Ausarbeitung

Mobilität und Dienste

CN4 - WS99/00

ausgearbeitet von:

Alexander Schneider
schnei@alpha.fh-furtwangen.de

Marc Löffler
loeffler@foo.fh-furtwangen.de

Andreas Schlenk
schlenk@foo.fh-furtwangen.de

Inhaltsverzeichnis

1 Einleitung

1.1 Auswirkungen von Mobilität auf das Protokolldesign

Die Mobilität stellt eine besondere Anforderung an das Protokolldesign. Dabei ist darauf zu achten, daß die Datenübertragungen bei mobilen Diensten meist gering ist. Deshalb muß darauf geachtet werden, daß man so wenig wie möglich Overhead hat. Manche mobile Dienste bedienen sich deshalb dem Binär-Code. Auch ist darauf zu achten, daß man bei mobilen Geräten meist keinen großen Monitor hat und schlechte Eingabemöglichkeiten, wie bei einem Handy und Palm-Top.

Bei unserem Vortrag und unserer Ausarbeitung haben wir 3 aktuelle Beispiele für Mobile Dienste herausgenommen, um die verschiedenen Möglichkeiten der Datenübertragung und Darstellung der Daten aufzuzeigen.

1.1.1 Auswirkung auf Adressierungsverfahren

Beim Adressierungsverfahren bieten sich die verschiedensten Möglichkeiten. Während man bei Bluetooth und IrDA eine nur begrenzte Teilnehmerzahl hat (aufgrund kleiner Reichweiten), sind bei WAP nahezu unbegrenzte Teilnehmerzahlen möglich. Entsprechend sieht dann die Adressierung aus. Bluetooth beispielsweise bietet für lediglich 8 Teilnehmer die Möglichkeit zur Teilnahme an der Datenkommunikation - also benötigt man eine Adresscodierung von 3 Bit -, während bei WAP eine große Anzahl an Kommunikationspartnern möglich ist und deshalb die Adresse wie beim herkömmlichen Internet (IP) sehr viel länger ist.

1.1.2 Auswirkung auf Übertragungsverfahren

Die Übertragungsverfahren müssen optimiert sein, da wegen der geringen Bandbreite kein Overhead gebraucht werden kann. Mögliche verfahren wären wir bei der Satellitenverbindung das ARQ-Protokoll, bei dem nicht jedes Paket einzeln bestätigt wird, sondern ein bestimmtes Paket, welches die vorigen Pakete mit bestätigt. Dabei ist das ARQ-Protokoll das Pendant zum HDLC, jedoch können 127 Pakete übertragen werden, bevor eine Empfangsbestätigung erfolgen muß.

1.1.3 Auswirkung auf Fehlerkorrekturverfahren

Fehler sind möglichst zu vermeiden. Wegen der geringen Bandbreite wäre es schlecht, wenn viele Fehler auftreten würden und dann viele Pakete ein zweites oder drittes mal geschickt werden müssen. Deshalb wäre zum Beispiel ein Verfahren zu empfehlen, bei dem zusätzliche Informationen geschickt werden, mit denen im Fehlerfall die defekten Pakete repariert werden können. Solche verfahren nennt man FEC-Verfahren (Forward Error Correction). Diese benötigen jedoch zusätzliche Bandbreite, weshalb ein geeigneter Kompromiß gefunden werden muß.

2 Bluetooth

2.1 Motivation

Bluetooth wurde 1998 den von Firmen Ericsson. Nokia, Toshiba, IBM und Intel ins Leben gerufen, wobei jede der beteiligten Firmen ihr Know-How in ihrem jeweiligen Bereich einbrachte. Dadurch waren weitreichende Kenntnisse aus den Bereichen Funkübertragung, Mobile Computing und Chipfertigung vorhanden.

Die Motivation der fünf Firmen, die sich in der „Bluetooth Special Interest Group“ (SIG) zusammenschlossen, war die verwirrende Vielzahl von Peripherieschnittstellen durch eine einzige zu ersetzen.

Das Ziel war eine universelle, kostengünstige, für Enduser leicht zu bedienende Schnittstellentechnologie zu entwickeln, welche zusätzlich noch kleine, drahtlose Netze im Home-, Homeoffice und Smalloffice ermöglicht. Durch eine weitestgehend vollautomatische Konfiguration der BT-Netze, welches sowohl Sprach- als auch Datendienste anbieten kann, soll die Mobilität der User in der Hinsicht unterstützt werden, dass diese ihre mobilen Endgeräte (Laptop, PDA, Handy) ohne Probleme untereinander vernetzen und somit Daten austauschen können.

2.2 Netzstrukturen

Das BT-Netz basiert auf dem Master/Slave-Prinzip. Wobei sowohl Punkt-zu-Punkt- als auch Punkt-zu-Multipunkt-Verbindungen möglich sind. Obendrein finden sich aber auch neuartige Eigenschaften, die dezentral und selbstorganisiert sind. Zwei oder mehr BT-Gegenstellen können beispielsweise bei gegebener Freischaltung denselben Kanal teilen und ad hoc ein so genanntes Piconetz bilden - das sind räumlich gesehen sehr kleine Netze. Eine BT-Einheit agiert dabei als Master, die übrigen - bis zu sieben je Piconetz - als aktive Slaves. Jedes Piconetz verwendet ein anderes Frequenz-Hopping-Muster (siehe 2.4.2.1 Kanalaufteilung - Seite 1), das der Master vorgibt und so den Kanalzugriff regelt. Neben dem Master und den aktiven Slaves können an einem Piconetz viele geparkte Slaves passiv teilnehmen. Ebenfalls synchron mit dem Master, wechseln sie die Frequenzen und Slots, doch ist ihnen der Kanalzugriff nicht gestattet; sie horchen lediglich.

Mehrere Piconetze können nebeneinander im selben oder in überlappenden Versorgungsbereichen koexistieren, miteinander ein Scatter-Netz bilden - die Mitglieder der verschiedenen Pico-Netze stehen indirekt mit den übrigen Teilnehmern dieses Verbunds über die jeweiligen Master in Kontakt. Jedes Piconetz darf nur einen Master haben (nur einer kann den Takt des Frequenz-Hoppings vorgeben), doch Slaves können auf Basis eines Zeitmultiplexschemas an verschiedenen Piconetzen partizipieren. Ein Master in einem Piconetz kann in einem anderen Piconetz Slave sein.

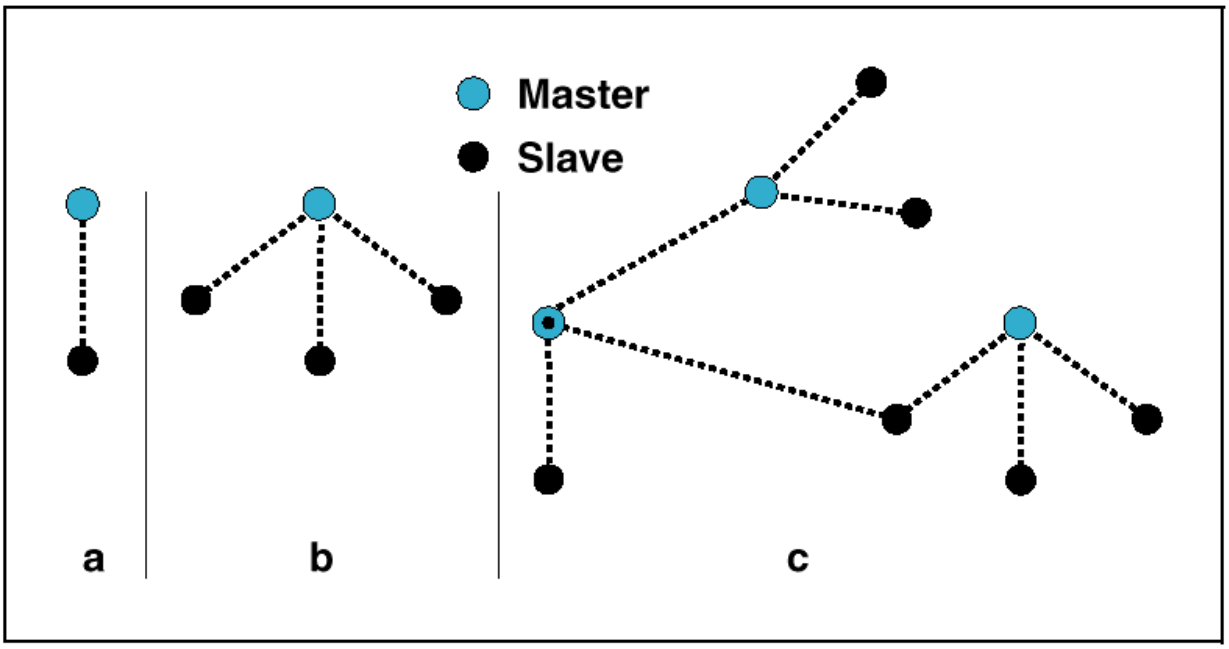


Figure 1.2: Piconets with a single slave operation (a), a multi-slave operation (b) and a scatternet operation (c).

Abbildung 2-1 / Netzstrukturen

2.3 Protokollstack

Um den Anforderungen in den in 2.1 Motivation (Seite 1) geschilderten Einsatzgebieten gerecht zu werden, wurde der Aufbau und die Funktionen der einzelnen Protokollschichten einfache und „schlank“ gehalten.

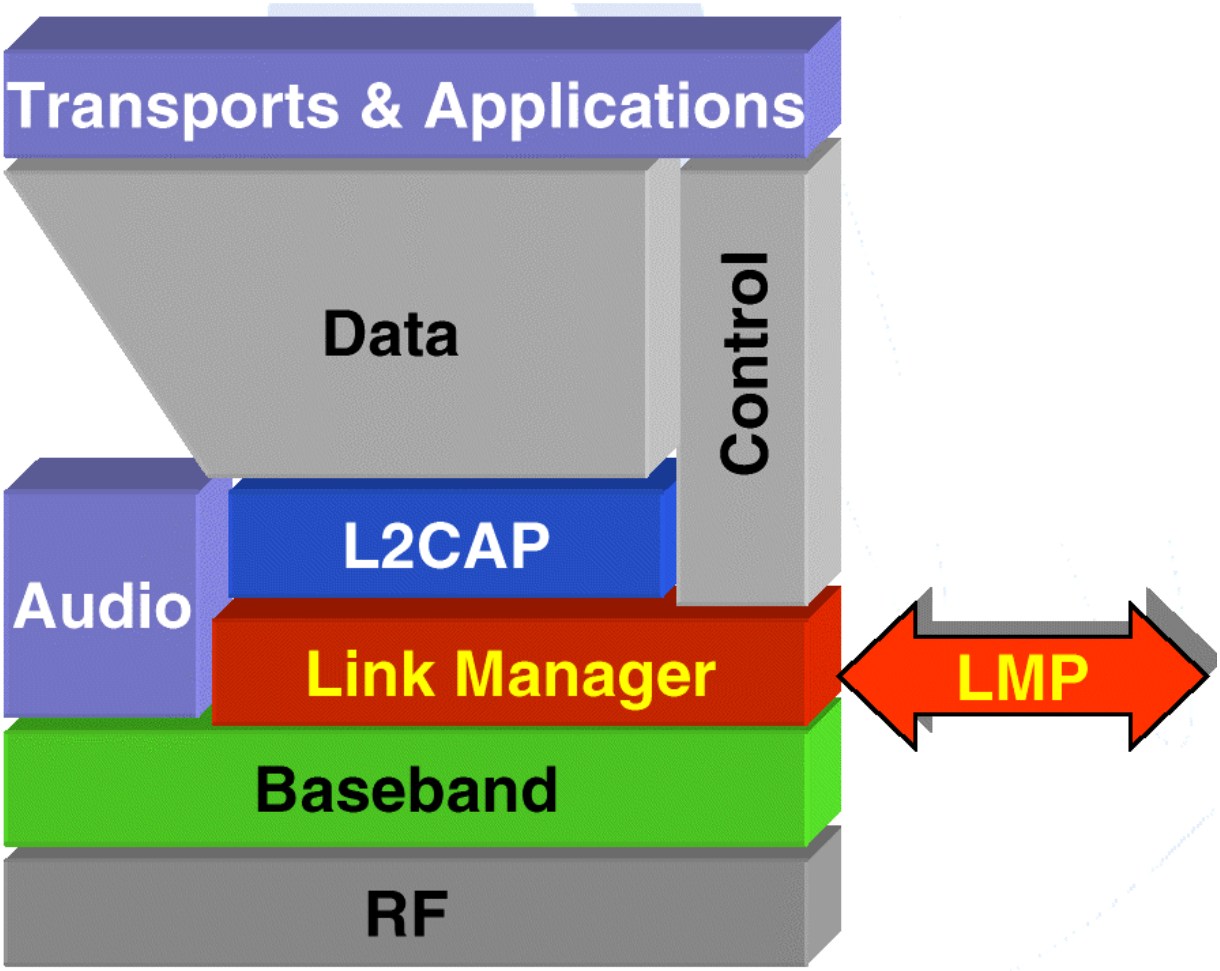


Abbildung 2-2 / Protokollstack

2.4 Funktionsblöcke

Eine BT-Einheit von einem Endgerät besteht im wesentlichen aus drei Funktionsblöcken (siehe hierzu auch Abbildung 2-3 / Funktionsblöcke).

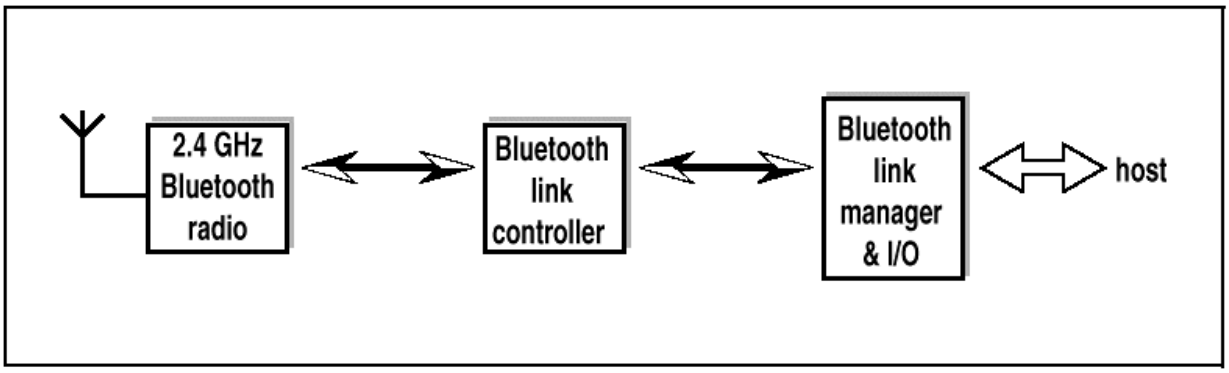


Figure 1.1: Different functional blocks in the Bluetooth system

Abbildung 2-3 / Funktionsblöcke

2.4.1 Radio-Unit

Die Radio-Unit realisiert die eigentlichen mittels Transceiver. Die Übertragung findet im lizenzfreien ISM-Band (2,4 GHz) statt.

2.4.2 Link-Controller-Unit

In der Link-Controller-Unit sind die Baseband-Protokolle definiert sowie einige Low-Level-Link-Routinen. D.h. hier wird der Medienzugriff, Signalmodulation und Übertragungungsmodus gesteuert.

2.4.2.1 Kanalaufteilung

Oberhalb von 2042 MHz sind 79 Kanäle (bzw. 23 in diversen Ländern) mit einem Frequenzband von je 2 MHz definiert, welche mittels GFSK-Modulation (Gaussian-Frequency-Shift-Keying) verwendet werden. Eine 1 wird als eine positive Frequenz-Deviation, eine 0 als negative Frequenz-Deviation übertragen. Typisch für Frequenzmodulationen kommt GFSK mit einem effizienten Verstärker aus, was den Stromverbrauch und die Hitzeentwicklung senkt. Auch ist die Demodulation einfach.

Als Übertragungsmodus wird das Time-Division-Duplex-Verfahren (TDD) eingesetzt, wobei Timeslices von 625µs definiert sind und dadurch 1600 Frequenzhops/s realisiert werden. Jedes Mal wenn ein Paket gesendet oder empfangen wurde, wechseln die beteiligten Funkmodule die Kanalfrequenz um Interferenzen mit anderen Funksignalen zu vermeiden. Verglichen mit anderen Systemen, die im ISM-Band funken (etwa Drahtlosnetze gemäß IEEE 802.11, aber auch Mikrowellenöfen), „hüpft“ BT schneller, und die Pakete sind kleiner. Letzteres erhöht zwar den Anteil der Verwaltung gegenüber den Nettodaten, doch ist das bei stark gestörter Funkumgebung, in der zerstörte Pakete erneut gesendet werden müssen, ökonomischer - je kleiner die Pakete, desto höher die Wahrscheinlichkeit, dass sie gerade in dem Moment gesendet werden, wenn keine Funkstörungen auftreten.

Die Reichweite von normalen BT-Systemen liegt bei ca. 10m, wobei ein Mindestabstand zwischen zwei Geräten von 10cm zu beachten ist. Die Funkleistung von normalen Systemen liegt bei ca. 1mW. Mit einem sog. Giga-Bluetooth-System können auch Reichweiten von bis zu 100m erreicht werden.

2.4.2.2 Fehlerkorrektur

Neben dem oben genannten TDD-Mode, womit versucht wird, Fehler erst gar nicht auftreten zu lassen, kommen als Fehlererkennung- bzw. -korrekturverfahren die so genannte Vorwärtsfehlerkorrektur (Forward Error Correction, FEC), sowie das Automatic Repeat Request-Verfahren (ARQ) und CRC (Cyclic Redundancy Check) zum Einsatz. Da die FEC einigen Raum im Paket beansprucht sowie den Verwaltungsaufwand erhöht und dadurch den Durchsatz senkt, dient sie dem Schutz der Nutzlast des Pakets nur dann, wenn der Funkraum stark gestört ist.

2.4.2.3 Paketaufbau

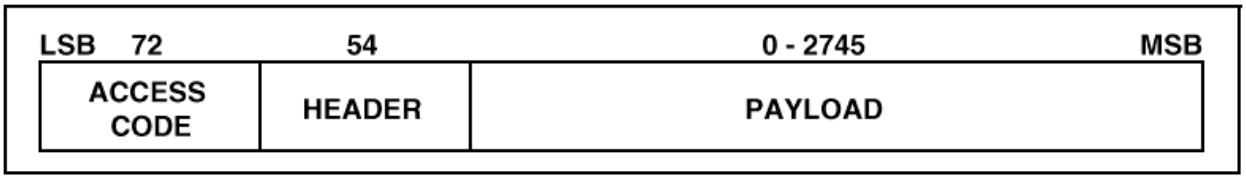


Figure 4.1: Standard packet format.

Abbildung 2-4 / Paketaufbau

Access-Code	zur Synchronisation
	DC-Offset-Kompensation
	Idendifikation ⇨ Access-Types (Channel-Acces / Device-Access / Inquiry-Acces)
Header	wird für Managed Link-Control verwendet
	FHS-Pakete für Hop-Syncronisation
	für Reorganisation im Netz
Payload	0 - 2745Bits (enthält je nach Pakettyp weitere Control-Informationen)

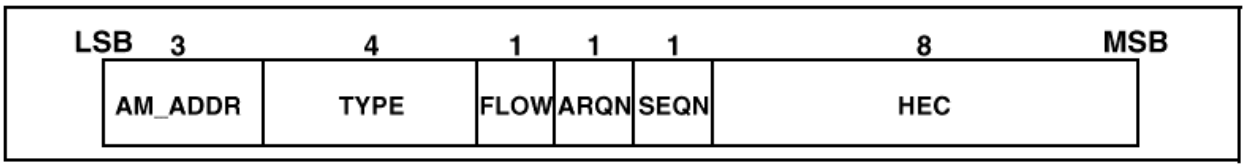


Figure 4.5: Header format.

Abbildung 2-5 / Headeraufbau

AM_ADDR	3-bit active member address	ARQN	1-bit acknowledge indication
TYPE	4-bit type control	SEQN	1-bit sequence number
FLOW	1-bit flow control	HEC	8-bit header error chech

2.4.3 Link-Manager- und I/O-Unit

Neben der Link-Controller-Unit (siehe 2.4.2 Link-Controller-Unit - Seite 1) stellt der Link-Manager einen zweiten wichtigen Funktionsblock in BT dar. Hier geschieht das Netzwerk- und Link-Management.

2.4.3.1 Link-Management-Protocol (LMP)

Im LMP werden das Management für das Piconetz, die Link-Konfiguration und die Security-Funktionen realisiert.

Übersicht der wichtigsten Funktionen des LMP

Piconetz-Management	Link-Konfiguration	Security-Funktionen
ein- und austragen von Slaves	Supported Features	Authentifikation
Master/Slave-Umschaltung	Quality of Service (Qos)	Verschlüsselung
ACL- und SCO-Links herstellen	useable packet types	Schlüsselmanagement
Behandlung der Energiespar-Moden (Hold, Sniff, Park)	Power Control	

2.5 Logical Link-Control and Adaption-Protocol (L2CAP)

Das L2CAP dient zur Adaption der Protokolle der höheren Layer.

Protokolle die auf BT aufsetzten sind zur Zeit:

- IrDA per IrOBEX (Object Exchange Protocol)
- TCP/IP
- RFCOMM (einfaches serielles Interface, vergleichbar mit V.24)
- SDP (Service Discovery Protocol)
- TSC (Telephony control Protocol)

Des weiteren übernimmt das L2CAP das SAR-Management (Segmentation und Reassembling) sowie das Group-Management wenn mehrere Teilnehmer in verschiedenen Scatternetzen logische Gruppen bilden wollen, welche nicht der Scatternetz-Struktur entsprechen.

2.6 Verbindungstypen

BT stellt sowohl verbindungslose, sogenannte „Asymetric Connection Less“-Links (ACL), als auch „Syncronuous Connection Oriented“-Links zur Verfügung. Wobei der SCO-Links für Point-toPoint-Verbindungen und der ACL-Link für Point-to-MultiPoint-Verbindungen gedacht ist.

2.7 Bandbreiten

Die Aufteilung der zur Verfügung stehenden Bandbreiten kann variabel gestaltet werden. Grundsätzlich werden asynchrone und synchrone Übertragungen unterschieden.

Bei der synchronen Übertragung, welche vornehmlich für Sprachdaten konzipiert ist (PCM oder CVSD-Modulation), stehen bis zu 3x 64kb/s zur Verfügung.

Asynchrone Verbindungen, welche für Datenübertragung ausgelegt sind, gibt zum einen eine asymetrische Aufteilung der Bandbreite in 723,2kb/s für den Downlink und 57,6kb/s für den Uplink, sowie eine symetrische Aufteilung in jeweils 433,9 kb/s.

3 IrDA

3.1 Einleitung

IrDA (Infrared Data Association) bildet den Standard für Datenübetragungen mit Licht im infraroten Bereich. IrDA-Schnittstellen gibt es an fast jedem Laptop oder neueren

Mobiltelefon, trotzdem fristet IrDA eher ein Schattendasein im Bereich der Datenübetragung ! Unter anderem liegt das daran, daß sich IrDA nicht gerade für jedes Betriebssystem eignet. Die einzigen Betriebssysteme auf denen IrDA implementiert ist, sind Windows 95/98, MacOS und OS/2 ! In Linux ist der Protocol-Stack von IrDA immer noch als 'Experimental Feature' im Kernel zu finden und bei Windows NT ist eine Zusatzsoftware notwendig um IrDA zu nutzen.

Jede IrDA-Verbindung ist bidirektional, funktioniert aber nach dem Master-Slave-Prinzip. Bei drei Geräten tauschen immer zwei davon als Slave mit dem Master Daten aus; die Slave-Geräte wissen nichts voneinander. Welcher von beiden Rechnern die Master-Rolle übernimmt, bleibt dem Zufall überlassen: Beide können alle paar Sekunden `Schnüffel-Signale´ aussenden. Wer zuerst antwortet, begnügt sich mit der Slave-Rolle.

Wie bei allen kabellosen Verbindungen sollte man sich darüber im Klaren sein, wie abhörsicher die Daten vom Sender zum Empfänger gelangen. Da IrDA eine gerichtete Lichtaussendung für geringe Entfernungen definiert, hören andere IrDA-Empfänger nicht `zufällig´ mit. Zudem tauschen die beiden beteiligten Transceiver ständig Daten aus. Zumindest unter Windows 95 und 98 sieht der Nutzer, zu welchem Port er die Daten schickt.

Das IrDA-Protokoll enthält keine Verschlüsselung der übertragenen Daten. Hoch empfindliche Infrarotempfänger sind aber in der Lage, Streulicht und damit Daten aufzunehmen und ohne große Mühe zu lesen. Befindet sich in dem 30° breiten Lichtkegel eine reflektierende Oberfläche, die womöglich noch als Konvexlinse wirkt, dann können Spione Infrarotdaten unkontrolliert und unbemerkt auch über größere Entfernungen abhören. Bei hohen Sicherheitsanforderungen sollte man die Daten daher vor der Übertragung verschlüsseln.

3.2 Historie

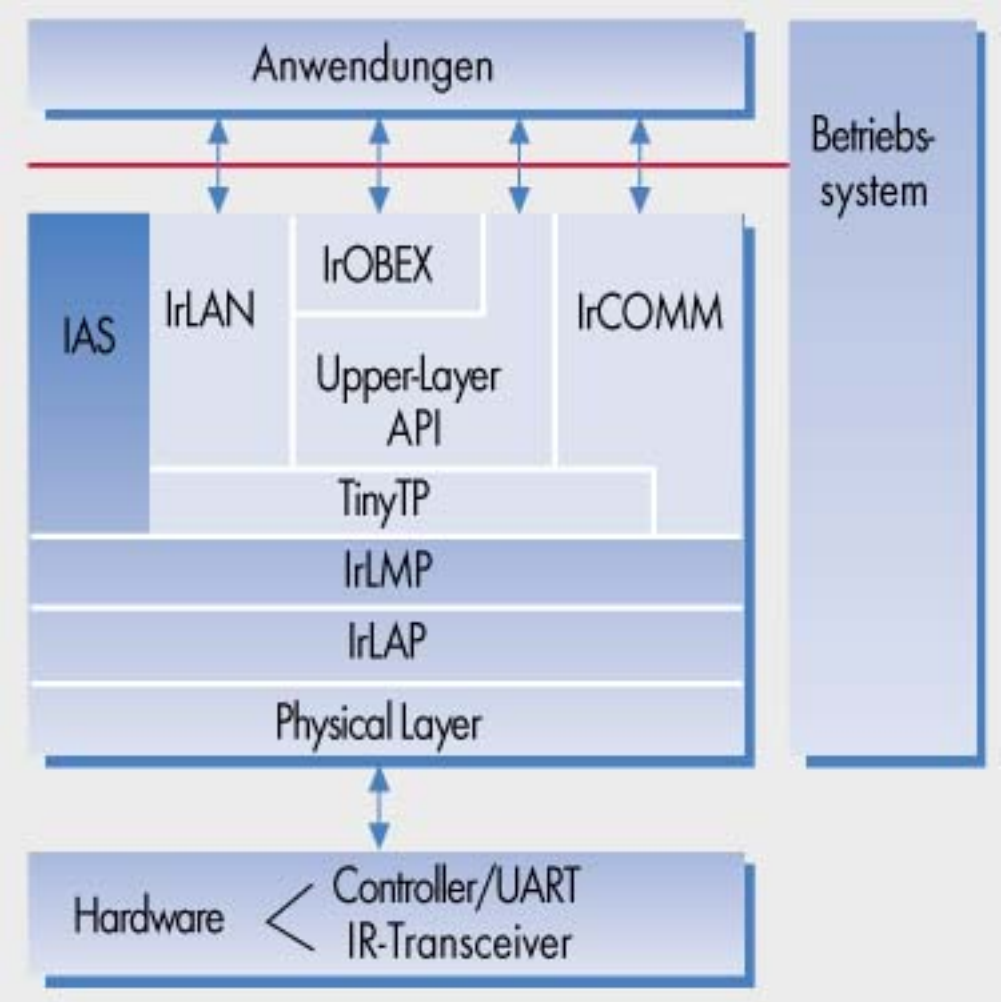
Nachdem sich 1993 über 30 Firmen, darunter Hewlett-Packard, Digital und IBM, zur Infrared Data Association (IrDA) zusammenschlossen, dauerte es noch bis Juni 1994, bevor sie sich auf den Standard IrDA 1.0 einigten, auch SIR (Standard IR) genannt. Der Standard folgte in Teilen den Vorarbeiten von Hewlett-Packard. Die maximale Datenrate entspricht dabei der UART-Rate von 115,2 KBit/s, wie man sie von der seriellen Schnittstelle her kennt. Der Vorteil liegt auf Hardware-Seite darin, dass ein UART-Chip sowohl den seriellen Port als auch den Infrarot-Transceiver (Transmitter und Receiver) steuern kann.

Für LAN-Anwendungen oder größere Datenmengen reicht das nicht aus. Im Oktober 1995 folgte deshalb mit IrDA 1.1 die Erweiterung Fast IrDA oder FIR, mit der die Datenrate bis zu 4 MBit/s ansteigt. In Zukunft soll infrarotes Licht Daten noch schneller übertragen, nachdem im Januar 1999 IrDA die Erweiterung Very Fast IR (VFIR) spezifiziert hat. Die maximale Datenrate beträgt damit 16 MBit/s. Noch ist aber keine Hardware verfügbar.

3.3 Protocol Stack

Die Infrared Data Association (www.irda.org) unterteilt den IrDA-Standard in IrDA DATA für die Kommunikation von Geräten und IrDA CONTROL für die Anbindung von Rechnerperipherie wie Infrarot-Maus und -Tastatur.

Der Protocol – Stack setzt sich, wie sein großes Vorbild das ISO-OSI – Modell, aus verschiedenen Schichten (Layern) zusammen, von denen je nach Einsatz unterschiedlich viele zum Einsatz kommen !



Für die minimale Implementierung des Protocol-Stack verlangt IrDA folgende Schichten :

Physical Layer : Diese Schicht beschreibt die Spezifikation des Infrarotlichts und die legt die Lichtmodulation für die Datenübertragung fest.

IrLAP : Das Link Access Protocol sorgt für den automatischen Verbindungsaufbau zweier Geräte für deren verlässliche Verbindung durch die implementierte Fehlerkorrektur.

IrLMP : Das Link Management Protocol dient dazu mehrere Geräte als Slave mit einem Host als Master kommunizieren zu lassen.

IAS : Die `gelben Seiten´ der Infrarotverbindung enthalten alle Parameter über beteiligte Geräte. Diese Information braucht eine Anwendung, um die Daten korrekt zu übertragen.

Alle anderen Schichten des Protocol-Stacks sind optional :

TinyTP : Um die Fehlerkorrektur während der Übertragung kümmert sich das Tiny Transport Protocol, vor allem dann, wenn mehr als zwei Geräte über IrDA miteinander kommunizieren.

IrOBEX : Zwischen Anwendungsebene und Protocol-Stack definiert das Infrared Object Exchange Protocol den Austausch von Daten in unterschiedlichen Formaten. Daten können Visitenkarten, Grafiken und andere Dateien sein.

IrCOMM : Die IrCOMM-Erweiterung emuliert einen seriellen und parallelen Port. Anwendungen können dann auf die (virtuellen) Ports zurückgreifen, ohne sich um den kompletten Stack kümmern zu müssen.

IrDA Lite : Abgespeckter Protocol-Stack für Geräte mit wenig Ressourcen. IrDA Lite reduziert die Übertragungsrate auf 9600 Bit/s, kann Geräte nicht automatisch erkennen, und die Kontrolle der IrLAP-Verbindung wird in die Anwendung verlagert.

IrTran-P : Für die Bildübertragung von einer Digitalkamera zum Rechner dient das Protokoll IrTran-P. Die Implementierung verschlingt mit knapp über 10 kByte wenig Ressourcen und kennt mehrere Bildformate.

IrMC : Spezifikation für die Kommunikation mit Mobiltelefonen und den Austausch von Telefonbuch, Kalender und Nachrichten.

IrLAN : Das Protokoll beschreibt die LAN-Verbindung eines IrDA-Gerätes über mehrere Vertriebsmodi (PC an LAN über IrDA, Verbindung eines IrDA-Gerätes mit einem anderen, das an ein Netzwerk angeschlossen ist).

3.4 IrDA-Verbindung

Nun möchte ich erklären, wie eine typische IrDA-verbindung abläuft ! Sobald sich zwei Infrarot-Receiver in IrDA-Reichweite (< 1m) befinden löst das asymmetrische Protocol IrLAP den Verhandlungsvorgang über die Art der Verbindung aus. Gerät 1 sendet am Anfang eine dynamische Adresse mit 9600 Bit/s, somit ist Gerät 1 Master und verantwortlich für die Verbindung ! Gerät 2, das netterweise die Sklavenrolle übernommen hat übermittelt jetzt dem Master seinen Parametersatz. Sobald ein drittes Gerät in Reichweite kommt, muß das IrLMP-Protocol einspringen, da IrLAP nur eine Einzelverbindung kennt ! Das

Protocol arbeitet in einem Multiplex-Verfahren, um Daten seiner Clients zu akzeptieren. Die Clients können sich untereinander nicht verständigen, da Gerät 1 (Master) die Fäden in der Hand hält. Bevor die ersten Daten über die Infrarotdioden ausgetauscht werden, müssen alle Informationen an die IAS-Schicht weitergegeben werden. Ohne IAS weiss der Master nicht, wie er auf Anfragen seiner Clients zu antworten hat um beispielsweise Daten zu übertragen. Jetzt erst startet die Datenübetragung.

3.5 Anwendungsgebiete

IrDA wird vorallem bei Laptops angewendet. Hier dient IrDA unteranderem dazu seinen Terminplaner auf den neuesten Stand zu bringen, Mails auszutasuchen oder sonstiger Datenaustausch zwischen PC und Laptop. Ein anderes Einsatzgebiet ist das Handy (auch Handy und Laptop), wobei es hier fast keine sinnvolle Anwendung gibt außer vielleicht gegen seinen Freund eine Partie 'Snake' zu spielen. Natürlcih bietet sich IrDA gerade für PDAs an. Wer seinen Kleinrechner sowohl zu Hause, am Arbeitsplatz oder unterwegs synchronisiert, braucht beim Einsatz von IrDA keine teueren Kabel zu kaufen.

Alles in allem kann man sagen, daß IrDA wohl keine allzu große Zukunft bevor steht. Neue, bessere Protokolle wie z.B. Bluetooth sind auf dem Vormarsch und werden wohl keine großen Probleme haben IrDA abzulösen. Ein riesiger Nachteil bei IrDA ist unteranderem, daß sich die Geräte gegenüber liegen/stehen müssen, damit eine Verbindung zustande kommt, während man z.B. bei Bluetooth auf Funkverbindungen setzt.

4 WAP

4.1 Einleitung

Dieses Dokument ist nur ein grober Überblick von WAP. Es ist fast unmöglich in entsprechender Zeit sämtliche Dokumente des WAP–Forums durch zu arbeiten. Die Dokumente des WAP–Forums umfassen insgesamt 815 Seiten in englischer Sprache. In den Dokumenten ist zwar sehr genau beschrieben, wie WAP aufgebaut ist, aber aus den Angaben kann man unmöglich einen WAP–Stack programmieren. So ist man darauf angewiesen, das Protokoll zu kaufen, falls man es in Produkten einsetzen will. WAP wurde in Zusammenarbeit mit verschiedenen Mobil–Telefon Herstellern wie Nokia, Motorola, und Ericsson erarbeitet. Genauere Angaben über die Adressen und woher man Programme bekommt sind dem Literaturverzeichnis zu entnehmen.

4.1.1 Was ist WAP

WAP ist die Abkürzung für Wireless Application Protocol und heißt auf deutsch „schnurloses Anwendung Protokoll“. Es wurde initiiert um den Internet–Zugang über ein Handy zu ermöglichen. Der Internet–Zugang soll vor allem die Möglichkeit bieten über ein Handy Nachrichten zu senden und zu empfangen, das Wetter und Aktien–Kurse abzurufen, über das Internet einzukaufen, Geld Transaktionen vorzunehmen und ein Navigationssystem zur Verfügung zu stellen.

WAP ist nicht ein Protokoll, das auf einer Schicht des ISO/OSI Schichtenmodells arbeitet, wie es viele fälschlicherweise meinen, sondern es setzt sich dabei aus verschiedenen Schichten zusammen. Allerdings erinnert es sehr an das ISO/OSI Schichtenmodell. Jede Schicht stellt bestimmte Dienste zur Verfügung. Unter anderem übernehmen sie die Aufgabe zum Transport der Daten, Sicherheit und Anwendungen wie E–Mail, oder Browser.

All diese Schichten zusammen nennt man WAP.

4.1.2 Wozu braucht man WAP

Ziel von WAP ist es den Internet–Zugang für digitale, schnurlose Telefone und andere schnurlose Terminals zu realisieren. Dazu muß man ein Weltweit gültiges Protokoll für schnurlose Telefone - das auf den verschiedensten Netzen funktioniert - erstellen. Dabei wurden vorhandene Standards und Technologien wo immer sie verwendet werden in WAP eingebunden und für die Bedürfnisse von WAP verbessert.

Sowohl das Internet, als auch Mobiltelefone nehmen sehr stark an Bedeutung zu. Immer mehr Leute besitzen ein Handy und gehen ins Netz der Netze. Da bietet es sich geradezu an, die beiden Technologien miteinander zu verbinden.

4.1.3 Was ist das besondere an WAP

WAP muß speziellen Anforderungen gerecht werden. So hat ein Handy beispielsweise ein kleines Display auf dem die Informationen dargestellt werden müssen.

Schnurlose Terminals haben eine leistungsschwache CPU, wenig Speicher, kleine Displays, die verschiedensten Eingabegeräte und eine niedrige Bandbreite.

Dazu kommt, daß die Hersteller eine vielseitige Einsetzbarkeit auf den verschiedensten Terminals benötigen. Dabei muß eine automatische Skalierung erfolgen, wie der Benutzer bzw. das Handy sie benötigt. Es muß eine hohe Zuverlässigkeit und Sicherheit bieten und Effizient muß das ganze auch noch sein.

All dies wird mehr oder minder von WAP unterstützt.

4.2 Aufbau von WAP

Das WAP–Modell erinnert sehr stark an das Modell des World–Wide–Web. Dies ist damit begründet, um vorhandene Tools und Dienste einbinden zu können, wie zum Beispiel einen Web–Server. Der Mikro–Browser ist analog zum Web–Browser. WAP definiert ein Set von Standardkomponenten zur Realisierung der Kommunikation zwischen Mobilen Terminals und Web–Servern. Dabei wird das Standard – Namens – Modell des WWW implementiert.

WAP realisiert die Kommunikation vom Mobil–Netz ins Internet. Es beinhaltet Kalender Informationen, elektronische Geschäfte, Bilder und Skript–Sprache.

WAP ist der Oberbegriff einer ganzen Reihe von Protokollen, Diensten und Applikationen. Im Anhang A befindet sich über folgendes eine Grafik.

Das Schichtenmodell von WAP erinnert sehr stark an das ISO/OSI–Schichtenmodell. Es besteht ebenfalls aus mehreren Schichten, die ähnliche Aufgaben wie die der OSI–Schichten übernehmen. Auf der untersten Schicht befinden sich die „Bearers“. Auf der zweiten Schicht ist der Transport Layer (WDP) zu finden. Die dritte Schicht wird vom Security Layer (WTLS) belegt. Die nächste Schicht beinhaltet den Transaction Layer (WTP). Schließlich ist auf der fünften Schicht der Session Layer (WSP) zu finden und auf der sechsten Schicht der Application Layer (WAE).

4.2.1 Die Bearers

Sie sind das Pendant zum Physical Layer oder auch der Bitübertragungsschicht. Darin beinhaltet sind die Standards von GSM, IS-136, CDMA, PHS, CDPD, PDC-P, iDEN, FLEX, etc. Die Bearers unterstützen verschiedene Stufen von QoS (Quality of Service), d.h. die Verbindungsqualität und die Bandbreite kann beeinflußt werden.

4.2.2 Transport Layer (WDP–Wireless Datagram Protocol)

Auf der zweiten Schicht befindet sich der Transport Layer (WDP). Der Transport Layer hat die Aufgabe, die Daten zu transportieren wie bei OSI. Dabei wird eine Verbindung zwischen dem Physical Layer (der Bitübertragung) und dem Security Layer hergestellt. Das WDP erlaubt transparente Operationen mit verschiedenen Services. IP geht über UDP. TCP scheint nicht verwendet zu werden (Siehe Anhang B).

4.2.3 Security Layer (WTLS–WirelessTransportLayerSecurity)

Die dritte Schicht ist für die Sicherheit zuständig. Es handelt sich dabei um den Security Layer (WTLS). Der Security Layer unterstützt Sicherheitsaspekte wie Datenintegrität (Sicherstellung, daß die Nachricht nicht verändert wurde), Privat (verschlüsselte Pakete, für andere unverständlich) und Authentifizierung (Überprüfung der Echtheit).

Der Ablauf funktioniert in etwa so, daß der Client ein „Hello“ an den Server schickt. Dieser beantwortet den „Hello“ in der Regel. Bei der Übertragung werden Zufallszahlen übermittelt, die für die spätere Codierung benötigt werden und den „pre_master_secret“ Schlüssel bilden. Zur Erstellung eines Schlüssels existiert ein einheitlicher Algorithmus: master_secret = PRF (pre_master_secret, „master secret“, ClientHello.random + ServerHello.random);

Der „master_secret“ Schlüssel ist immer 20 Byte lang.

Nach diesem Szenario gibt es dann 3 Möglichkeiten der Verschlüsselung. Die eine ist eine RSA Verschlüsselung. Diese Verschlüsselung wird für die Server Authentifizierung und den Schlüsselaustausch benötigt. Dabei wird der Schlüssel vom Client mit dem Public Key des Servers verschlüsselt und an ihn übermittelt. Der Server entschlüsselt den Schlüssel mit seinem Private Key. Bei dieser Verschlüsselung handelt es sich also um eine Asymmetrische Verschlüsselung. Nun kann der Server Daten mit dem Symmetrischen Schlüssel des Clients übermitteln.

Eine andere Möglichkeit ist der Diffie-Hellman Algorithmus. Dieser funktioniert gleich, wie die RSA Verschlüsselung, jedoch pre_master_secret Schlüssel invertiert.

4.2.4 Transaction Layer (WTP–Wireless Transport Protocol)

Auf der vierten Schicht arbeitet der Transaction Layer (WTP). Er stellt die Verbindung her, wobei die Verbindung eine Ein–Wege–Verbindung oder eine Zwei–Wege–Verbindung sein kann. Die Ein–Wege–Verbindung kann dabei unzuverlässig (Verbindungslos) oder zuverlässig (Verbindungsorientiert) sein. Dabei handelt es sich um eine asynchrone Datenübertragung. Die Empfangsbestätigung erfolgt verzögert, um

die Anzahl der Nachrichten zu reduzieren. Das könnte man entfernt mit dem Silly Window Syndrome vergleichen. Beim Silly Window Syndrome teilt der Empfänger dem Sender jede kleine Änderung der Window–Size mit und der Sender schickt ein neues Paket. Die folge daraus ist, daß der Hauptteil der Übertragung darin besteht, Steuersignale (Acknowledgements, ...) zu übertragen, anstelle von Daten. Im WTP wird auch eine Fehlerbehandlung vorgenommen. Die Timer sind hier angesiedelt. WTP ist also für den Nachrichtentransfer zuständig. Informationen über das letzte Acknowledgement werden gespeichert.

4.2.5 Session Layer (WSP–Wireless Session Protocol)

Der Session Layer (WSP) befindet sich auf der fünften Schicht. Er besitzt eine HTTP/1.1 Funktionalität, ist jedoch für geringe Bandbreiten optimiert. Er stellt außerdem die Verbindung zu einem WAP–Proxy–Server zur Verfügung. Die Verbindungen sind Verbindungslos.

4.2.6 Application Layer (Wireless Application Environment)

Auf der sechsten und letzten Schicht arbeitet der Application Layer (WAE). Dieser beinhaltet einen Mikro–Browser, der eine spezielle Sprache darstellen kann. Bei dieser Sprache handelt es sich um WML (Wireless Markup Language). Sie ist sehr ähnlich zu HTML. Solche Software–Programme wie der Mikro–Browser und die WML Sprache bezeichnet man als User Agent. Wie auch bei HTML wird eine Skript–Sprache Unterstützt. Diese Skript–Sprache hat eine große Ähnlichkeit zu Java–Skript. Des weiteren bietet WAE die Verbindung zu einem speziellen WML–Server (WTA – Wireless Telephony Application).

WAE soll kompatibel sein zu HDML (Handheld Markup Language), HTML (Hypertext Markup Language) und WWW Technologien wie URL und HTTP (Uniform Ressource Locator und Hypertext Transfer Protocol).

Es können auch Applikationen auf anderen Servern gestartet werden, wie CGI–Skripte.

WAE ist in zwei logische Schichten unterteilt, nämlich User Agents (Browser, PhoneBook, Message Editors) und Services, welche dem User Agent Zugriff auf WML, WML–Skript, image–Formate, vCards und vCalender ermöglichen.

Parallel zu den Schichten können ab der dritten Schicht (WTLS) zusätzliche Dienste und Applikationen laufen. Dabei kann es sich zum Beispiel um E–Mail, anzeige von Bildern, Kalender Information, Geldtransaktionen, E–Commerce, Telefonbucheinträge, etc. handeln. Es wurde ein spezielles Bitmap–Format erstellt. Es nennt sich WBMP (Wireless Bitmap). Diese Spezifikation beinhaltet die Höhe und Breite, die Versionsnummer, ein kompaktes binär–Format, die Skalierbarkeit (Farbtiefe, Animation, Datenstrom), eine Pixel Organisation, eine Kompression und Animation.

WAP unterstützt keine zusätzlichen Protokolle wie beispielsweise FTP.

4.2.7 Verbindung ins Netz

Die Verbindung ins Netz der Netze kann über zwei Wege erfolgen. Die eine Möglichkeit ist, einen speziellen WML–Server (WTA–Server) zu kontaktieren, der speziell für WML–Seiten eingesetzt wird. Die zweite ist, über einen WAP–Gateway die Verbindung ins Internet herzustellen. Dabei können die Web–Seiten bereits im WML–Format auf einem Web–Server vorliegen, oder es kann eine HTML–Seite aufgerufen werden, die dann durch einen HTML–Filter läuft und die Seite in WML umwandelt (Siehe Anhang C). Bei dieser Umwandlung werden gegebenenfalls unnötige Informationen, wie Werbung oder Images einfach „raus geschmissen“.

Im allgemeinen ist ein WAP–Server oder ein WTA–Server fest eingestellt. Auf diesen werden dann WML–Seiten zur Verfügung gestellt wie der Abruf von Aktienkursen oder die Wetterdaten. Innerhalb dieser Seite navigiert man sich dann hindurch. Will man andere Funktionen nutzen, verbindet man sich einfach - mit einem erneuten Verbindungsaufbau - mit einem anderen Server. Anbieter wie D1, D2, E–Plus, VIAG Interkom stellen solche WML Seiten mit unterschiedlichem Leistungsumfang zur Verfügung. Will man heute Dienste von D1 nutzen, ruft man einfach den WAP–Server von D1 an und morgen vielleicht den WAP–Server von D2. Innerhalb dieser Seiten findet sich meist eine Menü–Struktur, durch die man sich dann „durch klickt“.

Die WML–Dateien werden in einem Bytecode (im binär–Format) übertragen, um zusätzlich Bandbreite einzusparen. Für die Verbindung benötigt WAP eine Full–Duplex Verbindung. Der allgemeine Verbindungsaufbau wird in einer Grafik in Anhang D dargestellt.

4.2.8 WML (Wireless Markup Language)

WML ist eine Sprache ähnlich dem HTML. Sie basiert auf XML. Wie bei HTML unterstützt WML die Textausgabe und Darstellung von Images. Jedoch sollte bei den Images beachtet werden, daß auf den Handys derzeit keine Farbbilder dargestellt werden können. Davon abgesehen benötigen Farbbilder, oder Graustufen–Bilder sehr viel Speicherplatz im Gegensatz zu Schwarz–Weiß Bildern, was für die Datenübertragung wieder negative Konsequenzen hat. In naher Zukunft wird dies allerdings kein Problem mehr sein, da die Bandbreite von GSM stark erhöht werden soll. Die zukünftigen Handys werden auch größere Display besitzen.

Der Aufbau einer WML–Datei ähnelt einer Software aus der Macintosh–Welt. Dieses Programm heißt Hypercard. Wie die MAC–Sprache unterscheidet WML sogenannte Karten (cards) und Kartenstapel (decks). Der Benutzer navigiert zwischen den einzelnen Karten eines Stapels, liest ihre Inhalte oder gibt Informationen ein. Decks können parametrisiert werden. So kann zur Laufzeit ein unterschiedliches aussehen der WML–Seite, abhängig von den Parametern, definiert werden.

WML–Dokumente haben für die Navigation in den Karten und Stapeln ID´s und Classes, damit eine Karte und ein Stapel identifiziert werden können. Eine relative Adressierung ist möglich. Sie richtet sich nach dem Unified Naming Model, und nach den Uniform Ressource Locators (URL). WML–Dokumente werden wie HTML interpretiert. Die einzelnen Strings werden dabei auseinander genommen und mit einem Parser interpretiert. WML unterstützt Alphanumerische Zeichen. Der Unicode Zeichensatz wird ebenfalls unterstützt, um ein weltweites verbreiten der WML–Seiten zu ermöglichen. Unter WML können Meta–Angaben definiert werden. Somit ist es möglich, WML–Seiten in einer Meta–Suchmaschine sichtbar zu machen.

WML wurde für kleine Bildschirmgrößen, schlechte Eingabemöglichkeiten, geringe Rechenleistung, wenig Speicher, geringe Bandbreite, etc. konzipiert. Wie HTML unterstützt WML eine ganze Reihe Besonderheiten. So kann man beispielsweise Listen, Eingabefelder, Timer und Tabellen definieren.

Auf jedes einzige Detail einzugehen würde hier zu weit gehen. Dies soll keine Anleitung dafür werden, wie man WML programmiert. Deshalb ist im nächsten Abschnitt nur ein kleines Programmbeispiel abgedruckt. Ansonsten verweise ich auf das Literaturverzeichnis. Der Code wird anschließend erläutert.

4.2.9 WML–Beispiel

```
1.  <wml>
2.      <card>
3.          <p>
4.              <do type="accept">
5.                  <go href="#card2"/>
6.              </do>
7.              Hello world!
8.          </p>
9.      </card>
10.     <card id="card2">
11.         <p>
12.             Auf Wiedersehen !
13.         </p>
14.     </card>
15. </wml>
```

Hier kann man schon die sehr ähnliche Struktur zu HTML sehen. Wie auch bei HTML muß das Programm erst mit einem TAG eingeleitet werden. Hier heißt dieser natürlich <wml>.

In der zweiten Zeile wird dann schon die erste Karte definiert. Die ID ist optional, wenn man jedoch auf eine Karte springen will, muß eine ID angegeben werden. So wird hier nur eine Karte definiert, da auf diese Karte kein Link mehr gesetzt wird.

In der dritten Zeile steht ein , was die selbe Bedeutung hat wie in HTML, nämlich einen Absatz.

Die vierte Zeile aktiviert den folgenden Link. Natürlich müssen wie bei HTML geöffnete TAG´s wieder geschlossen werden, was hier in der sechsten Zeile geschieht, indem dem TAG ein Slash vorgesetzt wird (</do>).

In der fünften Zeile befindet sich der eigentliche Link, der auf die Karte 2 zeigt. Hier ist auch wieder die Ähnlichkeit zu HTML zu sehen, da der Befehl „href“ benutzt wird.

In der siebten Zeile steht einfacher Text, der innerhalb der Karte 1 angezeigt wird.

 in der achten Zeile beendet den Absatz–TAG.

In Zeile neun wird die Karte mit dem </card> TAG nun geschlossen. Somit ist die erste Karte fertig konfiguriert und abgeschlossen.

Nun wird in Zeile zehn die zweite Karte definiert. Hier muß nun die ID angegeben werden, da Karte 1 auf diese Karte mit einem Link verweist.

Der Rest verhält sich gleich, wie in der ersten Karte. In Zeile zwölf wird einfach nur noch ein Text ausgegeben.

Zum Schluß muß der wml–TAG noch geschlossen werden. Dies geschieht in Zeile fünfzehn mit </wml>. Dieses gesamte Beispiel bildet nun ein „deck“ oder zu Deutsch einen sogenannten Stapel.

4.2.10 WML–Script

Das Wireless Application Protocol bzw. der Mikro–Browser unterstützt ebenfalls Skripts. Die Skripts ähneln syntaktisch sehr stark dem Java–Skript. Wie auch WML werden aus Performance gründen die Skripts in Bytecode kompiliert und so übertragen. Dennoch benötigen sie zusätzliche Bandbreite. Also je mehr Skripts in einer WML–Datei verwendet werden, desto länger dauert der Download der Seite.

Skripts ermöglichen unter anderem Rechenoperationen oder Paßwort–Abfragen. Wie auch in Java sind Datentypen wie int (Integer = Ganze zahlen), float (Fließkomma zahlen) und String (Zeichenketten) definiert.

Literale, Identifizierer, Funktionen, Rückgabeparameter, for–Schleifen, while–Schleifen, break, continue und Umwandlungen (von zum Beispiel Datentypen) werden zur Verfügung gestellt. Es gibt auch vorgefertigte Methoden, wie round(), sqrt(), isEmpty(), charAt(), substring(), find(), replace(), elements(), elementAt(), removeAt(), replaceAt(), insertAt(), compare(), toString(), getHost(), getPort() und getPath(), die ohne größere Komplikationen verwendet werden können.

Einige Beispiele:

- round() rundet eine Fließkomma Zahl (float) zu einer Ganzen Zahl (int)
- sqrt() zieht die Quadratwurzel aus einer Zahl
- isEmpty() überprüft, ob ein Objekt leer ist. Die Rückgabe ist eine
- boolean (true / false)
- charAt() gibt das Zeichen an einer bestimmten Position in einem String zurück.
- insertAt() fügt ein Zeichen an einer bestimmten Position in einen String ein.
- compare() vergleicht zwei String´s miteinander.
- toString() wandelt einen anderen Datentyp in einen String um.
- getHost() liefert die Host – Adresse, nach der man frägt.

In WML–Skript wird das selbe Fehlerbehanlungs–Modell wie bei Java und Java–Skript eingesetzt.

Um noch näher auf WML–Skript einzugehen, würde an dieser Stelle zu weit führen.

4.3 Schlußteil

4.3.1 Anwendungsbeispiele

Die WAP–Technologie könnte beispielsweise in Navigationssystemen eingesetzt werden. Hier liegt der entscheidende Vorteil darin, daß für Fahrtrouten dinge wie Baustellen oder Staus gleich berücksichtigt werden können und entsprechend umfahren werden können.

Zur Abfrage von Aktienkursen kann es auch eingesetzt werden. Somit kann man schneller agieren, wenn man beispielsweise mit dem Zug unterwegs ist.

Man kann das Wetter abrufen, um sich zum Beispiel seine Kleidung auszuwählen, bevor man irgendwo hin geht.

Man kann e–mails schreiben um auch während einer Geschäftsreise in schriftlichem Kontakt mit einem Geschäftspartner zu bleiben.

Man kann bei Online–Auktionen über´s Netz mit bieten und gute Schnäppchen erzielen.

Ob diese Beispiele für jeden einzelnen Sinn machen, muß jeder für sich selbst entscheiden. Es sind wie gesagt nur Beispiele.

4.3.2 Schlußwort

Insgesamt ist das WAP–Modell ein gut ausgetüfteltes Schema. Es ist schon beeindruckend, wie auf solch kleinen Displays noch so viel dargestellt werden kann. Auch die Navigationsmöglichkeiten sind gut gestaltet. Es wäre ein Unding, wenn man eine 20–stellige URL über ein Nummern–Pad eingeben müßte, wobei eine solch lange URL nichts ungewöhnliches ist. Das auch HTML–Seiten betrachtet werden können, ist ebenfalls eine tolle Sache.

Sicher ist noch nicht alles zu 100% ausgereift, aber wann war jemals ein Standard oder ein Software–Produkt gleich beim ersten Anlauf Perfekt?

Ob die Entwicklung - und wo sie hin führt - jeder einzelne gut findet, oder ob das Ganze eine Spielerei ist muß jeder für sich selbst entscheiden.

Ich persönlich halte diese Technik für eine „Spinnerei“. Letztendlich werden wir immer Häufiger mit einer Datenflut überschüttet, da muß man froh sein, wenn man mal ruhe davor hat. Es werden so wie so nur die „wichtigen“ Leute ein solches Handy besitzen und einsetzen. Ich kenne sehr viele Geschäftsführer, die kein Handy besitzen.

Die einzige Einsatzmöglichkeit die ich für sinnvoll halte ist im Navigation–System–Bereich.

4.4 Anhang

4.4.1 Anhang A: WAP-Schichtenmodell

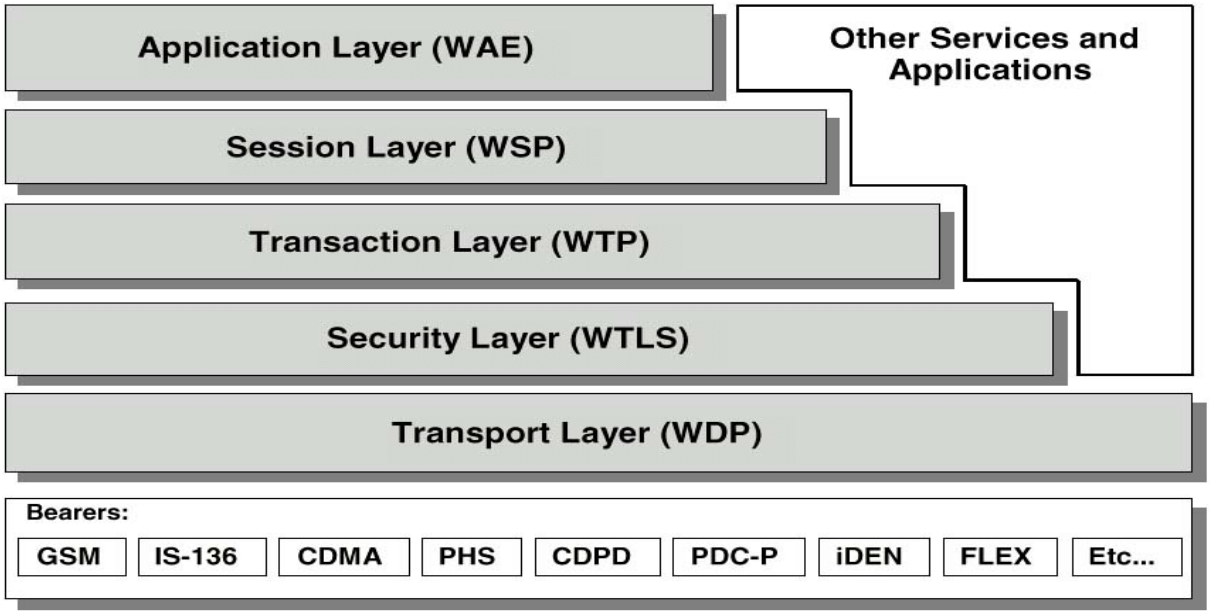


Figure 4. WAP Architecture

4.4.2 Anhang B: Stack-Beispiel

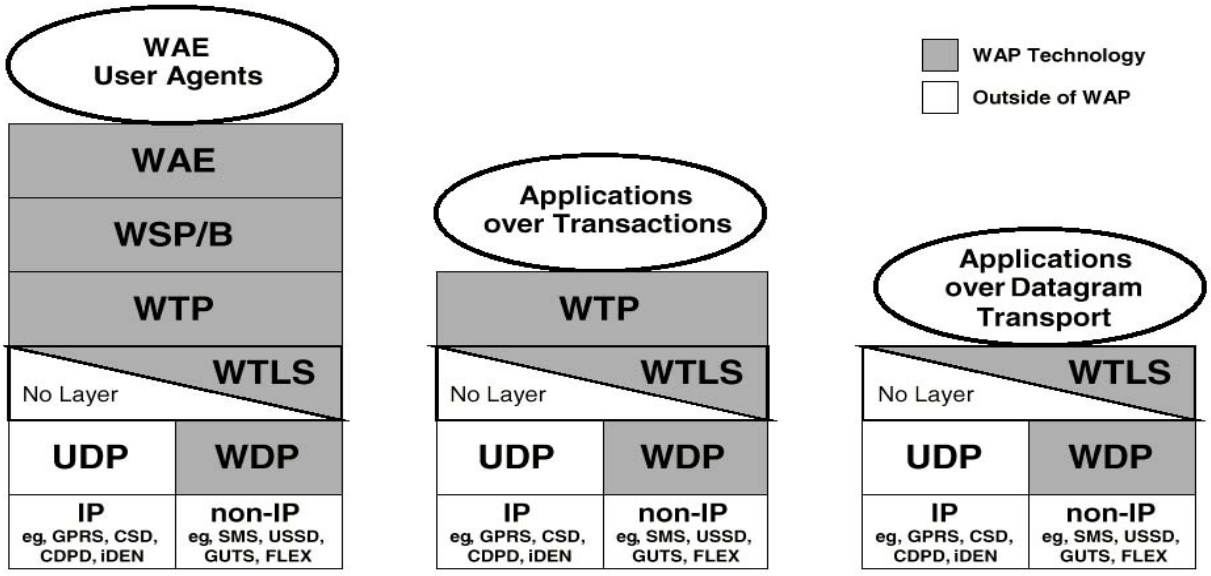


Figure 5. Sample WAP Stacks

4.4.3 Anhang C: HTML-Filter-Schema

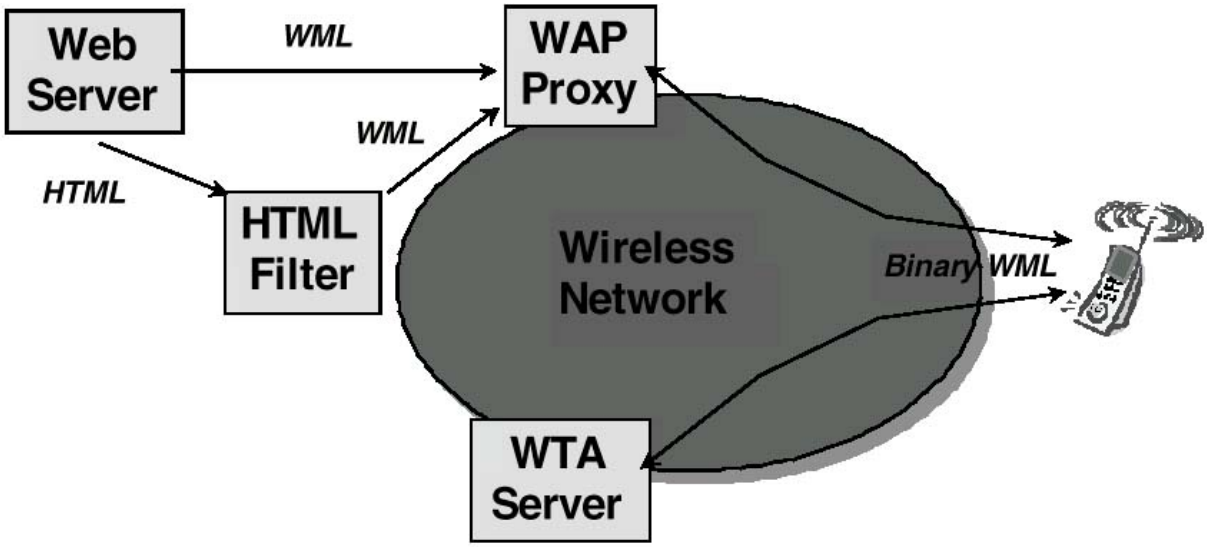


Figure 3. Example WAP Network

4.4.4 Anhang D: Verbindungsschema

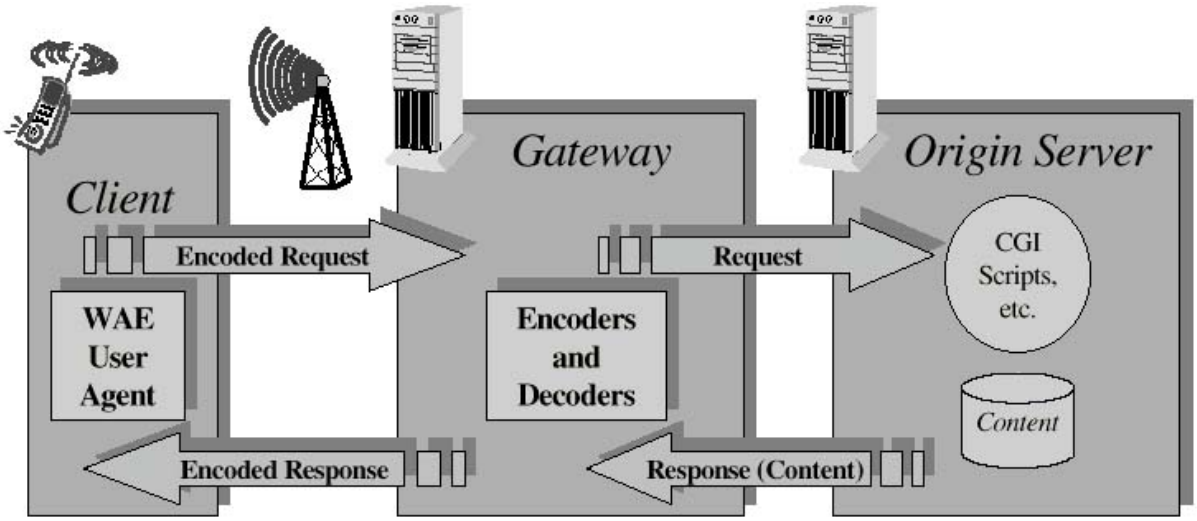


Figure 2. WAP Programming Model

5 Fazit

Mobile Dienste können sehr wichtig und sinnvoll sein. Abschließend kann man sagen, daß die drei Beispiele nach ihren Anforderungen sehr gut implementiert sind. Alle drei mobilen Protokolle und Dienste erfüllen fast vollständig die Anforderungen, die an mobile Dienste gestellt werden. Optimierungen der Datenübertragung, wegen der geringen Bandbreiten und Sicherheitsaspekte werden dabei ausreichend berücksichtigt. Auch die Tatsache, daß man eventuell ein sehr kleines Display und schlechte Eingabemöglichkeiten zur Verfügung hat, werden ausreichend berücksichtigt.

Jedoch stellt sich die Frage, wie sinnvoll die Implementationen sind. IrDA beispielsweise benötigt Sichtkonakt, Sender und Empfänger müssen genau aufeinander gerichtet werden und die Reichweite ist sehr beschränkt.

Mit WAP hingegen kann man weltweit im Internet Surfen, jedoch mit den derzeitigen Möglichkeiten wie Handy´s macht das recht wenig Sinn.

Bluetooth hingegen bietet die Möglichkeit in einem beispielsweise Denkmal geschützten Gebäude ein internes Netz aufzubauen, oder eine Maus, Tastatur oder einen Drucker Kabellos an einen Rechner anzubinden. Außerdem kann man zusätzlich auf Bluetooth IrDA oder WAP aufsetzen.

6 Verzeichnisse

6.1 Quellenverzeichnisse

6.1.1 Bluetooth

Specification of the Bluetooth System	Specification Volume 1 / core	Bluetooth special interest group (SIG)	http://www.bluetooth.com
Specification of the Bluetooth System	Specification Volume 2 / profile	Bluetooth special interest group (SIG)	http://www.bluetooth.com
Link Manager Protocol	Presentation-Documents / Developer Converence, London, June 1999	Thomas Müller (Nokia)	tho-mas.t.muller@nmp.nokia.com
Privatfunk	Bluetooth als Netzwerk- und Schnittstellenmodul	Verlag Heinz Heise GmbH & Co KG	c't 25/99, Seite 126, Dusan Zivadinovic

6.1.2 IrDA

Jürgen Rink, Morgenröte, Der IrDA-Standard für die Datenübertragung mit Infrarotlicht, c't 6/98, S. 316

Jürgen Rink, Infra-Rotlichtbezirk, IrDA-Verbindungen mit PC, Notebook, PDA und Mobiltelefon, c't 25/99, S. 114

Infrared Data Association, http://www.irda.org

6.1.3 WAP

- www.wapforum.org (White Papers ohne Ende)
- www.phone.com (White Papers und Development Kit)
- www.nokia.com (SDK – Development Kit for WML)
- www.ericsson.com (SDK – Development Kit for WML)
- www.virtuacom.com/wap (WML – Browser für Windows)
- www.uplanet.com
- www.wapnet.com
- www.wap-portal.de
- www.ginco.de/wap
- www.heise.de (Verlag von z.B. C´t)
- C´t Heft Nr. 22 1999

6.2 Glossar

6.2.1 Bluetooth

ACL	Asymetric Connection Less (verbindungslose Übertragungsart von BT)	LMP	Link-Management-Protocol (Bestandteil der BT-Protokollstacks)
ARQ	Automatic Repeat Request (Verfahren zur Fehlersicherung)	RFCOMM	einfaches serielles Interface, vergleichbar mit V.24
CRC	Cyclic Redundancy Check (Verfahren zur Fehlersicherung)	SAR	Segmentation und Reassembling
FEC	Forward Error Correction (Verfahren zur Fehlersicherung)	SCO	Syncronuous Connection Oriented (verbindungsorientierte Übertragungsart von BT)
GFSK	Gaussian-Frequency-Shift-Keying (Modulationsverfahren)	SDP	Service Discovery Protocol
IrOBEX	Object Exchange Protocol	TDD	Time-Division-Duplex (Übertragungsmodus)
L2CAP	Logical Link-Control and Adaption-Protocol (Bestandteil der BT-Protokollstacks)	TSC	Telephony control Protocol

6.2.2 IrDA

FIR :	Fast Infrared
IrDA :	Infrared Data Association
Konvexlinse :	Sammellinse
LAN :	Local Area Network
SIR :	Standard Infrared
Tranceiver :	Wort das sich aus den Begriffen 'Transmitter' und 'Receiver' zusammensetzt.
VFIR :	Very Fast Infrared

6.2.3 WAP

A	API	**Application Programming Interface**
	A-SAP	**Application Service Access Point**
B	BMI	**Base Station, MSC, Interworking Function (IWF)**
	BNF	**Backus-Naur Form**
	BSD	**Berkeley Software Distribution**
C	CA	**Certification Authority**
	CBC	**Cipher Block Chaining**
	CDMA	**Code Division Multiple Access**
	CDPD	**Cellular Digital Packet Data**
	CGI	**Common Gateway Interface**
	CS	**Cell Station**
	CSD	**Circuit Switched Data**
D	DBMS	**Database Management System**
	DCS	**Digital Communications System**
	DH	**Diffie-Hellman**
	DM	**DataTAC Messaging**
	DTD	**Document Type Definition**
E	EC	**Elliptic Curve**
	ECC	**Elliptic Curve Cryptography**
	ECDH	**Elliptic Curve Diffie-Hellman**
	ECDSA	**Elliptic Curve Digital Signature Algorithm**
	ECMA	**European Computer Manufacturer Association**
	ETSI	**European Telecommunication Standardisation Institute**
G	GC	**Garbage Collection**
	GPRS	**General Packet Radio Service**
	GSM	**Global System for Mobile Communication**
	GTR	**Group Trailer, indicates the end of packet group**
	GUTS	**General UDP Transport Service**
H	HDML	**Handheld Markup Language**
	HLR	**Home Location Register**
	HTTP	**Hypertext Transfer Protocol**
	HTML	**HyperText Markup Language**
I	IANA	**Internet Assigned Number Authority**
	iDEN	**Integrated Digital Enhanced Network**
	IE	**Information Element**
	IMC	**Internet Mail Consortium**
	IN	**Intelligent Network**
	IP	**Internet Protocol**
	IS-136	**TDMA PCS–Radio Interface–Mobile Station–Base Station Compatibility Standard**
	ISO	**International Organization for Standardization**
	IV	**Initialisation Vector**
	IWF	**Interworking Function**
L	LAPi	**Link Access Protocol iDEN**
	LSB	**Least Significant Bits**
M	MAC	**Medium Access Control**
	MAC	**Message Authentication Code**
	MAP	**Mobile Application Part**
	MDBS	**Mobile Data Base Station**
	MDG	**Mobile Data Gateway**
	MD-IS	**Mobile Data - Intermediate System**
	MDLP	**Mobile Data Link Protocol**
	ME	**Management Entity**
	MGL	**Maximum Group Length**
	MMI	**Man-Machine Interface**
	MOM	**Maximum Outstanding Method requests**
	MOP	**Maximum Outstanding Push requests**
	MPL	**Maximum Packet Lifetime**
	MPS	**Maximum Packet Size**
	MRU	**Maximum Receive Unit**
	MS	**Mobile Station**
	MSB	**Most Significant Bits**
	MSC	**Mobile Switch Centre**
	MSISDN	**Mobile Station International Subscriber Device Number**
	MSS	**Maximum Segment Size**
N	NCL	**Native Command Language**
	NEI	**Network Element Identifier**
O	OS	**Operating System**
	OSI	**Open System Interconnection**
P	PCI	**Protocol Control Information**
	PCS	**Personal Communications System**
	PDA	**Personal Digital Assistant**
	PDC	**Personal Digital Cellular**
	PDLP	**Packet Data Link Protocol**
	PDU	**Protocol Data Unit**
	PHS	**Personal Handy Phone System**
	PICS	**Protocol Implementation Conformance Statement**
	PLMN	**Public Land Mobile Network**
	PPP	**Point-to-Point Protocol**
	PRF	**Pseudo-Random Function**

R	RAS	**Remote Access Server**
	R-Data	**Relay Data**
	RFC	**Request For Comments**
	RFCL	**Radio Frequency Convergence Layer**
	RLP	**Radio Link Protocol**
	RTT	**Round-Trip Time**
S	SAP	**Service Access Point**
	SAR	**Segmentation and Re-assembly**
	SCR	**Standard Context Routing**
	SDU	**Service Data Unit**
	SEC-SAP	**Security Service Access Point**
	SGML	**Standardised Generalised Markup Language**
	SNDCP	**SubNetwork Dependent Convergence Protocol**
	SHA-1	**Secure Hash Algorithm**
	SMS	**Short Message Service**
	SMSC	**Short Message Service Center**
	SMSCB	**Short Message Service Cell Broadcast**
	SMTP	**Simple Mail Transfer Protocol**
	SPT	**Server Processing Time**
	S-SAP	**Session Service Access Point**
	SS7	**Signalling System 7**
	SSAR	**Simplified Segmentation and Reassembly**
	SSL	**Secure Sockets Layer**
T	TCAP	**Transaction Capability Application Part**
	TCP/IP	**Transmission Control Protocol/Internet Protocol**
	TDMA	**Time Division Multiple Access**
	TIA/EIA	**Telecommunications Industry Association/Electronic Industry Association**
	TID	**Transaction Identifier**
	TLS	**Transport Layer Security**
	TOD	**Time Of Day**
	TR-SAP	**Transaction Service Access Point**
	T-SAP	**Transport Service Access Point**
	TTR	**Transmission Trailer**
U	UCS	**Universal Multiple-Octet Coded Character Set**
	UCS-4	**Universal Character Set - 4 byte**
	UDCP	**USSD Dialogue Control Protocol**
	UDH	**User-Data Header**
	UDHL	**User-Data Header Length**
	UDL	**User-Data Length**
	UDP	**Unreliable Datagram Protocol**
	UI	**User Interface**
	URI	**Uniform Resource Identifier**
	URL	**Uniform Resource Locator**
	URN	**Uniform Resource Name**
	USSD	**Unstructured Supplementary Service Data**
	USSDC	**Unstructured Supplementary Service Data Centre**
	UTF	**UCS Transformation Format**
	UTF-8	**UCS Transformation Format 8 [ISO10646]**
V	VLR	**Visitor Location Registry**
	VPLMN	**Visitor Public Land Mobile Network**
W	W3C	**World Wide Web Consortium**
	WAE	**Wireless Application Environment**
	WAP	**Wireless Application Protocol**
	WBMP	**Wireless BitMaP**
	WDP	**Wireless Datagram Protocol**
	WML	**Wireless Markup Language**
	WORM-ARQ	**WORM-Auto Repeat Request**
	WSP	**Wireless Session Protocol**
	WTA	**Wireless Telephony Applications**
	WTAI	**Wireless Telephony Applications Interface**
	WTLS	**Wireless Transport Layer Security**
	WTP	**Wireless Transport Protocol**
	WWW	**World Wide Web**
X	XML	**Extensible Markup Language**