Cellular Networking Perspectives

Editor: David Crowe • Phone +1-403-289-6609 • Email: David.Crowe@cnp-wireless.com Vol. 10, No. 12 December

In This Issue...

Data Snapshot

For a summary of major cellular and PCS data technologies, consult:

www.cnp-wireless.com/data.html

Next Issue: January 3rd, 2002

TIA President Flames ATIS

ATIS and TIA are the two major telecommunications standards-setting organizations in North America. ATIS evolved from ECSA (Exchange Carrier Standards Association), with a focus on backbone networks for telecommunications. TIA evolved from the EIA (now known as the Electronics Industries Alliance), focusing more on end-user equipment. The distinction between their study areas has gradually blurred, and both are now studying telecommunications signaling networks, wireless telecommunications and fiber optics.

Since 1998, ATIS and TIA have been discussing a merger of their standards-setting operations, which would leave the TIA as a pure trade association. In September 2001, the TIA announced it was spinning off its standards-setting activities to a separate organization. Feeling this was an attempt to avoid a merger, on October 25th, 2001, the ATIS board of directors sent a letter directly to the TIA board, calling for the stalled merger talks to proceed.

On November 2nd, 2001, Matt Flanigan, the President of the TIA wrote back to Susan Miller, his counterpart at ATIS, stating he was “disappointed and offended” by this direct action. He stated he did not believe that the latest ATIS proposal was a “merger of equals” – it seemed more like a “takeover”.

He noted that the TIA is many times larger than ATIS (although its standards-setting operation is similar in size), it has been around for many more years (the TIA has only been around since 1988, but the EIA existed long before TIA) and it has had one standards committee operating since 1944.

It seems inevitable that North American telecommunications standards-setting will some day be performed by a single organization. But, given the hard feelings of today, it may not be soon.

Circumnavigating SS7, Part III: TCAP in Context

TCAP is commonly incorporated in application protocols using SS7 for transport, as described in our November 2001 issue.

TCAP is not a perfect solution, however. In this issue, we identify the problems TCAP is designed to solve, and we analyze how well this fits the needs of SS7 applications such as ANSI-41 and GSM MAP. We will also briefly discuss the ASN.1 specification definition language and its relationship to TCAP.

Motivation for TLV Encoding

There are a number of basic requirements for telecommunications network applications protocols, including:

- Interoperability with older and newer revision levels of the protocol. Telecommunications networks are rarely upgraded as a whole.
- The ability to add new elements to a protocol – messages or parameters.

Editor:	David Crowe.
Accounts:	Evelyn Goreham.
Marketing:	Muneerah Vasanji.
Distribution:	Debbie Brandelli.
Production:	Doug Scofield.

Cellular Networking Perspectives (issn 1195-3233) is published monthly by Cellular Networking Perspectives Ltd., 2636 Toronto Crescent NW, Calgary AB, T2N 3W1, Canada. **Phone:** 1-800-633-5514 (+1-403-274-4749) **Fax:** +1-403-289-6658 **Email:** cnpsales@cnp-wireless.com **Web:** www.cnp-wireless.com **Subscriptions:** CDN$350 in Canada (incl. GST), US$350 in the USA and US$400 elsewhere. Payment by cheque, bank transfer, American Express, Diners Club, MasterCard or Visa. **Delivery**: Email or 1st class mail. **Back Issues:** Single issues are $40 in the US and Canada and $45 elsewhere, or in bulk at reduced rates. **Discounts**: Educational and small businesses: 25% off any order. **Copies**: Each subscriber is licensed to make up to 10 copies of each issue or back issue. Please call for rates to allow more copies.

- Optional elements in protocol messages.
- Multiple simultaneous operations, requiring the association of each message with a current transaction by its recipient.

The TCAP Tag, Length, Value (TLV) encoding is a general solution for such protocol requirements:

- Elements, such as messages and parameters, can be added to the protocol, because each can be uniquely identified by a tag.
- The number of elements is effectively unconstrained, because an enormous number of unique tags can be defined.
- Unknown elements (e.g. generated using a higher revision level) can be ignored.
- Elements can vary in length, because the length is always included.
- The maximum length of element values is very large, because lengths can occupy 1 octet (for lengths up to 127) or several octets.
- The TCAP transaction package contains at least one transaction identifier to associate messages.

These capabilities come at a price, and TCAP also includes features that are not required, but that do add complexity and overhead to protocols.

Tag Troubles

The Tag in TCAP is a number that uniquely identifies an element of a protocol, such as a parameter.

The scope of a TCAP tag varies from universal (e.g. for TCAP protocol elements) to application specific (e.g. unique within an application) or context-specific (only unique within the context of a specific operation).

This scope concept is over-generalized, as only the basic TCAP protocol elements really need to be unique outside an application (and they do not really need tags). Within an application, operation codes need to be unique throughout the application, but parameter identifiers only need to be unique within a single operation (i.e. 'context specific').

There is very little practical benefit to having parameter tags which, for example, have the same meaning in different operations, and there is even less benefit to having them standardized between applications. Each operation within an application protocol can re-use the same identifiers without fear of confusion, because of the unique identification of the protocol and operation within each transaction. There is, in fact, a substantial cost to identifiers with a wide scope, as they tend to run to 3 octets instead of one (1) for virtually all 'context specific' identifiers.

Long Identifiers (Tags) in ANSI-41

This flexibility in identifier scope allows for very inefficient implementations, perhaps best illustrated by the ANSI-41 parameter identification. The protocol insists that every one of the more than 300 parameter tags be unique, meaning that the majority are 3 octets long. Ironically, ANSI-41 incorrectly encodes its parameter identifiers as 'context-specific'. If they were correctly implemented, each ANSI-41 identifier would only be one octet long.

Tags for Mandatory Parameters

TCAP demands that a tag be included for every parameter. Clearly, however, this is not necessary for mandatory parameters, nor is it necessary for the TCAP headers. It just constitutes additional overhead, particularly for protocols using multi-octet tags, such as ANSI-41.

Length Limitations

The length in TCAP is included for every element, because of strict adherence to the T,L,V format. Just as mandatory tags do not really need a tag to be explicitly included, fixed length parameters do not really need their length included. The length is also extraneous for parts of a TCAP message extending to the end (the package, the component sequence and, usually, the single component and its parameter list). This information does not usually need to be supplied at all, because the total length of a TCAP message is usually available from the encapsulating protocol layer.

Transactions: Yes! Components: No!

TCAP provides a transaction layer that is critical to the operation of protocols such as ANSI-41. The transaction identifier associates messages that belong together, allowing multiple simultaneous operations, without confusion.

For example, using distinct transaction identifiers, an MSC can initiate many RegistrationNotification messages to a variety of HLRs. Responses can be received in any order, and yet they can still be associated with the specific mobile that triggered the operation, by the transaction identifier that is 'reflected' from the initiating message.

The component layer is, by contrast, of debatable value. It is a double layer – a list of components, each of which acts like a message within a message.

In actuality, most TCAP protocols implement the 'list' of components as a single component, as there is no benefit to having multiple components.

Even the distinction between the package (transaction) type (Query, Conversation, Response etc.) and the component type (INVOKE, RETURN RESULT etc.) of the single component being included is unnecessary. Package types are adequate to define whether a message initiates a new transaction (Query), continues an existing transaction (Conversation), ends a transaction (Response) or exists apart from a transaction (Unidirectional). Although the transaction layer did not originally define an error package (Abort), a Response package with an identifiable error parameter would be more than adequate to indicate an abnormal end to a transaction.

Value Vagaries

TCAP does not specify the encoding of values if they are 'primitive'. Only those defined as 'constructor' parameters have an internal TCAP structure. The use of constructors is, in practice, discouraged by their complexity and overhead, however. For example, a parameter composed of three one-octet sub-parameters could be implemented as:

- A single parameter with three octets of data and 2 octets of overhead (assuming 1 octet parameter tags).
  Total size: 5 octets.
- Three parameters, each with one octet of data and 2 octets of overhead.
  Total size: 9 octets.
- One structured parameter (2 octets overhead), containing 3 sub-parameters (2 octets of overhead each).
  Total size 11 octets.

The overhead of TCAP for small bit fields is so great (3 octets=24 bits) they are rarely implemented as separate parameters. Usually, multiple bits are organized into a single parameter to reduce the overhead.

Often, more complex multi-octet parameters are also organized in an *ad hoc* fashion within a primitive parameter. For example, the Calling Geodetic Location parameter used in many location-services standards is implemented as a primitive parameter within TCAP – using, however, an internal, quite complex structure outside of TCAP, presumably to avoid the overhead that would have come with up to nine (9) separate data elements.

Ordering of Parameters

TCAP allows lists (e.g. of parameters) to be specified as ordered (Sequence) or unordered (Set). There is no value in this capability. Ordered lists are just slightly more complex to create, and they are a lot simpler to analyze. For example, parsing software can immediately determine if a parameter is missing when a parameter later in the list is encountered, and software can be organized in a linear fashion.

Transmitting Syntax and Instructions

One of the fundamental flaws of TCAP, which is not inherent in its TLV structure, is a confusion between Syntax, Instructions and Data – the three fundamentals of a protocol.

Only the data should be transmitted in a protocol. The syntax, as well as the instructions – for generating a message for transmission and for processing received data – must be known to both the transmitter and receiver. Including information about syntax and instructions in the transmissions of a protocol is redundant, adding no real value.

There are several examples of this conceptual flaw in TCAP:

- The syntax of a protocol can identify which parameters are simple (primitive) and which are constructors. By reserving a bit in every tag for this, TCAP reduces the number of one-octet identifiers from 62 to 30.
- The instructions for a protocol can identify which lists are sets and which are sequences. Or, it can just define all lists as sequences to simplify parsing. There is no reason to transmit this information as TCAP does.
- The distinction between Query and Conversation packages “with” or “without” permission (to terminate the transaction) can also be embedded in the instructions for a protocol. This does not need to be transmitted.
- The syntax can define the tag of mandatory elements and the length of fixed length elements. There is no need to transmit this information.

What is the Overhead?

The overhead of TCAP is quite substantial, particularly considering the 252 octet limit of SS7 messages (which includes the SCCP layer). The overhead varies substantially, based on how enthusiastically designers incorporate TCAP concepts (the more enthusiastically, the greater the overhead).

The following example illustrates how great the overhead in a TCAP message can be, using the following assumptions:

- The message is initiating a transaction
- It contains 20 one-octet parameters
- 10 are mandatory and 10 are optional
- 10 are fixed in length and 10 are variable
- 5 use one-octet tags, 5 use two-octet tags and 10 use three-octet tags

Using TCAP to implement this message would result in a TCAP header of 21 octets and a parameter list of 85 octets, for a total of 106 octets.

Removing all the TCAP overhead, as described above, would result in a header of 7 octets (1 initial octet with flags, a 4-octet transaction ID and a 2-octet operation code) and a parameter list of 40 octets (one-octet tags omitted when the parameter is mandatory and with lengths omitted when the parameter is fixed in length), for a total of only 47 octets.

Consequently, it can be estimated that at least half of every TCAP message represents overhead that could be removed without difficulty.

Alternatives to TCAP

One alternative to TCAP is to design every protocol message by hand, instead of relying upon a ‘one size fits all’ structure. Bits can be laid out so that a message can be encoded compactly, yet still be unambiguously interpreted by a human or machine with knowledge of the syntax and instructions for the protocol. For example, the inclusion of optional components can be identified by a single bit instead of a full octet, or even by specific values of other protocol elements. This is the approach taken by most lower level protocols, including SS7 MTP and IP. The major drawback to this approach lies in the skill needed to perform a design of this type. The most significant danger is that the protocol will work well today, but will prove to be impossible to enhance in a compatible fashion for the future (e.g. not enough bits to identify optional extensions). Both SS7 and IP suffer from exactly this problem.

At the other end of the spectrum, XML is being used to define many internet application protocols. It is an HTML-like meta-language (language to define other languages). Defining a protocol in XML requires designers to list the tags for the data elements (unlike HTML, which has already defined a set of tags). Instead of the TLV format of TCAP, XML uses ASCII names enclosed in angle brackets (e.g. <digits>8006335514</digits>).

The overhead of XML is considerable. Even assuming two character tags (very conservative, as most readable tags will be much longer), because of the overhead of these tags and the need to encode everything in ASCII, it is likely that the 20 parameter message described above would be around 300 octets in XML – two to three times greater than TCAP, and five to ten times greater than a hand built protocol.

XML is, however, very easy to define, quite extensible, and easy to understand. It is quite applicable for applications where bandwidth is not a significant constraint.

ASN.1 and BER

Some protocols, such as GSM MAP (3GPP TS 29.002), use ASN.1 (Abstract Syntax Notation 1) and BER (Basic Encoding Rules) to define their TCAP protocol.

ASN.1 is a meta-language (like XML). Instead of using HTML as a model, ASN.1 is based upon BNF (Backus Naur Form), which was first used to define the Algol computer programming language in the 1960's. Most modern computer languages are still defined using a variant of BNF. An example of a modern form of BNF is the IETF ABNF (Augmented BNF) defined at:

www.ietf.org/rfc/rfc2234

ASN.1 - Abstract Syntax Notation

ASN.1 is based on statements like:

```
imsi ::= OCTET STRING
            (SIZE (3 .. 8 ) )
min ::= OCTET STRING
            (SIZE (5) )
```

which defines 'imsi' as a 3-8 octet data element and 'min' as a 5 octet data element. These could be used as parameters inside messages or other parameters, through statements such as:

```
message ::= SEQUENCE {
   param1 [1] imsi,
   param2 [2] min OPTIONAL
}
```

which defines a message containing a mandatory parameter of type 'imsi', followed by an optional parameter of type 'min'.

BNF is very effective at defining computer languages, with definitions that are clear, compact and easy to read. However, it does not seem to work so well for protocols. There are a number of possible reasons for this:

- BNF is not used to define data structures in programming languages. It just defines the tools required to do this.
- Protocol specifications become unwieldy. The GSM ASN.1 specification is over 100 pages long.
- ASN.1 is very difficult to read. The reader may have to skip over many pages to find a referenced data element. GSM attempts to reduce this problem by including a cross reference which, at 200 pages, is longer than the ASN.1 specification. GSM also includes a fully expanded form of ASN.1 so that the definition of parameters is directly within the definition of operations.

Perhaps the biggest conceptual flaw with the use of ASN.1 to define a protocol is because of a seductive – but flawed – analogy with the definition of data structures in programming languages. When writing a computer program, it is not important to know whether a 'short integer' is 16, 32 or 64 bits, as long as you can rely on it being at least 16 bits. If different compilers allocate different amounts of storage based on computer word sizes and alignment constraints, the program will still work, although it may not be optimal.

With protocols, on the other hand, the specific layout of the bits is critically important. Designers of software to transmit receiver or monitor protocol messages need to have total assurance that they understand the meaning of every bit. If the transmitter sends 8 bits when the receiver expects 16, chaos will reign.

The flaw in the analogy is that a computer program generally runs within a single computer, whereas a protocol is always used between at least two different computers – computers that may have different word sizes and alignment constraints for data.

For those who must understand ASN.1 – and perhaps do not want to read the daunting ITU-T specifications, X.680-X.683 – an English translation of the book *ASN.1 Communication between Heterogeneous Systems* is freely available online at:

www.oss.com/asn1/dubuisson.html

BER - Basic Encoding Rules

Basic Encoding Rules (BER; ITU-T X.690) define how an ASN.1 specification is translated into a string of octets. BER is compatible with TCAP, as its encodings are based on the TLV concept. However, not every TCAP protocol (ANSI-41 for example) is compatible with BER.

Most complex ASN.1 implementations will be translated by a special parser. Its output can be used to help software generate BER-compatible output, and it can also interpret received messages, verifying that they conform to the syntax.

The ability to use a parser is sometimes touted as an advantage of ASN.1. On the other hand, protocols defined without ASN.1 are often simple enough to be understood by human beings.

PER - Packed Encoding Rules

An answer to the criticism that the BER/TCAP encoding is inefficient is the Packed Encoding Rules (PER). These rules, defined in ITU-T X.691, perform a number of optimizations:

- Tags are eliminated for mandatory parameters, being replaced by a bitmap for optional parameters.
- Length octets are only included for variable length parameters.
- Octet alignment is not maintained if the *unaligned* variant is used. This allows for the efficient transmission of fields, although it can make analysis of the resulting bitstream very difficult to analyze, particularly by humans.

PER has not yet achieved a great amount of use. One of the reasons may be that even the slightest incompatibility between implementations could cause a one-bit error, which could turn the rest of a transmission effectively into gibberish. With BER, by contrast, it is likely that an encoding error would only affect one parameter.

Conclusions

TCAP is an important protocol for many modern telecommunications application protocols, including both major Mobile Application Parts – ANSI-41 and GSM. The basic protocol is not without flaws, and standards built upon it often introduce their own idiosyncrasies. However, it can satisfy the goals of interoperability, extensibility and compatibility between protocol revisions. Its inefficiencies are manageable for signaling protocols which generally contain messages that are high in value and modest in size.

Status of IS-41 Rev. C & TIA/EIA-41-D (ANSI-41) Implementations

Cellular Networking Perspectives

Editor: David.Crowe@cnp-wireless.com

Last published July, 2001

	Vendor and Radio Technology									
Intersystem Operations Capability	Alcatel	Ericsson			LG	Lucent			Motorola	
	CDMA	Analog	CDMA	TDMA	CDMA	Analog	CDMA	TDMA	Analog	CDMA
Authentication (CAVE)	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
IS-778 Authentication Enhancements			🕓		🕓					
CNAP/CNAR			🧪	✔	✔		🧪	🧪		✔
CNIP/CNIR	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Data (IS-737)	✔	▨	✔	✔	✔	▨			▨	✔
Inter-MSC handoff: Analog to…	▨	✔	▨	✔	▨	✔	▨	✔	✔	▨
Inter-MSC handoff: CDMA to…	✔		✔	▨	✔	✔	✔	▨	✔	✔
Inter-MSC handoff: TDMA to…	▨	✔	▨	✔	▨	✔	▨	✔	✔	▨
International (IS-751 IMSI and IS-807)		▨	✔	✔	🧪	▨	🧪	🧪	▨	
Hyperband handoff (TSB-76)	✔	▨		✔		▨	✔	✔	▨	✔
LNP Phase I (IS-756)	✔	✔	✔	✔		✔	✔	✔	✔	✔
LNP Phase II (IS-756-A)		✔	✔	✔					🕓	🕓
MWN	✔	✔	✔	✔	✔		✔	✔	✔	✔
Origination Triggers	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Over-the-air Activation (IS-725)	✔	▨	✔	🧪	✔	▨	✔	🕓	▨	✔
SMS Origination	✔	▨	✔	✔	✔	▨	🧪	✔	▨	✔
SMS Termination	✔		✔	✔	✔		✔	✔	✔	✔
Termination Triggers	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Voice Privacy - basic	✔	▨		✔	✔	▨	✔	✔	▨	
Voice Privacy - EPE	▨	▨	▨		▨	▨	▨		▨	▨
WIN Phase I (IS-771)	🕓	✔	✔	✔	🧪	🕓	🕓	🕓	🕓	🕓
WIN Phase II (Prepaid)	🕓		🧪		🕓				🕓	🕓

Status of IS-41 Rev. C & TIA/EIA-41-D (ANSI-41) Implementations

Cellular Networking Perspectives

Editor: David.Crowe@cnp-wireless.com

Last published July, 2001

	Vendor and Radio Technology							
Intersystem Operations Capability	NEC		Nortel (MSC/BS)			Telos		
	Analog	CDMA	Analog	CDMA	TDMA	Analog	CDMA	TDMA
Authentication (CAVE)		✔	✔	✔	✔	✔	✔	✔
IS-778 Authentication Enhancements								
CNAP/CNAR			⏲	⏲	⏲			
CNIP/CNIR	✔	✔	✔	✔	✔	✔	✔	✔
Data (IS-737)	▨	✔	▨	⏲	⏲	▨	**⏲**	
Inter-MSC handoff: Analog to...	✔	▨	✔	▨	✔	✔	▨	
Inter-MSC handoff: CDMA to...	✔	✔	✔	✔	▨	✔	✔	▨
Inter-MSC handoff: TDMA to...		▨	✔	▨	✔	✔	▨	✔
International (IS-751 IMSI and IS-807)	▨		▨	⏲	⏲	▨	**⏲**	
Hyperband handoff (TSB-76)	▨		▨	✔	✔	▨		
LNP Phase I (IS-756)			✔	✔	✔	✔	✔	✔
LNP Phase II (IS-756-A)			🧪	🧪	🧪	**3Q'01**	**3Q'01**	**3Q'01**
MWN	✔	✔	✔	✔	✔		✔	✔
Origination Triggers	✔	✔	✔	✔	✔	✔	✔	✔
Over-the-air Activation (IS-725)	▨		▨	✔	✔	▨	✔	✔
SMS Origination	▨	✔	▨	✔	✔	▨	✔	✔
SMS Termination		✔	✔	✔	✔		✔	✔
Termination Triggers	✔	✔	✔	✔	✔	✔	✔	✔
Voice Privacy - basic	▨	✔	▨			▨		
Voice Privacy - EPE	▨	▨	▨	▨	⏲	▨	▨	
WIN Phase I (IS-771)			✔	✔	✔	✔	✔	✔
WIN Phase II (Prepaid)			⏲	⏲	⏲			**⏲**

Terms & Acronyms

www.cnp-wireless.com/glossary.html.

Symbols

✔	In field trial or commercial service.
XQ'XX	Specifies the quarter during which commercial availability is expected (e.g. 4Q'01).
🧪	In lab trial.
⏲	Under Development
▨	Shading indicates a capability that is not technically feasible at present, or for which no standard yet exists.
Bold type	Company names in **bold type** have indicated a change in status since the last report.
Red	Text and figures in red indicate specific changes since the last report (visible only in electronic edition of newsletter).

TIA TR-45.6 and TSG-P 2G and 3G Wireless Packet Data Standards

Cellular Networking Perspectives

Editor: David Crowe • Phone +1-403-289-6609 • Email David.Crowe@cnp-wireless.com

Last published June, 2001

Note:
1. IS- Interim Standard, TSB- Telecommunications Systems Bulletin, P.Sxxxx - 3GPP2 TSG-P Specification, P.Rxxxx - TSG-P Report, PN- Project Number, SP- ANSI Standards Proposal.
2. Bold Type indicates a modification since the previous publication of this information.
3. Published TIA standards can be obtained from the TIA at www.tiaonline.org/standards

Thanks to Ed Campbell (3Com) and Raymond Hsu (Qualcomm) for their assistance compiling the information in this table.

CDPD - Cellular Digital Packet Data

Standard	Project	Description	Status
IS-732	PN-4033	Cellular Digital Packet Data (CDPD) - multiple parts	Published 02/98
TSB87	PN-4001...	CDPD support services (Directory, Authentication, DNS, Testing, Identifiers, Numbering)	Published 02/98

CDPD - Cellular Digital Packet Data (Revised)

Standard	Project	Description	Status
TIA/EIA-732	**SP-4033-UG**	**Revisions to CDPD and Upgrade to ANSI**	**Published 08/01**

3G Packet Data

Standard	Project	Description	Status
IS-835	PN-4732	cdma2000 Wireless IP Network Standard	Published 12/00
IS-835-1	**PN-4732-1**	**Addendum to IS-835**	**Replaced by IS-835-A**
IS-835-A	**PN-3-4732-RV1**	**cdma2000 Wireless IP Network Sandard**	**Published 05/01**
IS-835-B		**Supports IPv6, Dynamic Home Agent, QoS and Push Services**	**Development**
TSB115	PN-4286	cdma2000 Wireless IP Architecture based on IETF Protocols	Published 12/00

3GPP2 TSG-P Projects

3GPP2	Description	Status
P.R0001	Wireless IP Network Architecture based on IETF Protocols	Published 07/00
P.S0001	Wireless IP Network Standard based on IETF protocols (same as IS-835)	Published 12/99
P.S0001-A	**Wireless IP Network Standard (same technical content as IS-835)**	**Published 07/00**
P.S0001-A-1	**Addendum to P.S0001-A**	**Published 12/00**
P.S0001-Av3	**Wireless IP Network Standard (same technical content as IS-835-A)**	**Published 07/01**
P.S0001-B	**Wireless IP Network Standard (in V&V)**	**Development**