

SIMULATION UND TEST DES D-KANAL PROTOKOLLS

ERWIN P. RATHGEBB
 Institut für Nachrichtenvermittlung und Datenverarbeitung
 Universität Stuttgart

HONG LINH TRUONG, WERNER KIRCHNER
 Standard Elektrik Lorenz AG, Stuttgart

ZUSAMMENFASSUNG

In diesem Bericht sollen einige Aspekte des Tests von Protokollimplementierungen dargestellt werden. Bedingt durch die immer komplexer werdenden Kommunikationsprotokolle gewinnt dieser Bereich mehr und mehr an Bedeutung. In dem Beitrag soll, nach der Einführung einiger Grundbegriffe und Methoden, am Beispiel des D-Kanal-Protokolls die Entstehung einer Sammlung von Testfällen, ihre Implementierung und das dazu verwendete Testsystem, der "ISDN D-Kanal Simulator", vorgestellt werden.

1. EINLEITUNG

Der teilnehmerseitige Zugang zum ISDN wird durch die sogenannte S-Schnittstelle mit den zwei 64 kbit/s-Nutzkanälen (B-Kanäle) und dem 16 kbit/s-Zeichengabekanal (D-Kanal) gebildet. Die B-Kanäle werden für die Übertragung von Nutzinformation benutzt. Im D-Kanal werden u.a. Signalisierungsinformationen zwischen Endeinrichtungen und Vermittlungsstellen ausgetauscht.

Entsprechend dem OSI-Referenzmodell kommen für das Signalisierungsprotokoll im D-Kanal die drei unteren Schichten zur Anwendung. In der Schicht 1 wird neben den elektrischen und physikalischen Eigenschaften der S-Schnittstelle ein Mehrfach-Zugriffsverfahren definiert, welches die gemeinsame Benutzung des D-Kanals durch mehrere Endgeräte ermöglicht.

Das D-Kanal-Protokoll in der Schicht 2 (LAP D) kann als Anpassung der Datenübertragungsprozedur LAP B der CCITT-Empfehlung X.25 an die Besonderheiten des ISDN-Basisanschlusses angesehen werden. Wegen der Busfähigkeit der S-Schnittstelle unterstützt LAP D das gleichzeitige Betreiben von mehreren Schicht-2-Verbindungen. Außerdem bietet LAP D den höheren Schichten die Möglichkeit, eine Nachricht an alle am S-Bus angeschlossenen Endgeräte zu verteilen.

In der Schicht 3 sind neben den Signalisierungsprozeduren für den Aufbau und Abbau von Nutzkanalverbindungen ebenfalls die Prozeduren für die Benutzung und Steuerung von Dienstmerkmalen definiert.

Die obige kurze Beschreibung des D-Kanal-Protokolls hebt deutlich die Komplexität des Protokolls und daraus folgend die Notwendigkeit für Methoden und Einrichtungen zum Testen dieses Protokolls hervor. Nicht nur die Hersteller müssen ihre ISDN-Komponenten während der Entwicklung testen, auch die Netzbetreiber benötigen entsprechende Geräte

und (möglichst einheitliche) Verfahren zur Beurteilung von Komponenten, die zum Betrieb am Netz zugelassen werden sollen.

In diesem Bericht sollen nun einige Aspekte des Tests für die Signalisierungsprotokolle im D-Kanal dargestellt werden. Nach einer kurzen, einführenden Übersicht über die grundlegenden Methoden des Protokolltests beschäftigt sich der Beitrag mit einigen grundsätzlichen Überlegungen, die bei der Spezifikation der Testfälle angestellt worden sind.

Danach wird die Konzeption des "D-Kanal Simulators", eines Systems zum Testen der Implementierung des Signalisierungsprotokolls von ISDN-Einrichtungen, vorgestellt. Die Anwendungsmöglichkeiten dieses Simulators werden ebenfalls aufgezeigt und anhand eines Beispiels erläutert.

2. TEST VON KOMMUNIKATIONSPROTOKOLLEN

Die Notwendigkeit einheitlicher Testmethoden für Kommunikationsprotokolle, die in "offenen" Systemen eingesetzt werden, wurde bereits vor einigen Jahren erkannt. Es werden seit her z.B. bei der ISO Normungsanstrengungen unternommen, die sowohl zu Standards für die Vorgehensweise beim Test von Protokollen [9], als auch zu international einheitlichen Testfallsammlungen für bestimmte Protokolle führen sollen. Auf der Basis dieser Überlegungen wird auch im CCITT an entsprechenden Normen gearbeitet [1]. Im Folgenden sollen nun kurz einige Grundbegriffe aus diesem Gebiet eingeführt werden.

2.1 Grundlagen

Bild 1 zeigt das Modell einer geschichteten Protokollarchitektur. In jeder Schicht, z.B. in der Schicht (N), der Protokollarchitektur existiert in jedem an der Kommunikation teilnehmenden System eine Protokollinstanz (Protocol Entity). Diese Instanzen kommunizieren logisch nach dem sogenannten "Peer-To-Peer Protokoll" miteinander, indem sie schichtspezifische Protokoll dateneinheiten (PDU) austauschen.

Die PDUs werden nicht direkt sondern mit Hilfe der unter der betrachteten Schicht liegenden Schichten (1) bis (N-1) transportiert. Dazu stellen diese an der Schnittstelle zur Schicht (N) einen sogenannten (N-1)-Dienst (Service) zur Verfügung, dessen Benutzung durch das (N-1)-Dienstprotokoll (Service Protocol) festgelegt wird. Bei dieser Kommunikation zwischen zwei benachbarten Schichten werden "Abstrakte Dienstprimitive" (ASP) ausgetauscht. In diesen sind neben Steuerinformationen auch die in "Dienst-Dateneinheiten" (SDU) verpackten Protokoll dateneinheiten enthalten. Die Schicht (N) bietet ihrerseits der Schicht (N+1) einen erweiterten Dienst an.

Will man nun testen, ob eine Implementierung einer (N)-Protokollinstanz sich konform zu der Spezifikation im entsprechenden Standard verhält, dann wird dazu diese Instanz als "Schwarzer Kasten" betrachtet und sein Verhalten an den beiden Schnittstellen beobachtet und beeinflusst. Je nachdem, ob die Schnittstellen direkt (Local Testing) oder indirekt (durch andere Schichten hindurch) zugänglich sind, kann man verschiedene Testmethoden unterscheiden, wie sie z.B. in [9] definiert sind.

Wenn man eine komplette Protokollimplementierung mit mehreren Schichten testen will, dann kann man versuchen alle Schichten zusammen zu testen (Multi Layer Testing). I.a. ist es jedoch einfacher, die Schichten, beginnend mit der Schicht (1), hintereinander zu testen. Beim Test jeder Schicht wird dann vorausgesetzt, daß die darunterliegenden fehlerfrei sind (Embedded Single Layer Testing).

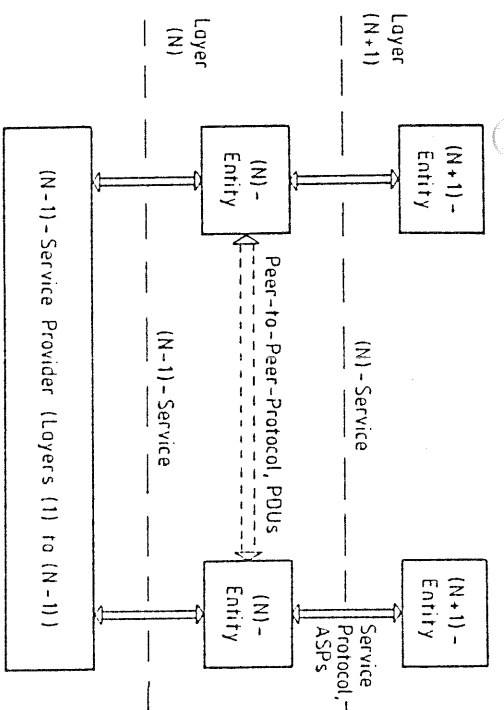


Bild 1 : Modell einer geschichteten Protokollarchitektur

Jenachdem ob z.B. nur die minimalen Grundfunktionen überprüft werden sollen, oder ob das Protokoll vollständig, mit allen Optionen und über den ganzen erlaubten Parameterbereich getestet werden soll werden verschiedene Testkategorien unterschieden [9].

Neben dem Konformitätstest für das Peer-To-Peer Protokoll (Conformance Testing) sollte bei der Bewertung einer Implementierung auch untersucht werden, ob sie die Spezifikationen der Dienstprotokolle erfüllt (Service Testing), wie sie auf Protokollverletzungen reagiert (Robustness Testing) und ob sie den zeitlichen Leistungsanforderungen genügt (Performance Testing).

2.2 Prinzipieller Aufbau eines Testsystems

Da man zwei Schnittstellen überwachen und manipulieren muß besteht ein Testsystem prinzipiell aus zwei Teilen, nämlich aus einem "Upper Tester" und aus einem "Lower Tester". Der Lower Tester, der auf die untere Schnittstelle der Protokollinstanz einwirkt, muß zunächst einen Testtreiber (Test Driver) enthalten, der den Ablauf der Tests steuert. Seine Funktionalität ist der Schicht zuzuordnen, die über der zu testenden Terminal liegt. Der Bereich der Gestaltungsmöglichkeiten für den Testtreiber reicht von einem einfachen Terminal für den manuellen Betrieb bis zu vollautomatischen Treibern, die die Abläufe aufgrund sogenannter "Szenarien" oder aber definiert durch erweiterte endliche Automaten (Extended Finite State Machines) steuern. Ein automatischer Testtreiber ist natürlich besonders für langwierige und sich oft wiederholende Tests von großem Vorteil.

Die von dem Testtreiber erzeugten Primitive müssen noch von einer Instanz innerhalb des Lower Testers in ein gültiges Schicht-(N)-Protokoll umgewandelt werden. Dies kann über eine "Referenzimplementierung" dieses Protokolls erfolgen, die zusätzlich einen Fehlergenerator enthält, damit auch gezielte Protokollverletzungen erzeugt werden können. Falls an

diesem Punkt noch eine weitergehende Flexibilität erforderlich ist, dann kann man auch einen einfachen Kodierer/Dekodierer für (N-1)-ASPs verwenden, was aber bedeutet, daß die Tests bis hinunter auf die Ebene dieser Primitive spezifiziert werden müssen. Dadurch werden die Testspezifikationen entsprechend umfangreicher weil die Funktionalität der Schicht (N) nicht benutzt werden kann.

Der Upper Tester für die obere Schnittstelle wird je nach Testmethode entweder in das zu testende System eingebaut (Distributed Testing) oder auf dessen Benutzerschnittstelle aufgesetzt (Remote Testing). Im letzteren Fall ist der Upper Tester oft ein menschlicher Benutzer, der die Schnittstelle bedient. Soll jedoch nach der verteilten Testmethode getestet werden, dann wird als Upper Tester eine dem Testtreiber des Lower Testers ähnliche Einheit verwendet, bei der besonders darauf geachtet werden muß, daß sie leicht in das Zielsystem integrierbar ist.

Bei heutigen Kommunikationssystemen laufen bestimmte Ereignisse sehr schnell ab. Es kann deshalb nötig sein beide Teile des Testsystems sehr eng zu synchronisieren. Dazu können, wenn sich beide Teile räumlich nah beieinander befinden, eigene Synchronisations- und Steuerleitungen verwendet werden. Andernfalls müssen aufwendigere Methoden eingesetzt werden, so z.B. der Aufbau einer zusätzlichen Verbindung über die unteren Schichten unter Umgehung der Schicht (N) im zu testenden System.

3. DIE TESTFÄLLE FÜR DIE SCHICHT 3

Im Rahmen der Entwicklung des ISDN D-Kanal Simulators wurden Testfallsammlungen für alle drei Schichten des D-Kanal-Protokolls erstellt. Da die Schicht 3 bei weitem die komplexeste dieser Schichten ist, sollen an ihr in diesem Abschnitt einige Aspekte der Testfalldefinition diskutiert werden.

Ziel der Arbeit war es, eine Testfallsammlung zu erstellen, die eine möglichst umfassende Bewertung einer Implementierung auf der Basis der für den Pilotversuch der Deutschen Bundespost maßgebenden FTZ-Richtlinie IR6 [7] erlaubt. Die Testfälle sollten unabhängig von einer speziellen Implementierung und damit im Geltungsbereich der IR6 universell einsetzbar sein. Bei der Komplexität des Protokolls ist ein erschöpfender Test nicht möglich, es mußte also versucht werden, das Testproblem so zu strukturieren, daß mit vertretbarem Aufwand und mit einer möglichst geringen Anzahl von Testfällen eine ausreichende Überdeckung des gesamten Protokollverhaltens erreicht werden konnte.

Die Spezifikation des Protokolls in verbaler Form schloß von Anfang an die Anwendung formaler Methoden zur Erstellung der Testfälle weitgehend aus. Die verbale Protokollspezifikation hat auch die Entscheidung über die Strukturierung der Testfälle entscheidend beeinflußt. Für eine Orientierung der Testfälle an den Zuständen und Übergängen des Protokollautomaten (wie z.B. in [5]) ist eine entsprechende formale Spezifikation als Basis notwendig, darüberhinaus auch noch als zwingende Implementierungsvorschrift festgelegt sein muß. Da dies nicht der Fall war wurde ein funktionsorientierter Ansatz gewählt, bei dem in jedem Testfall eine bestimmte Funktion, z.B. ein ordnungsgemäßer Verbindungsaufbau unter festgelegten Randbedingungen, getestet wird.

3.1 Die Strukturierung der Testfälle

Die Testfälle sollten sowohl den Test von Endgeräten und Nebenstellenanlagen (vermittlungssettig) als auch den Test von Vermittlungsstellen erlauben. Durch die starke Unsymmetrie im Protokoll zwischen Endgeräte- und Vermittlungsseite war es in den meisten Fällen

nicht möglich, die Testfälle für eine Seite einfach zu "spiegeln", um diejenigen für die andere Seite zu erhalten. Deshalb wurden vier Grundkonfigurationen definiert, für die jeweils eigene Testfälle erstellt wurden:

- 1.) Der Tester simuliert eine Vermittlungsstelle (Exchange Termination), es wird eine Endgeräteimplementierung getestet.
- 2.) Der Tester simuliert eines oder mehrere Endgeräte, es wird eine Vermittlungsstelle getestet.
- 3.) Der Tester simuliert eine Vermittlungsstelle, es wird die Vermittlungsseite einer Nebenstellenanlage getestet.
- 4.) Der Tester simuliert die Vermittlungsseite einer Nebenstellenanlage, es wird eine Vermittlungsstelle getestet.

Bei der weiteren Strukturierung wurde die Tatsache ausgenutzt, daß die einzelnen Phasen einer Verbindung (z.B. Aufbau, Gesprächsphase, Abbau) relativ unabhängig voneinander sind. Deshalb kann man die Tests trennen und z.B. für den Test der verschiedenen Fälle beim Verbindungsaufbau eine möglichst einfache Initialisierungsfolge festlegen, mit der der Testling zunächst in den Gesprächszustand gebracht wird. Weiterhin wurde ausgenutzt, daß in den meisten Fällen eine Aufspaltung der Testfälle in die rufende bzw. in die gerufene Seite einer Verbindung möglich ist.

Aus diesen Überlegungen ergab sich die folgende Einteilung der Testfälle für jede der Konfigurationen:

Tests für Grundfunktionen Mit diesen Testfällen wird untersucht, ob die grundlegenden Funktionen, wie z.B. das Empfangen und Dekodieren einer korrekten Meldung oder das Erkennen einer falsch kodierten Meldung, richtig durchgeführt werden. Besteht eine Implementierung diese Anfangstests nicht, dann ist weiteres Testen sinnlos.

Tests für die rufende Seite Hier werden die Funktionen für eine einfache Verbindungssteuerung auf der rufenden Seite getestet. Die Tests umfassen die Varianten für den vollständigen und unvollständigen Verbindungsaufbau sowie für den Verbindungsaufbau.

Tests für die gerufene Seite Diese Tests entsprechen denen auf der rufenden Seite, es wird jedoch zusätzlich der Einfluß mehrerer auf einen Ruf antwortender Endgeräte berücksichtigt.

Tests für die Benutzung der Dienstmerkmale Hierbei werden alle Funktionen getestet, die über die normale, vom Telefon her bekannte Verbindungssteuerung hinausgehen.

Komplexe Tests Diese Tests umfassen die Fälle, in denen eine komplette Verbindung (mit rufender und gerufener Seite) in einem Testfall behandelt werden muß. Diese Art von Testfällen existiert nur für den Test von Vermittlungsstellen. Die Tests werden als komplexe Tests bezeichnet, da sie auch die über das D-Kanal-Protokoll hinausgehenden Funktionen der Vermittlungsstelle benutzen und i.a. auch kompliziertere Testkonfigurationen erfordern.

Es soll noch darauf hingewiesen werden, daß es sinnvoll ist, die Tests in der oben angegebenen Reihenfolge auszuführen, da die höheren Funktionen ein einwandfreies Funktionieren der Grundfunktionen voraussetzen.

3.2 Die Testfallbeschreibungen

Die Beschreibung der Testfälle besteht aus zwei Teilen. In einem ersten, verbalen Teil wird das jeweilige Testziel genau beschrieben. Unter Testziel ist die Funktionalität zu verstehen, die mit diesem speziellen Testfall untersucht werden soll. Es wurde darauf geachtet, daß mit einem Testfall möglichst nur ein einziges Testziel verknüpft ist. Dadurch wird die Interpretation eines erfolglosen Testfalls wesentlich erleichtert.

Neben der Definition des Testziels enthält dieser Teil den Titel und die laufende Nummer des Testfalls, sowie eine Auflistung spezieller Vorkonditionen, die für eine erfolgreiche Ausführung erfüllt sein müssen. Um diese Bedingungen so einfach wie möglich zu halten fängt jeder Test im Ruhezustand der zu testenden Einheit an, d.h. ein Test für den Verbindungsaufbau fängt zunächst mit dem Aufbau der entsprechenden Verbindung an.

Der zweite Teil der Testfallbeschreibung entspricht dem in Bild 2 gezeigten Beispiel. Es wird dabei die Folge der vom Tester zu sendenden und zu empfangenden Meldungen einschließlich der einzuhaltenden Zeitbedingungen dargestellt. Der Aufbau der Meldungen entspricht genau den Spezifikationen in der IR6. Optionale Meldungselemente sind nur angegeben, soweit sie den Testablauf beeinflussen.

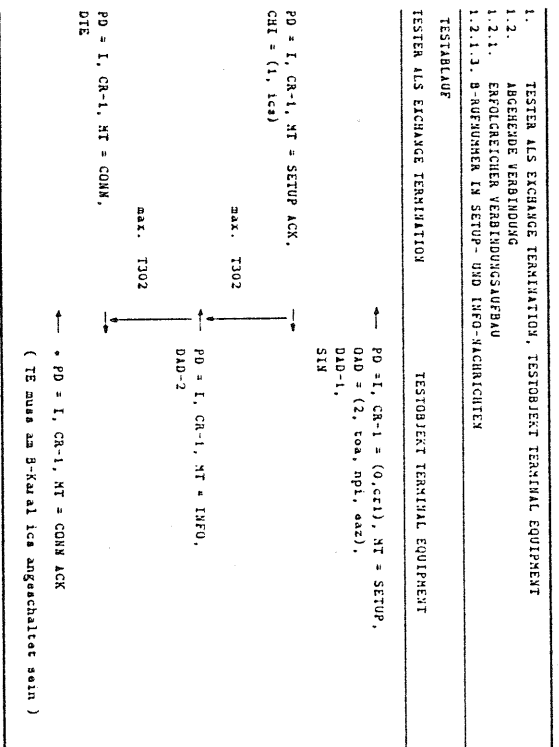


Bild 2 : Beispiel einer Testfallbeschreibung

Bei den Parametern wurden zwei Fälle unterschieden. Zum einen gibt es Parameter, die einen bestimmten Wert annehmen müssen, damit der vorgegebene Ablauf eingehalten wird, so muß z.B. der Protokollkollimator bei allen Meldungen im Beispiel gleich "p" sein, sonst wird der Ablauf mit einer Fehlermeldung abgebrochen. Parameter dieses Typs werden immer mit dem jeweils vorgeschriebenen Wert angegeben. Daneben gibt es Parameter,

deren Werte im zulässigen Bereich frei wählbar sind (z.B. Ursprungs- und Zieladressen von Meldungen). Diese Parameter werden in Form von Variablen dargestellt und müssen bei der Implementierung durch geeignete Werte ersetzt werden.

Die Zeitbedingungen werden durch Pfeile zwischen den beteiligten Meldungen dargestellt und durch die Angabe des entsprechenden Timers aus der Norm spezifiziert. Auf diese Weise werden sowohl Zeitbedingungen spezifiziert, die der Tester beim Senden von Meldungen einzuhalten hat (z.B. CONN), als auch solche, deren Einhaltung er beim Empfang von Meldungen überwachen muß (z.B. INFO).

Eine weitere Besonderheit stellt die "CONN ACK" Meldung dar, die mit einem "*" gekennzeichnet ist. Diese Meldung ist optional, sie muß also vom Testling nicht unbedingt gesendet werden. Der Testfall muß also so implementiert werden, daß ein Fehlen dieser Meldung den Erfolg eines Testfalls nicht beeinflusst.

Auf diese Weise wurden auf der Basis der IR6 etwa 500 Testfälle für Vermittlungsstellen, Endgeräte und Nebenstellenanlagen erstellt. Sie stellen eine Grundlage für die Bewertung von Implementierungen dar und decken einen Großteil der Funktionen der Schicht 3 des D-Kanal-Protokolls ab.

4. DER ISDN D-KANAL SIMULATOR

In diesem Abschnitt soll auf ein spezielles Testsystem, den "ISDN D-Kanal Simulator" näher eingegangen werden. Er bietet die Möglichkeit, abstrakte Testfälle, wie sie oben beschrieben wurden, zu implementieren und auszuführen. Er erlaubt damit die Überprüfung des spezifizierten Verhaltens von ISDN-Einrichtungen an den digitalen Anschlußleitungen für den Basisanschluß [2], den Primärmultiplexanschluß [2,3] sowie für den zweidreihelligen Teilnehmeranschluß U_{P0} [6]. Für den Pilotversuch der Deutschen Bundespost wurde nur die Anschaltung über den Basisanschluß (S_0 Schnittstelle) verwendet, die beiden anderen Anschlussarten stellen Erweiterungen für die Serie dar.

Der Simulator erlaubt die Ausführung von Tests für alle vier in Abschnitt 3.1 definierten Konfigurationen, er kann also insbesondere sowohl die Vermittlungsseite als auch die Endgeräteseite nachbilden.

4.1 Das Systemkonzept des Simulators

Der Simulator stellt dem Benutzer Unterstützung auf drei Ebenen zur Verfügung: Er erlaubt den physikalischen Anschluß an die zu testenden Komponenten, er unterstützt durch ein menugesteuertes "Basis-System" die Erstellung, Archivierung, Ausführung und Auswertung der Tests und er bietet darüber hinaus Sammlungen von vorgefertigten Testfällen, die in sofort ausführbarer Form vorliegen.

Bild 3 zeigt die Komponenten im Einzelnen. In der Schicht 1 werden die international standardisierten Schnittstellen S_0 (entsprechend CCITT I.430 [2]) und S_{2M} (CCITT I.431 [2]) sowie die U_{P0} Schnittstelle gemäß [6] unterstützt. Die entsprechenden Testfallsammlungen erlauben den Test der typischen Funktionen, z.B. Aktivierung und Deaktivierung der Schicht 1 oder die Prüfschleifensteuerung.

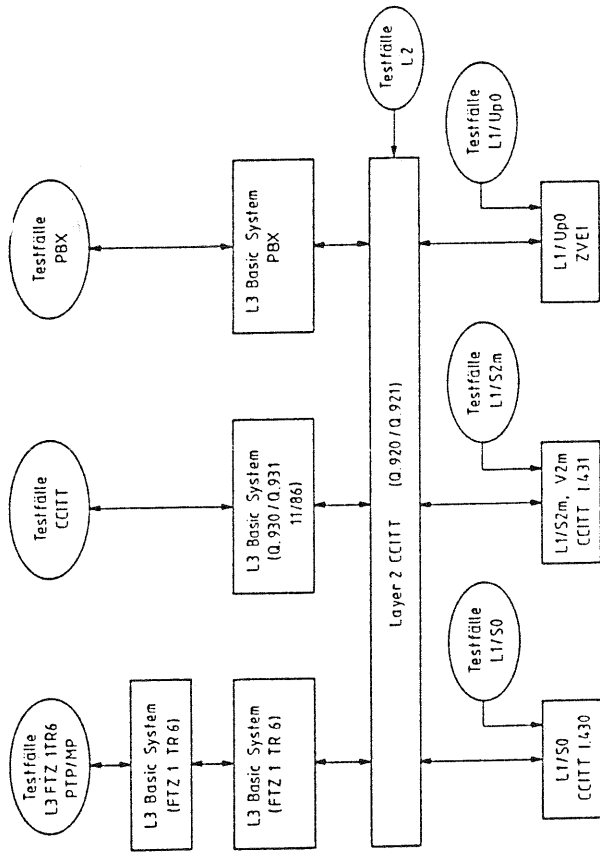


Bild 3 : Das Konzept des D-Kanal Simulators

Die Schicht 2 entspricht CCITT Q.920/Q.921 (Blaubuch [3]). Typische Testfall-Szenarien umfassen hier den Auf- und Abbau von Schicht-2-Verbindungen (LAP D). Eine Besonderheit der Schicht 2 ist die dynamische Vergabe der Schicht-2-Adressen (TEI Management), für die es eigene Testfälle gibt.

Entgegen den international einheitlichen Spezifikationen für die unteren beiden Schichten gibt es für die Schicht 3 bisher noch national unterschiedliche Normen, die auch noch je nach Anwendungsfall (öffentliches oder privates Netz) voneinander abweichen und sich teilweise gegenseitig ausschließen. Deshalb gibt es verschiedene Basissysteme, die jeweils beim Kaltstart des Simulatorsystems ausgewählt werden müssen. Dementsprechend gibt es auch für die einzelnen Fälle spezifische Testfallsammlungen.

Es soll noch bemerkt werden, daß die Basissysteme die Spezifikation der Testfälle in einer Nomenklatur erlauben, die der in den jeweiligen Protokollspezifikationen verwendeten entspricht. Für den Test einer Schicht wird die Funktionalität aller darunterliegenden Schichten automatisch vom Simulator zur Verfügung gestellt. Falls erforderlich können mehrere (max. 8) Schicht-2-Verbindungen gleichzeitig verwaltet werden, wobei auf jeder Schicht-2-Verbindung bis zu 10 verschiedene Schicht-3-Verbindungen simultan aktiv sein können.

In die Testfall-Szenarien der verschiedenen Schichten lassen sich Steuerbefehle für zusätzliche, externe Schnittstellen einbauen, die das Zusammenwirken und die Synchronisation mehrerer, evtl. verschiedener Testeinrichtungen erlauben und damit eine Automatisierung der Testabläufe ermöglichen.

4.2 Die Simulatorkonfiguration

Der Simulationsrechner (Bild 4) besteht aus einem Mikrorechnersystem mit einer 15 MHz CPU und 2 MB Hauptspeicher. Zur Speicherung des Betriebssystems, der Anwenderprogramme und der Test-Datenmodelle (Testfälle) steht eine 71 MB Winchesterplatte zur Verfügung, ein 95 MB Kassettenbandlaufwerk erlaubt die Sicherung und Übertragung der Daten. Zum Kaltstart des Systems und zur Erstellung von Ausdrucken wird ein Druckerterminal verwendet.

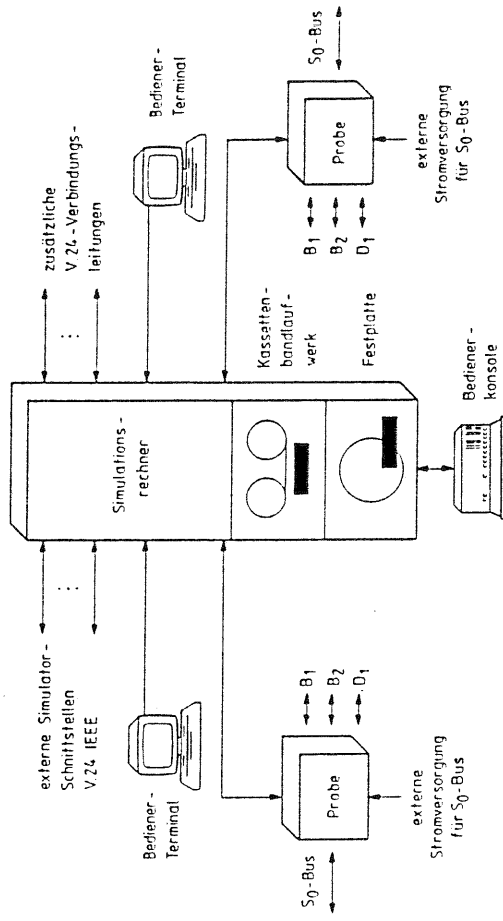


Bild 4 : Die Simulatorkonfiguration

Zusätzlich enthält der Simulator noch ein eigenständiges Ein-/Ausgabesystem, das ebenfalls über eine 15 MHz CPU und einen eigenen Speicher, sowie über einen HDLC-Baustein und 2 DMA-Kanäle (zur Verbindung mit dem Hauptrechner) verfügt.

An den Simulator können über parallele Schnittstellen bis zu zwei voneinander unabhängige "Adapter-Probes" zur Anschaltung an die gewünschten Schnittstellen (S_0 , S_{2M} , U_{P0}) angeschlossen werden, die über separate Bedienerterminals gesteuert werden. Eine wesentliche Funktion der Probe besteht darin, die B- und D-Kanäle zu trennen und zum Anschluß anderer Meßgeräte zugänglich zu machen.

Externe Standardschnittstellen (V.24, IEC-625/IEEE) erlauben die Anschaltung weiterer Testeinrichtungen, eine automatische Steuerung des Simulatorbetriebs oder die Einspielung von Testfall-Dateien.

5. ZUSAMMENFASSUNG UND AUSBLICK

Der in diesem Bericht vorgestellte "ISDN D-Kanal Simulator" bildet zusammen mit dem umfangreichen Angebot an bereits implementierten Testfällen ein wirkungsvolles Hilfsmittel zur Prüfung und Bewertung von ISDN D-Kanal Implementierungen.

Der Simulator und die Testfälle können nicht nur von den Herstellern von ISDN-Komponenten in der Entwicklung und in der Qualitätssicherung eingesetzt werden, sie werden insbesondere auch von der Deutschen Bundespost im Rahmen des ISDN-Pilotversuchs zur Zertifizierung der entsprechenden Geräte verwendet.

Um dem Ziel eines wirklich "offenen" Kommunikationssystems näher zu kommen, müssen in Zukunft nicht nur die verwendeten Protokolle, sondern schrittweise damit auch die Testmethoden und Testabläufe international vereinheitlicht werden. Dadurch wird es dann für Hersteller und Netzbetreiber möglich werden, mit Geräten wie dem hier vorgestellten, die Kompatibilität aller Netzkomponenten sicherzustellen und damit einen sicheren Betrieb des Netzes zu ermöglichen.

LITERATURANGABEN

- [1] CCITT Draft Recommendation X.290, "On Testing and Verification of Data Communication Protocols", July 1987.
- [2] CCITT Red Book, Vol. III, Fascicle III.5, "Integrated Services Digital Network (ISDN), Recommendations of the Series I", Geneva 1985.
- [3] CCITT Recommendations I.440, I.441, Stand 11/1986 (Blue Book).
- [4] CCITT Recommendations I.450, I.451, Stand 11/1986, TD-61-E.
- [5] CEPT Draft Recommendation "On the Terminal Requirements for Attachment to the ISDN Basic Access (NET 3)", May 1987.
- [6] Fachkreis "Übertragungstechnik" im ZVEI, "Bestimmungen für die Schicht 1 der Schnittstelle U_{P0} ", Stand 25.6.1986.
- [7] FTZ Richtlinie IR6 Pilotprojekt, "Kennzeichenaustausch zwischen DIVO(ISDN)-Vermittlungsstellen und ISDN-Teilnehmereinrichtungen - ISDN-D-Kanal-Protokoll -", Ausgabe 2.86.
- [8] FTZ Richtlinie ITR6 Serie, "Kennzeichenaustausch zwischen DIVO(ISDN)-Vermittlungsstellen und ISDN-Teilnehmereinrichtungen - ISDN-D-Kanal-Protokoll -", Ausgabe 2.86.
- [9] ISO/TC97/SC21/WG 16-1, Working draft for "OSI conformance testing methodology and framework", ISO/TC97/SC21 N410, Paris, February 1985. (ISO DP 9646/1 and DP 9646/2 since September 1986).
- [10] Kirchner W., Wagner P., "ISDN D-Channel Protocol Simulator", Electrical Communication, Vol. 61, No. 1, 1987.
- [11] König W., Truong H.L., Waldmann G., "D-Kanal Protokoll im ISDN-Pilotprojekt", NTG-Fachberichte 88, 1985, 112-119.
- [12] Rathgeb E.P., Homann C., Truong H.L., Waldmann G., "Protocol Testing for the ISDN D-Channel Network Layer", 7th Int. Conference on Protocol Specification, Testing and Verification, May 1987.
- [13] Rayner D., "Standardizing Conformance Testing for OSI", 5th Int. Workshop on "Protocol Specification, Testing and Verification", June 1985.