwischen den Paketen – beispielsweise im Falle von Skype – unabhängig von Parametern, wie dem verwendeten Codec ist [23]. In [24] wird beschrieben, dass die Packet Interarrival Time von „Server-zu-Client“-Paketen ebenfalls einen konstanten Wert von 200ms bei 50% des Verkehrs aufwies. Bei dem „Client-zu-Server“-Verkehr hängt die Zeit von den durchgeführten Aktionen des Spielers ab – Spieler, die keine Pausen während des Spielens einlegen, erzeugen ebenfalls konstante Werte [24].

Nach [23] genügen die beiden erläuterten stochastischen Merkmale Paketlänge und Packet Interarrival Time für eine Klassifizierung des Netzwerkverkehrs mit einer sehr geringen Fehlerrate.

2) rt-CPL: Die Klasse rt-CPL beschreibt Anwendungen, die möglichst geringe Paketverlust- und Paketverzögerungswerte benötigen [12]. Dies können beispielsweise Live-Audio- oder Live-Video-Übertragungen, sowie generell Anwendungen, die RTP nutzen, sein [25]. Im Gegensatz zu der zuvor beschriebenen Verkehrsklasse rt-VPL, zeichnet die Verkehrsklasse rt-CPL eine konstante Paketlänge während des gesamten Flows aus. Diese Konstanz kann beispielsweise in Bezug auf das Audio- und Video-Streaming dadurch hervorgerufen werden, dass die Anwendungen, anders als z.B. Skype, keine Anpassung der Paketgröße an die verfügbare Bandbreite unterstützen. In [25] wurde zur Verkehrsanalyse von Live-Streams entsprechender Verkehr simuliert. Dabei wurde insbesondere die Packet Interarrival Time auf ein Maximum von 200ms beschränkt, sodass der Verkehr einer Buffering-freien Übertragung entspricht [25]. Dieser Wert entspricht den Untersuchungen in [24], sodass die Packet Interarrival Time demnach auf die gleiche Art und Weise wie für rt-VPL dargestellt werden kann. Die Übertragungsrichtung ist für diese Klassifizierung nicht relevant.

3) nrt-UPL: nrt-UPL stellt einen „best effort“-Verkehr dar und enthält Web-Traffic wie das Besuchen von Webseiten, das Herunterladen von Dateien oder das Streamen von Audio- und Videodateien. Das Besuchen von Webseiten verursacht einen Netzwerkverkehr mit Paketen von unterschiedlichen Größen und Packet Interarrival Time-Werten. Das Herunterladen von größeren Datenmengen verursacht ebenfalls Netzwerkverkehr mit unterschiedlich großen Paketgrößen und Packet Interarrival Time-Werten [22]. In [19] wird der von YouTube verursachte Netzwerkverkehr beschrieben. Es wird deutlich, dass bei der Übertragung zunächst initial 40 Sekunden des angefragten Videos und im Anschluss Pakete mit einer Größe von 64 KB nach und nach übertragen werden. Eine ähnliche Vorgehensweise nutzt auch der Streaming-Dienst Netflix [19]. Aus diesen Werten ergibt sich, dass die Klasse nrt-UPL den Features die geringsten beziehungsweise keine Einschränkungen auferlegt. Sowohl Paketlänge, als auch Packet Interarrival Time sind stark variabel. Die Übertragungsrichtung ist für diese Klassifizierung nicht relevant.

VI. Anforderungen an die Klassifizierung

Um eine Klassifizierung des Netzwerkverkehrs zufriedenstellend durchzuführen, muss zunächst das passende Tool gefunden oder entwickelt werden. Im Folgenden werden sieben Anforderungen aufgestellt, welche die Wahl des passenden Tools vereinfachen sollen. Die Anforderungen basieren auf bereits gestellten Anforderungen aus der Literatur [26] [27] [28] und wurden für den Zweck dieser Arbeit angepasst.

A. Definition von Verkehrsklassen

Der Nutzer sollte eigene Verkehrsklassen durch die Verwendung unterschiedlicher Attribute (z.B. durch die Vorgabe statistischer Daten) definieren können. Dies fördert die Nutzbarkeit und Flexibilität des Tools in dem ohnehin sehr komplexen Feld der Verkehrsklassifizierung [26]. Insbesondere wird hier gefordert, dass die in Kapitel IV vorgestellten Verkehrsklassen in dem Tool definiert werden können. So müssen die Attribute Paketlänge, Packet Interarrival Time und Übertragungsrichtung definiert werden können. Aus diesen Attributen muss die Modellierung der drei Verkehrsklassen rt-CPL, rt-VPL und nrt-UPL nach der Modellierung aus Kapitel V erfolgen können.

B. Input/Automatisierung

Die Klassifizierung sollte, nach dem Start der Analyse durch den Anwender, von dem Tool automatisiert durchgeführt werden. Das Beziehen von Daten aus den zuvor definierten Datenquellen sollte ebenfalls automatisiert geschehen. Dies bezieht eventuell vorklassifizierte Testdaten für ein statistisches supervised-Verfahren mit ein. Als Datenquellen sollten das tcpdump-Format für gespeicherte Pakete zur Offline-Analyse und Netzwerkschnittstellen für die Online-Analyse unterstützt werden [26] [27].

C. Definition der Granularität

Der Nutzer sollte die Tiefe und den Detailgrad der Klassifizierung selbst bestimmen können. Während in der Fachliteratur sich oft die Anforderung, eine Klassifizierung auf Applikationsebene durchzuführen, findet (bspw. [2] [7] [8] [9]), wird in dieser Arbeit gefordert, dass das Tool nach den zuvor definierten Verkehrsklassen rt-CPL, rt-VPL und nrt-UPL klassifizieren kann. Diese Erweiterbarkeit ist in Anbetracht der Komplexität und Flexibilität der Netzwerkverkehrsanalyse förderlich [26].

D. Effizienz

In [28] wurde die Implementierung des Benchmark-Tools NeTraMark für Klassfizierungstools von Netzwerkverkehr vorgestellt. NeTraMark enthält unterschiedliche Klassifizierungs-Plugins, die auf C++, Java und Perl basieren und die zu untersuchenden Tools auf diese Weise einbinden. Es existieren Plugins für ein Payload-basiertes Tool (crl_pay), zwei Verhaltensbasierte Klassifizierer (BLINC, Traffic Dispersion), ein Port-basiertes Tool (CoralReef) und sieben Machine-Learning-Algorithmen (u.a. C4.5 und Naive Bayes). Als Anforderungen an den zugrunde liegenden Rechner wurden eine 2.4 GHz Intel Core 2 Quad Kentzfield Q6600 CPU und 4 GB Arbeitsspeicher angegeben. Auf einer solchen Maschine wurden die unterschiedlichen Tools mit einem Netzwerkverlauf von mehreren Gigabyte eines Universitäts-Campus getestet. Es wurden rund 92-158 K Flows pro 5 Minuten untersucht [28]. Da die Veröffentlichung auf das Jahr 2011 datiert, können die Anforderungen an einen heutigen Rechner angepasst und entsprechend erhöht werden. Das Klassifizierungstool sollte daher den Netzwerkverkehr eines mittelgroßen Unternehmens auf einem handelsüblichen Rechner (16 GB Arbeitsspeicher, Intel i7 3.4 GHz CPU) analysieren können.

E. Genauigkeit der Ergebnisse

Das Tool sollte Ergebnisse mit einer Genauigkeit von 85% liefern können. Diese Genauigkeit sollte von einer dritten Instanz oder dem Hersteller mit Hilfe einsehbarer, öffentlicher Tests bestätigt werden [26] [27].

F. Output/Bedienung

Das Tool sollte Ergebnisse sowohl textuell, als auch insbesondere grafisch darstellen können. Eine grafische Darstellung vereinfacht die Untersuchung der Ergebnisse und ermöglicht ein intuitives Verständnis dieser [26].

G. Online-/Offline-Analyse

Das Tool sollte Netzwerkverkehr sowohl online, als auch offline analysieren können. Abhängig von den zur Verfügung stehenden Ressourcen, ist es nötig Analysen mit Live-Daten online oder mit gespeicherten Daten offline durchzuführen. Ebenso ist die geforderte Analyse abhängig von den zur Verfügung stehenden Datenquellen [26].

VII. Evaluation von Tools zur Klassifizierung

In diesem Kapitel wird eine Auswahl von Tools zur Klassifizierung von Internet-Traffic anhand der in Kapitel VI vorgestellten Anforderungen evaluiert. Da eine umfassende Analyse aller auf dem Markt erhältlichen Tools nicht in dem zeitlichen Rahmen dieser Arbeit liegt, begrenzt sich die Tool-Auswahl auf drei Tools, die in der Fachliteratur mehrfach berücksichtigt wurden (siehe beispielsweise [9], [29], [30], [27], [28], [6]) und deren Basismethoden den in Kapitel II beschriebenen Payload-basierten, statistischen oder Verhaltensbasierten Verfahren entspricht. Für die Kategorie der Tools, die ausschließlich Port-basierte Methoden nutzen, wird in dieser Arbeit kein Tool untersucht, da dieser Ansatz in der heutigen Zeit wegen Port-Wrapping und zufällig genutzten Ports in der Regel unzuverlässige Ergebnisse liefert [1]. Die Evaluation stützt sich primär auf die veröffentlichte Literatur zu den Tools. Die in dieser Arbeit untersuchten Tools sind:

- nDPI: http://www.ntop.org/products/deep-packet-inspection/ndpi/
- netAI: http://caia.swin.edu.au/urp/dstc/netai/index.html
- BLINC: [31]

Dabei ist nDPI ein Vertreter der DPI-Kategorie, netAI ein Repräsentant der statistischen Verfahren und BLINC ein verhaltensbasiertes Tool.

Tabelle II zeigt die Ergebnisse der Evaluation. Die einzelnen Anforderungen werden qualitativ mit **+** (Anforderung erfüllt), **o** (Anforderung teilweise erfüllt), **-** (Anforderung nicht erfüllt) und **n.v.** (Information nicht vorhanden) bewertet. ✓ markiert das nach der Evaluation am besten geeignete Tool.

Anforderung\Tool	nDPI	netAI	BLINC
Def. Verkehrsklassen	-	+	o
Input/Automatisierung	o	+	o
Def. Granularität	o	+	+
Effizienz	o	n.v.	+
Genauigkeit	+	+	+
Output/Bedienung	o	+	o
Online-/Offline-Analyse	o	+	+
Ergebnis		✓	

Tabelle II. ZUSAMMENFASSUNG DER EVALUATIONSERGEBNISSE

nDPI erfüllt nur wenige der gestellten Anforderungen an ein Klassifizierungs-Tool, was größtenteils darauf zurückzuführen ist, dass nDPI im Kern ein DPI-Verfahren verwendet und lediglich eine Bibliothek ist, die erst durch die Einbindung in eine andere Software zu einem vollwertigen Klassifizierungs-Tool wird. Ein Beispiel für so eine Software ist ntopng (http://www.ntop.org/products/traffic-analysis/ntop/), das von den nDPI-Entwicklern stammt und unter anderem neben einem Web-Interface auch eine grafische Ergebnisdarstellung bietet [32].

netAI gewinnt dank der Einbindung der Tools NetMate (verantwortlich für die Gewinnung statistischer Features aus dem Netzwerkverkehr) und Weka (Bibliothek und Implementierung von Machine Learning-Algorithmen zur Auswertung der Features) an Flexibilität in Bezug auf die Feature- und Datenquellenauswahl und an Mächtigkeit in Bezug auf die Verarbeitung der Features [33]. netAI stellt ein geeignetes Tool zur Klassifizierung nach den in Kapitel IV definierten Verkehrsklassen dar.

BLINC ist dank der zugrundeliegenden verhaltensbasierten Technologie ein mächtiges Tool. Insbesondere die Flexibilität durch die Nutzung von definierbaren Verkehrsmustern und die komplexen Konfigurationsmöglichkeiten von 28 unterschiedlichen Parameter fallen positiv auf. Beides erfordert allerdings einen gewissen Aufwand, um eine korrekte Funktionalität und insbesondere eine hohe Genauigkeit der Ergebnisse zu erzielen [31] [29].

Die detaillierten Informationen zu den drei untersuchten Tools werden in den folgenden Unterkapiteln erläutert.

A. nDPI

Das Tool nDPI ist eine Open-Source-DPI-Bibliothek, die auf der nicht mehr weiterentwickelten OpenDPI-Bibliothek basiert. OpenDPI war dabei kein reines DPI Tool, da es neben reinem Pattern-Matching auch andere Techniken, wie z.B. statistische und verhaltensbasierte Techniken, nutzt [34]. Die Grundstruktur von nDPI gleicht immer noch der Grundstruktur von OpenDPI: Die core-library ist für die Dekodierung der IP-Pakete und die Extraktion von Metadaten verantwortlich während die plugin-dissectors die Detektion der einzelnen unterstützten Protokolle auf Applikationsebene übernehmen. Daneben erkannten die nDPI-Entwickler allerdings viele Schwächen in OpenDPI, die sie zu beseitigen versuchen. So erkennt nDPI mehr Protokolle als OpenDPI und kann anhand der Serverzertifikate auch bestimmte Dienste bei SSL-verschlüsseltem Traffic erkennen [35]. Es ist nochmal zu erwähnen, dass nDPI eine Bibliothek und kein eigenständiges Komplettprogramm ist.

1) Definition von Verkehrsklassen: Da nDPI eine Klassifizierung auf Applikationsebene durchführt, können hier von Haus aus keine Verkehrsklassen definiert werden. Die einzelnen Applikationen werden mit Hilfe der plugin-dissectors detektiert, wobei für jede Anwendung ein solcher dissector benötigt wird. nDPI unterstützt dabei über 170 Protokolle und bei Bedarf kann nDPI durch das Schreiben von eigenen dissectors um weitere Applikationen erweitert werden. Da nDPI keine statistischen Werte über alle Pakete eines Flows bildet, sondern jedes Paket einzeln betrachtet und bei einem match den gesamten Flow klassifiziert, kann hier ohne weiteres keine statistische Klassifizierung anhand von Paketlänge, Packet Interarrival Time und Übertragungsrichtung erfolgen [35]. Ein Workaround, um mit nDPI nach Verkehrsklassen kategorisieren zu können, wäre die Programmierung einer eigenen, nDPI nutzenden Software, die einzelne von nDPI detektierbare Anwendungsprotokolle bestimmten Verkehrsklassen zuordnet (z.B. Skype zu rt-VPL) und als Ergebnis dann die Klassifikation nach Verkehrsklassen ausgibt.

2) Input/Automatisierung: Als Input fordert nDPI einzelne Flows auf der dritten Ebene des OSI-Referenzmodells [34]. Dies muss also die Software gewährleisten, die nDPI einbindet. Wie allerdings an der Beispiel-Anwendung pcapReader.c, die im nDPI-Softwarepaket enthalten ist, zu sehen ist, kann die Eingabe von Verkehrsdaten per tcpdump-pcap-Datei oder per Netzwerkschnittstelle erfolgen [36]. Die Automatisierung hängt ebenfalls von der nDPI nutzenden Software ab.

3) Definition der Granularität: Die Granularität beschränkt sich bei nDPI auf die Applikationsebene [35]. Eine gröbere Klassifizierung ist, wie in Kapitel VII-A1 beschrieben wird, nur durch die nDPI aufrufende Software möglich, indem einzelne Anwendungen im Nachhinein einer Verkehrsklasse zugeordnet werden.

4) Effizienz: Laut [35] konnte nDPI 2014 mit Dualcore-Prozessor und Standardhardware den Internetverkehr einer 10 Gbit-Verbindung performant klassifizieren. Diese Angabe ist allerdings mit Vorsicht zu genießen, da das Paper von den nDPI-Entwicklern selbst stammt. In [34] aus dem Jahr 2015 dagegen heißt es, dass nDPI nicht „die beste Performanz besitzt“. Im Vergleich zu OpenDPI ist nDPI bei gleicher Hardware effizienter, außer bei der Detektion des RTP-Protokolls [3].

5) Genauigkeit der Ergebnisse: In [34] wurden sechs verschiedene DPI tools auf ihre Genauigkeit hin untersucht. Dazu wurden insgesamt 51,93 GB Testdaten in 750 K Flows erzeugt, wobei die Verkehrsdaten von populären Anwendungen stammen. nDPI hatte in diesem Test bei mehreren Anwendungen und Protokollen eine Genauigkeit von über 85% (BitTorrent (non-encrypted), Dropbox, FTP clients (active), Pando Media Booster, RDP clients, Skype (all), DNS, ICMP, IMAP STARTTLS, NETBIOS name service, NETBIOS session service, SAMBA session service, NTP, POP3 PLAIN, POP3 TLS, SMTP PLAIN, SOCKSv5, SSH), bei anderen aber 0% (z.B. League of Legends, IMAP TLS, Webdav).

Bemerkenswert ist auch, dass HTTP in nur 17,25% der Fälle richtig erkannt wurde. Insgesamt lieferte nDPI aber zusammen mit Libprotoident als Open-Source-Tools die besten Ergebnisse [34].

6) Output/Bedienung: Für die Darstellung der Ergebnisse und für die Bedienung ist die nDPI einbindende Software verantwortlich. Das Beispiel-Programm pcapReader.c stellt die Ergebnisse beispielsweise textuell dar [36].

7) Online-/Offline-Analyse: Auch hier ist zum Teil die nDPI nutzende Software verantwortlich. So hat pcapReader.c die Möglichkeit, die Pakete direkt an der Netzwerkschnittstelle abzufangen [36]. Ob nDPI tatsächlich den Verkehr auch in Echtzeit analysiert, hängt von der Menge der Eingabedaten ab, da ab einer bestimmten Menge von Daten die Effizienz von nDPI nicht mehr die zeitnahe Abarbeitung gewährleisten kann.

B. netAI

Mit Hilfe des Tools netAI lassen sich aufgenommener oder in Echtzeit mitgeschnittener Netzwerkverkehr analysieren und Verkehr-generierende Tools identifizieren. Abbildung 2 zeigt den Aufbau von netAI, inklusive der zwei vorhandenen Komponenten. Die Packet Classification-Komponente empfängt Online- oder Offline-Daten von Netzwerkschnittstellen bzw. aus zuvor gespeicherten Traces und extrahiert unterschiedliche statistische Features wie beispielsweise die Paketlänge oder die Packet Interarrival Time. Aus Trainigsdaten kann eine Verkehrsmodellierung erstellt werden, die den Classifier erzeugt und mit deren Hilfe die Machine Learning-Komponente vorhandene Applikationen aus dem zugrunde liegenden Traffic identifizieren kann. Alternativ kann statt der Modellierung und dem Classifier direkt mit Testdaten gearbeitet werden, was allerdings zu Leistungseinbußen führt. Nach der Identifikation der Applikationen durch den Machine Learning-Algorithmus können die Ergebnisse beispielsweise zur Einteilung des Netzwerkverkehrs in unterschiedliche QoS-Klassen genutzt werden [33].

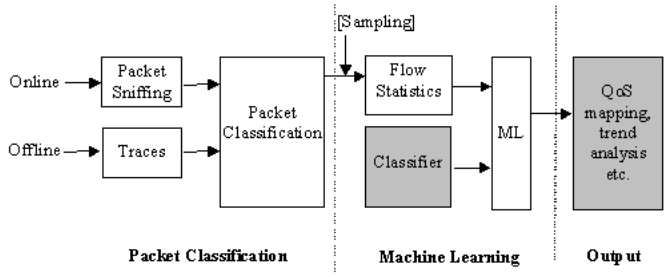

Abbildung 2. Architektur des netAI-Ansatzes [33]

Das Beziehen von Online- bzw. Offline-Daten wird durch das Tool NetMate Meter (http://sourceforge.net/projects/netmate-meter/) realisiert. Für die Machine Learning-Komponente wird die Machine Learning-Algorithmen-Bibliothek Weka (http://www.cs.waikato.ac.nz/ml/weka/) verwendet [33].

1) Def. Verkehrsklassen: Die Auswahl von Features erfolgt über NetMate. NetMate unterstützt 44 Features aus denen frei gewählt werden kann. Unter den unterstützten Features befinden sich unter anderem das 5-Tupel eines Flows, unterschiedliche Werte zu Paketlängen und der Packet Interarrival Time (Minimum, Mittel, Maximum, Standardabweichung) und die Anzahl von Paketen, die in bestimmte Richtungen verschickt wurden [37]. Zwar ist der Output von netAI die Identifikation von Applikationen, die in Kapitel V definierten Verkehrsklassen ließen sich dennoch durch die eigene Definition von Verkehrsmodellen klassifizieren.

2) Input/Automatisierung: Input für die Klassifizierung können Netzwerkschnittstellen für die Online-Analyse und Daten im tcpdump-Format für die Offline-Analyse sein. Testdaten können für die Netzwerkmodellierung verwendet werden [33].

3) Def. Granularität: Durch die Auswahl von Features, die NetMate bietet, und die Möglichkeit der Modellierung eigener Klassen mit Hilfe von Trainingsdaten, erhält der Nutzer die Kontrolle über die gewünschte Granularität der Klassifizierung [33] [37].

4) Effizienz: In der Veröffentlichung [38] aus dem Jahr 2006 wurde netAI auf einem Rechner mit Pentium 4 3 GHz CPU, 512 MB Arbeitsspeicher ausgeführt. Die Menge des damals untersuchten Traffics wurde nicht genannt. Da weder zu netAI, noch zu NetMate oder Weka offizielle Systemanforderungen oder Untersuchungen gefunden werden konnten, ist diese Anforderung zum jetzigen Zeitpunkt nicht bewertbar und sollte in einer weiterführenden Arbeit betrachtet werden.

5) Genauigkeit: Die Genauigkeit der Ergebnisse hängt von dem gewählten Machine Learning-Algorithmus ab. Unter [39] sind beispielhafte Ergebnisse der Klassifizierung mit netAI zu sehen. In der Untersuchung wurden acht Online-Spiele mit Hilfe der MA-Algorithmen C4.5, Naive Bayes, Bayessches Netz und NBTree untersucht. Dabei erzielten laut [39] die Algorithmen C4.5 und NBTree eine Genauigkeit von 95% bei der Identifikation der einzelnen Spiele und eine Genauigkeit von 99% bei der Unterscheidung von Spiele-Verkehr und anderem Verkehr.

6) Output/Bedienung: netAI bietet sowohl eine grafische Oberfläche als auch die Bedienung über die Kommandozeile. NetMate kann nur über die Kommandozeile bedient werden. Weka bietet eine grafische Oberfläche zur Bedienung und kann die berechneten Ergebnisse sowohl textuell als auch grafisch darstellen [33].

7) Online-/Offline-Analyse: Mit netAI ist sowohl eine Online-Analyse von Live-Daten als auch eine Offline-Analyse von zuvor gespeicherten Datensätzen möglich [33].

C. BLINC

BLINC, die Kurzform von „BLINd Classification“, ist ein in [31] vorgestelltes verhaltensbasiertes Tool zur Klassifizierung von Netzwerkverkehr. Untersucht wird das Verhalten von Hosts auf drei verschiedenen Ebenen, um später aus der Kombination des Verhaltens auf allen drei Ebenen auf die den Traffic generierende Anwendung zu schließen. Die erste Ebene ist die soziale Ebene, auf der die Popularität eines Hosts und sogenannte „Communities“von mehreren Hosts identifiziert werden, um Clients mit ähnlichen Interessen und kollaborative Applikationen aufzudecken. Die zweite Ebene ist die funktionale Ebene, auf der die Rolle eines Hosts im Netzwerk identifiziert werden soll. So kann ein Host ein Client, ein Server

oder im Falle von kollaborativen Anwendungen auch beides sein. Auf der dritten, der Applikations-Ebene, werden die Interaktionen auf der Transportschicht eines Hosts untersucht. Hier spielen das genutzte Transportprotokoll, die genutzten Quell- und Ziel-Adressen und -Ports, sowie andere Flow-Charakteristika, wie z.B. die durchschnittliche Paketgröße, eine Rolle. Da es sich hier um einen supervised Machine-Learning-Ansatz handelt, werden Testdaten benötigt, mit denen sogenannte „Graphlets“als Muster für jede Applikation gebildet werden. Es werden außerdem Heuristiken gebildet, um sehr ähnliche Graphlets zu unterscheiden. BLINC zeichnet sich dadurch aus, dass viele Parameter für den aktuell zu klassifizierenden Verkehr genau konfiguriert werden können, um so das bestmögliche Ergebnis zu erzielen [31].

1) Def. Verkehrsklassen: Durch die Verwendung der Graphlets ist BLINC so konzipiert, dass Muster für neue zu klassifizierende Applikationen definiert werden können. Aktuell stehen Graphlets für Applikationen unterschiedlicher Kategorien zur Verfügung, z.B. P2P-Anwendungen (BitTorrent, FastTrack), FTP-Anwendungen oder Chat-Anwendungen (IRC, MSN Messenger, AIM). Die Enwicklung neuer Graphlets ist allerdings eine nicht-triviale Aufgabe bei der insbesondere auf Überschneidungen von Informationen zwischen existierenden und neuen Graphlets geachtet werden muss. Ähneln sich die Graphlets zu sehr, so müssen zusätzliche Features definiert werden, um eine genaue Unterscheidung zu ermöglichen [31].

2) Input/Automatisierung: Vor jeder Analyse sollten 28 zur Verfügung stehende Klassifizierungsparameter abhängig von dem zu klassifizierenden Verkehr angepasst werden. Input-Möglichkeiten sind Daten im tcpdump-Format zur Offline- und Daten von Netzwerk-Schnittstellen zur Online-Klassifizierung [31].

3) Def. Granularität: Die Granularität von BLINC kann über 28 Klassifizierungsparameter vor der Analyse konfiguriert werden. Die Anpassung der Parameter sollte für jede Analyse durchgeführt werden, um eine möglichst hohe Genauigkeit zu erzielen [29]. Die Parameter sind in der offiziellen Veröffentlichung nicht näher erläutert und müssen in einer weiterführenden Arbeit recherchiert werden [31].

4) Effizienz: In [29] wurde BLINC unter anderem auf die Performanz untersucht. Auf einem Rechner mit zwei 2.4 GHz Zeon Prozessoren und 4 GB Arbeitsspeicher konnte Netzwerkverkehr mit einem Speicherverbrauch von weniger als 2 GB mit durchschnittlich rund 1 Million Flows pro Minute in Echtzeit analysiert werden. In [28] wurde ein Benchmark-Tool, das unter anderem auch BLINC implementiert, vorgestellt. Getestet wurde BLINC auf einem Rechner mit einem 2.4 GHz Intel Core 2 Quad Kentzfield Q6600 Prozessor mit 4 GB Arbeitsspeicher. Hier wurde der mehrere Gigabyte große Netzwerkverkehr eines Universitäts-Campus unter ebenfalls ca. 2 GB Speicherverbrauch analysiert [28]. Die Entwickler von BLINC geben in [31] an, dass das Tool einen 34 stündigen Netzwerkverkehrsmitschnitt innerhalb von 8 Stunden auf einem DELL PE2850 (Xeon 3.4 GHz Prozessor, 2 GB Arbeitsspeicher) klassifizieren konnte.

5) Genauigkeit: Nach [29] klassifiziert BLINC WWW-, DNS-, Mail-, Chat-, FTP- und Streaming-Verkehr mit einer Genauigkeit von über 90%. P2P-Verkehr wird mit einer Genauigkeit von über 85% klassifiziert. Zu beachten ist dabei, dass BLINC Flows einer bestimmten Anwendung erst dann klassifiziert, wenn ein Mindestvorkommen im gesamten Verkehr vorhanden ist. Der Schwellenwert hierfür wird nicht genannt. Falls eine Anwendung nicht genügend Flows erzeugt, wird der Verkehr als „nicht-klassifizierbar“deklariert [29]. In [31] geben die Entwickler von BLINC an, dass ihr Tool 90% aller Flows mit einer Genauigkeit von 95% klassifiziert. Die Art des Verkehrs, der zu diesen Aussagen führte wird nicht weiter benannt [31].

6) Output/Bedienung: BLINC bietet als Output statistische Informationen (z.B Anzahl verschickter Pakete, Flows oder Bytes) zu jeder klassifizierten Klasse und eine Liste aller Flows, die zu einer identifizierten Anwendung gehören [31]. Eine öffentlich zugängliche Version von BLINC steht nicht zur Verfügung, sodass die vorhandenen Bedienmöglichkeiten in einer weiterführenden Arbeit untersucht werden müssen.

7) Online-/Offline-Analyse: BLINC bietet die Möglichkeit, Online- und Offline-Analysen durchzuführen [31] [29].

VIII. Zusammenfassung und Ausblick

In dieser Arbeit wurde zuerst ein Überblick über das Problem der Klassifizierung von Netzwerkverkehr erstellt. Dazu wurden zunächst in Kapitel II verschiedene, aktuelle Methoden zur Klassifizierung vorgestellt und einige Vor- und Nachteile gezeigt, bevor dann in Kapitel III aktuelle Problematiken der Klassifizierung genannt und Lösungsansätze aufgezeigt wurden. Als nächstes wurden in Kapitel IV die drei Verkehrsklassen rt-VPL, rt-CPL und nrt-UPL von den ATM-Service-Klassen abgeleitet, mit deren Hilfe ein Netzwerk hinsichtlich des Quality of Service optimiert werden könnte. Anschließend wurden diese Verkehrsklassen in Kapitel V mit statistischen Features modelliert. Im letzten Teil dieser Arbeit wurden dann in Kapitel VII drei verschiedene Klassifizierungs-Tools hinsichtlich von in Kapitel VI aufgestellten Anforderungen untersucht und bewertet.

Von den in dieser Arbeit untersuchten Klassifizierungs-Tools scheint im Hinblick auf die in Kapitel V vorgestellte Modellierung das Tool netAI am besten geeignet zu sein. Es handelt sich dabei um ein Tool mit statistischem Ansatz, so dass die zur Modellierung genutzten Features nativ genutzt werden können. Auch die anderen Anforderungen erfüllt netAI, wobei die Effizienz, die ein wichtiger Faktor für den Nutzen in der Praxis eines solchen Tools ist, in dieser Arbeit nicht bewertbar war und so in einer weiteren Arbeit genauer untersucht werden müsste. Am zweitbesten schneidet BLINC in der Bewertung ab. Die wenigsten Anforderungen erfüllt nDPI, was hauptsächlich der Tatsache geschuldet ist, dass es sich hier um eine Bibliothek und keine All-in-One-Lösung für die Klassifizierung handelt. Außerdem arbeitet nDPI auf Applikationsebene, wobei als Workaround für die Klassifizierung nach den in Kapitel IV vorgestellten Verkehrsklassen eine Zuweisung der Applikationen in eine der drei Klassen funktionieren könnte.

Neben der Untersuchung der Effizienz von netAI wäre für die Zukunft eine Analyse der hier definierten Verkehrsklassen wünschenswert, insbesondere die Frage, ob die Teilung der Klasse des Echtzeitverkehrs in VPL und CPL für die Optimierung eines Netzwerks hinsichtlich des Quality of Service sinnvoll ist oder nicht. Ein weiterer interessanter Bereich

für eine tiefgründigere wissenschaftliche Untersuchung sind die hybriden Ansätze, die mehrere Klassifizierungs-Methoden miteinander verbinden. Diese scheinen hinsichtlich der Genauigkeit gute Ergebnisse zu erzielen, wobei die Effizienz für die Praxis aber auch sehr wichtig ist. Insgesamt kann anhand der Menge an aktueller Literatur zu dem Thema gesagt werden, dass die Klassifizierung von Netzwerkverkehr ein vielfältiger und aus wissenschaftlicher Sicht interessanter Bereich ist.

LITERATUR

[1] S. Valenti, D. Rossi, A. Dainotti, A. Pescapè, A. Finamore, and M. Mellia, "Reviewing traffic classification," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7754, pp. 123–147, 2013.

[2] S. Ehlert, S. Petgang, T. Magedanz, and D. Sisalem, "Analysis and signature of skype voip session traffic," *4th IASTED International*, 2006.

[3] M. Finsterbusch, C. Richter, E. Rocha, J. A. Müller, and K. Hän\ss gen, "A survey of payload-based traffic classification approaches," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 2, pp. 1135–1156, 2014.

[4] A. Finamore, M. Mellia, M. Meo, and D. Rossi, "Kiss: Stochastic packet inspection classifier for udp traffic," *Networking, IEEE/ACM Transactions on*, vol. 18, no. 5, pp. 1505–1515, 2010.

[5] A. R. Khakpour and A. X. Liu, "High-speed flow nature identification," in *2009 29th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 510–517.

[6] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *Commun. Surv. Tutorials, IEEE*, vol. 10, no. 4, pp. 56–76, 2008.

[7] E. P. Freire, A. Ziviani, and R. M. Salles, "Detecting skype flows in web traffic," in *NOMS 2008 - 2008 IEEE Network Operations and Management Symposium*, pp. 89–96.

[8] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," *Passiv. Act. Netw. Meas.*, vol. 3431, pp. 41–54, 2005. [Online]. Available: http://www.springerlink.com/index/re7ej0uj7eep2htl.pdf

[9] A. Dainotti, A. Pescape, and K. Claffy, "Issues and future directions in traffic classification," *IEEE Network*, vol. 26, no. 1, pp. 35–40, 2012.

[10] T. Yildirim and P. Radcliffe, "Voip traffic classification in ipsec tunnels," in *Electronics and Information Engineering (ICEIE), 2010 International Conference On*, vol. 1, 2010, pp. V1–151–V1–157.

[11] H. A. H. Ibrahim, S. M. Nor, and H. A. Jamil, "Online hybrid internet traffic classification algorithm based on signature statistical and port methods to identify internet applications," in *2013 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pp. 185–190.

[12] The ATM Forum, "Traffic management specification," 1999.

[13] T. Mickelsson, "Atm versus ethernet," Helsinki, 1999. [Online]. Available: http://www.tml.tkk.fi/Opinnot/Tik-110.551/1999/papers/07ATMvsEthernet/iworkpaper.html

[14] J.-L. Costeux, F. Guyard, and A.-M. Bustos, "Qrp08-5: Detection and comparison of rtp and skype traffic and performance," in *IEEE Globecom 2006*, pp. 1–5.

[15] R. Sparks, "Sip: basics and beyond," *Queue*, vol. 5, no. 2, p. 22, 2007.

[16] W.-c. Feng and K. Mayer-Patel, Eds., *the international workshop*.

[17] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "Rtp: A transport protocol for real-time applications," 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

[18] Dongyan Chen, Daqing Gu, and Jinyun Zbang, "Supporting real-time traffic with qos in ieee 802.11e based home networks," in *2004 1st IEEE Consumer Communications and Networking Conference*, 5-8 Jan. 2004, pp. 205–209.

[19] P. Ameigeiras, J. J. Ramos-Munoz, J. Navarro-Ortiz, and J. M. Lopez-Soler, "Analysis and modelling of youtube traffic," *Transactions on Emerging Telecommunications Technologies*, vol. 23, no. 4, pp. 360–377, 2012.

[20] G. Lu, H. Zhang, M. Qassrawi, and X. Yu, "Comparison and analysis of flow features at the packet level for traffic classification," in *2012 International Conference on Connected Vehicles and Expo (ICCVE)*, pp. 262–267.

[21] F. Dehghani, N. Movahhedinia, M. R. Khayyambashi, and S. Kianian, "Real-time traffic classification based on statistical and payload content features," in *2010 2nd International Workshop on Intelligent Systems and Applications (ISA)*, pp. 1–4.

[22] I. Tsompanidis, A. H. Zahran, and C. J. Sreenan, "Mobile network traffic: A user behaviour model," in *2014 7th IFIP Wireless and Mobile Networking Conference (WMNC)*, pp. 1–8.

[23] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing skype traffic," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, p. 37, 2007.

[24] K.-T. Chen, P. Huang, C.-Y. Huang, and C.-L. Lei, "Game traffic analysis," in *the international workshop*, W.-c. Feng and K. Mayer-Patel, Eds., p. 19.

[25] R. Raghavendra and E. M. Belding, "Characterizing high-bandwidth real-time video traffic in residential broadband networks," in *WiOpt*, pp. 597–602. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5520342

[26] P. Teufl, U. Payer, M. Amling, M. Godec, S. Ruff, G. Scheikl, and G. Walzl, "Infect - network traffic classification," in *2008 Seventh International Conference on Networking ICN*, pp. 439–444.

[27] H. Kim, K. Claffy, M. Fomenkova, N. Browlee, D. Barman, and M. Faloutsos, "Comparison of internet traffic classification tools," *IMRG WACI*, vol. 2007, 2007. [Online]. Available: http://www.icir.org/imrg/waci07/docs/waci-3-abs.pdf

[28] S. Lee, H. Kim, D. Barman, S. Lee, C.-k. Kim, T. Kwon, and Y. Choi, "Netramark," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, p. 22, 2011.

[29] H. Kim, K. C. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, "Internet traffic classification demystified," in *the 2008 ACM CoNEXT Conference*, A. Azcorra, G. d. Veciana, K. W. Ross, and L. Tassiulas, Eds., pp. 1–12.

[30] Y. Jin, N. Duffield, J. Erman, P. Haffner, S. Sen, and Z.-L. Zhang, "A modular machine learning system for flow-level traffic classification in large networks," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 1, pp. 1–34, 2012.

[31] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "Blinc: multilevel traffic classification in the dark," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 229–240, 2005. [Online]. Available: http://portal.acm.org/citation.cfm?id=1090191.1080119

[32] ntop, "ntopng user's guide," 2014. [Online]. Available: https://github.com/ntop/ntopng/raw/dev/doc/UserGuide.pdf

[33] Centre for Advanced Internet Architectures, "netai - introduction," 2011. [Online]. Available: http://caia.swin.edu.au/urp/dstc/netai/netai-intro.html

[34] T. Bujlow, V. Carela-Español, and P. Barlet-Ros, "Independent comparison of popular dpi tools for traffic classification," *Computer Networks*, vol. 76, no. 0, pp. 75–89, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128614003909

[35] L. Deri, M. Martinelli, T. Bujlow, and A. Cardigliano, "ndpi: Open-source high-speed deep packet inspection," in *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 617–622.

[36] ntop, "ndpi - quick start guide," 2014. [Online]. Available: https://github.com/ntop/nDPI/raw/dev/doc/nDPI_QuickStartGuide.pdf

[37] D. Arndt, "netmate-flowcalc feature summary," 2011. [Online]. Available: https://code.google.com/p/netmate-flowcalc/wiki/Features

[38] N. Williams, "netai - network traffic based application identifier," Australia, 2006. [Online]. Available: http://caia.swin.edu.au/reports/060410E/CAIA-TR-060410E.pdf

[39] Centre for Advanced Internet Architectures, "netai - example results," 2011. [Online]. Available: http://caia.swin.edu.au/urp/dstc/netai/netai-examples.html