

IMPLEMENTASI ENKAPSULASI PADA VHF DATA LINK (VDL) MODE 2 BERBASIS SOFTWARE GNURADIO

Cecep Sujana, Iyan Turyana, A A N Ananda Kusuma
Pusat Teknologi Informasi dan Komunikasi (PTIK)
Badan Pengkajian dan Penerapan Teknologi (BPPT)
Lt 21, Gd II BPPT, Jl MH Thamrin 8, Jakarta, Indonesia
email : {cecep.sujana, iyan.turyana, ngurah.ananda}@bppt.go.id

Abstrak

Salah satu program prioritas BPPT (Badan Pengkajian dan Penerapan Teknologi) di unit PTIK (Pusat Teknologi Informasi dan Komunikasi) adalah penerapan CNS/ATM (Communications Navigation Surveillance / Air Traffic Management) di Indonesia. Program ini adalah untuk modernisasi pengaturan lalu lintas udara. Pada aspek komunikasi telah tersedia standar VDL (VHF Data Link). Dengan metoda baru ini pertukaran informasi dari ATC (Air Traffic Management) ke pesawat yang semula menggunakan komunikasi suara diubah menjadi komunikasi data.

BPPT saat ini sedang mencoba membuat prototype peralatan komunikasi yang menggunakan standar VDL Mode 2 pada layer 2 (layer datalink) dan ATN (Aeronautical Telecommunication Network) pada layer 3. Perangkat keras yang digunakan pada proyek ini adalah SDR (Software Defined Radio) merek USRP (Universal Software Radio Peripheral), sedangkan perangkat lunak yang digunakan ialah gnuradio yang merupakan software opensource berbasis linux.

Enkapsulasi yang dilakukan di layer datalink pada VDL Mode 2 dinamakan AVLC (Aviation VHF Link Control) yang berbasis pada standar HDLC (High level Data Link Control) ISO/IEC 13239. Implementasi enkapsulasi ini dilakukan dengan menggunakan bahasa python.

Kata Kunci— CNS/ATM, VDL, GNUradio , AVLC, HDLC.

I. PENDAHULUAN

Penggunaan new CNS/ATM berbasis satelit dan teknologi digital telah disepakati oleh negara – negara anggota International Civil Aviation Organization (ICAO) pada 10th Air Navigation Conference 1991 di Montreal, Canada. Teknologi ini diperlukan untuk mengantisipasi peningkatan jumlah penerbangan di seluruh dunia agar tetap aman, efficient, seamless, global dan environmental friendly sesuai dengan asas penerbangan sipil

Indonesia harus dengan segera mengejar ketinggalannya dalam implementasi teknologi CNS/ATM tersebut, untuk itulah BPPT – sebagaimana salah satu tugas utamanya – membantu mengaplikasikan teknologi tersebut berkerjasama dengan dinas perhubungan udara kementerian perhubungan dan juga PT Angkasa Pura sebagai pengelola bandara di Indonesia.

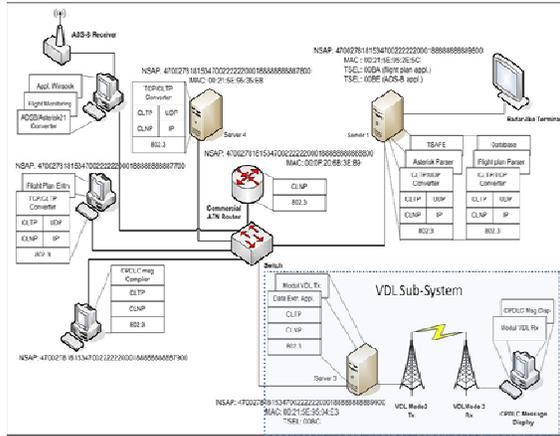
Karena menyangkut keselamatan penerbangan, implementasi teknologi CNS/ATM tidak bisa

dilakukan hanya dengan prosedur normal pembelian, instalasi, training, komisioning. Implementasi teknologi harus didahului dengan testing dan trial yang dilaporkan ke ICAO, dari berbagai option teknologi yang diperbolehkan untuk memilih salah satu yang dianggap paling cocok dengan iklim dan kondisi geografis Indonesia dan compatible dengan peralatan avionics di pesawat.

Untuk keperluan penguasaan teknologi ini PTIK-BPPT merancang suatu model prototype CNS/ATM yang disebut Stand-Alone Mobile Air-routes Radar Terminal (SMART)-Build sebagaimana ditunjukkan pada Gambar 1 [1]. Pada saat ini bagian yang sudah pada tahap implementasi ialah bagian Surveillance yaitu teknologi ADS-B (Automatic Dependent Surveillance-Broadcast). Sedangkan bagian Communication yaitu VDL masih pada tahap penelitian dan pembuatan prototype.

VDL Mode 2 digunakan untuk komunikasi antara ATC dengan pesawat. Standard ini - digabungkan dengan CPDLC (Controller Pilot Data Link Communication) - bertujuan untuk menggantikan

komunikasi suara antara ATC (Air Traffic Control) dengan pesawat dengan komunikasi berbasis text terstandar. Dengan demikian diharapkan tidak ada lagi miss-communication yang bisa menyebabkan kecelakaan.



Gambar. 1. Model uji coba SMART Build CNS/ATM

Untuk pembuatan prototype alat komunikasi ini BPPT menggunakan SDR (Software Defined Radio) yaitu suatu perangkat komunikasi yang fleksibel karena pada perangkat ini fungsi-fungsi yang biasanya menggunakan hardware (modulasi, filtering, dll) diubah menjadi menggunakan software [2]. SDR yang dipilih adalah merek USRP (Universal Software Radio Peripheral) [3] dengan software-nya Gnu Radio [4].

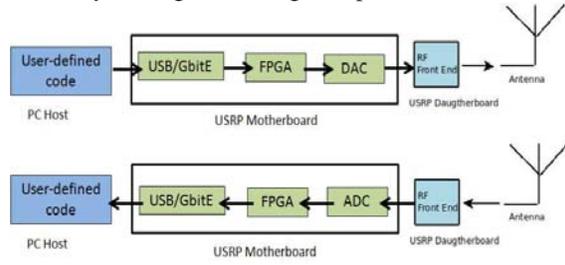
Gnuradio adalah suatu software yang berbasis FOSS (Free Open Source Software) yang ditulis dengan menggunakan 3 bahasa, yaitu :

- VHDL (Very high IC Hardware Definition Language) dan Verilog. Kedua software ini digunakan untuk programming FPGA (Field Programming Gate Array) yang merupakan otak dari USRP
- Bahasa C++, digunakan untuk membuat modul-modul yang berfungsi khusus (critical module)
- Python, digunakan untuk menggabungkan module-module yang dibuat dengan menggunakan bahasa C. Bahasa ini dirancang untuk semudah mungkin digunakan oleh end-user.

Bahasa VHDL dan Verilog dijalankan pada FPGA sedangkan bahasa C dan Python dijalankan pada PC yang terhubung dengan USRP. Gambar 2 menunjukkan diagram blok komunikasi USRP.

Untuk mengimplemetasikan standard VDL mode 2 pada SMART Build diperlukan modifikasi atau penambahan dari program python dan C yang sudah ada di Gnuradio. Pada Gambar 1 terlihat bahwa standard protokol VDL berada di layer 2 dan layer 1 sedangkan pada layer 3 digunakan protokol ATN

(Aeronautical Telecommunication Network). Tulisan ini fokus pada sebagian implementasi standard VDL Mode 2 yaitu bagian framing/enkapsulasi.



Gambar. 2. Diagram blok untuk jalur transmit-recv antar USRP

II. AVLC

Enkapsulasi (pembuatan frame) pada VDL Mode 2 menggunakan standard ISO/IEC 13239 [5]. Pada standard tersebut frame yang dibuat dinamakan HDLC (High-Level Data Link Control). Tetapi pada standard tersebut disebutkan pula bahwa fame bisa diubah sesuai kebutuhan, tentunya dengan batasan-batasan tertentu. Frame HDLC yang sudah disesuaikan untuk VDL Mode 2 dinamakan AVLC (Aviation VHF Link Control). Standard AVLC ini ditulis pada Doc 9776 “Manual on VHF Digital Link (VDL) Mode 2” yang dikeluarkan oleh ICAO [6]. Format dari AVLC bisa dilihat pada Gambar 3.

| DESCRIPTION | OCTET No | BIT NUMBER | | | | | | | | | |
|----------------------|---------------------|------------|----|----|-----|----|----|----|-----|---|--|
| | | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | | |
| Flag | | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | |
| DLS Address | Destination Address | 1 | 22 | 23 | 24 | 25 | 26 | 27 | A/G | 0 | |
| | | 2 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 0 | |
| | | 3 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 0 | |
| | | 4 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | |
| | Source Address | 5 | 22 | 23 | 24 | 25 | 26 | 27 | C/R | 0 | |
| | | 6 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 0 | |
| | | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 0 | |
| | | 8 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 1 | |
| Link Control Field | 9 | | | | P/F | | | | | | |
| Information | ... | | | | | | | | | | |
| | N-3 | | | | | | | | | | |
| Frame Check Sequence | N-2 | | | | | | | | | | |
| | N-1 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | | |
| Flag | N | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | |
| | | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | |

Gambar. 3. AVLC Frame format

❖ Flag

- Setiap frame diawali dan diakhiri oleh flag
- Nilai flag adalah 0x7E (01111110). Nilai ini hanya ada di awal dan di akhir frame tidak boleh ada di dalam frame.
- Bila ada data di dalam frame yang sama nilainya dengan flag maka harus ditambahkan “bit stuffing”. Dengan demikian tidak akan terjadi kelasalan pembacaan flag.

- Bit stuffing adalah penambahan bit '0' bila ada 5 bit '1' berturut-turut.

- Contoh :

Pola asli : 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0
 Stelah bit stuffing : 1 1 1 1 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 0 1 1 1 1 0 1 1 0 1 1 0

❖ DLS Address

DLS address terdiri atas 8 oktet dengan perincian 4 oktet untuk destination address dan 4 oktet untuk source address. Setiap oktet diawali (bit nomor 1) dengan bit '0', kecuali oktet terakhir diawali dengan bit '1' (untuk menunjukkan akhir dari DLS address). Dengan demikian destination dan source address masing-masing terdiri atas 28 bit dengan rincian :

- Bit ke 1 s.d. 24 adalah address, dimana bit ke 1 adalah MSB dan bit ke 24 adalah LSB
- Bit ke 25 s.d. 27 adalah address type
- Bit ke 28 bernama A/G pada destination address dan C/R pada source address
- A/G adalah air-ground indicator
- '0' = udara
- '1' = ground (termasuk pesawat yang sedang di darat)
- C/R adalah "command/response" indicator.
- '0' = command frame
- '1' = response frame
- Address type yang isinya berdasarkan tabel 1 berikut :

TABLE I
ADDRESS TYPE

| Bit encoding | | | Type | Comment |
|--------------|----|----|-----------------------|---------------------------------|
| 25 | 26 | 27 | | |
| 0 | 0 | 0 | Reserved | Future use |
| 0 | 0 | 1 | Ground station | ICAO-administered address space |
| 0 | 1 | 0 | Reserved | Future use |
| 0 | 1 | 1 | Reserved | Future use |
| 1 | 0 | 0 | Aircraft | 24-bit ICAO address |
| 1 | 0 | 1 | Ground station | ICAO-delegated address space |
| 1 | 1 | 0 | Reserved | Future use |
| 1 | 1 | 1 | All station broadcast | All station |

❖ Link Control Field

Isi dari Link Control Field bergantung pada format frame-nya. Menurut standard HDLC ada 3 jenis format frame, yaitu :

- I format
Digunakan untuk pertukaran informasi
- S format
Digunakan untuk pengawasan data-link seperti acknowledge I frame, permintaan untuk pengiriman kembali I frame, dan permintaan pengiriman I frame yang tertunda. Setiap S format harus mempunyai nomor urut N(R) yang

mungkin saja menjadi acknowledgement I frame di stasion penerima

- U format
Digunakan untuk control tambahan transmisi data-link. Frame ini tidak mempunyai nomor urut tapi mempunyai P/F. Bit modifier (M) digunakan untuk menentukan tipe frame menjadi 32 fungsi command dan 32 fungsi response.

Tabel 2 berikut memperlihatkan isi dari Link Control Field berdasarkan format frame-nya. AVLC menggunakan link control field modulo 8 dari HDLC.

TABLE II
ISI LINK CONTROL FIELD (MODULO 8)

| Control Field Format | Control Field bit | | | | | | | |
|-------------------------------------|-------------------|------|---|---|-----|------|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Information Transfer C/R (I format) | 0 | N(S) | | | P/F | N(R) | | |
| Supervisory C/R (S format) | 1 | 0 | S | S | P/F | N(R) | | |
| Unnumbered C/R (U format) | 1 | 1 | M | M | P/F | M | M | M |

- N(S) : nomor urut pengiriman (bit2=LSB)
- N(R) : nomor urut penerimaan (bit6=LSB)
- S : bit untuk fungsi supervisory
- M : bit untuk modifikasi fungsi
- P/F : poll bit / Final bit

Nilai-nilai bit di atas bergantung pada tipe frame yang dikirimkan. AVLC tidak menggunakan semua tipe frame pada HDLC. Tabel 3 adalah daftar frame yang digunakan pada AVLC.

TABLE III
FRAME YANG DIGUNAKAN DI AVLC

| Command | Response |
|-------------------------|---------------------------------|
| INFO [Information] | INFO |
| RR [Receive Ready] | RR |
| XID [Exchange Identity] | XID |
| TEST | TEST |
| SREJ [Selectice Reject] | SREJ |
| FRMR [Frame Reject] | |
| UI [Unnumbered Info] | UA [Unnumbered Acknowledgement] |
| DISC [Disconnect] | DM [Disconnected Mode] |

Tabel 4 menunjukkan nilai untuk bit link control field pada Tabel 2

TABLE IV
NILAI BIT TIAP TIPE FRAME

| Tipe Frame | Format | Nilai S pada Format S |
|------------|--------|-----------------------|
| | | Nilai M pada Format U |
| INFO | I | |
| RR | S | 00 |
| XID | U | 11-101 |
| TEST | U | 00-111 |
| SREJ | S | 11 |
| FRMR | U | 10-001 |
| UI | U | 00-000 |
| UA | U | 00-110 |
| DISC | U | 00-010 |
| DM | U | 11-000 |

❖ Information Field

Isi dari Information field terdiri atas

- Format Identifier subfield (1 octet)
- Data link layer subfield
Terdiri atas Group Identifier, Group length, dan Parameter Field
- User data subfield
Terdiri atas User data identifier dan User data field.

Dengan melihat tipe frame bisa dilihat fungsi dari information field, misalnya :

- Pada frame tipe I information field berisi informasi yang akan dikirimkan.
- Pada frame tipe SREJ information field berisi frame-frame yang di-reject
- Pada XID information field berisi informasi untuk pertukaran data.

❖ Frame Check Sequence

Frame check sequence pada AVLC menggunakan metoda Reed Solomon 16 bit.

III. IMPLEMENTASI ENKAPSULASI

Karena kajian physical layer VDL dilakukan dengan memodifikasi script benchmark_tx.py dan benchmark_rx.py, maka kajian implementasi AVLC diawali dengan script yang sama. Terdapat beberapa tingkatan enkapsulasi yang disediakan oleh python script pada Gnu Radio. Untuk memanipulasi suatu paket data dapat dilakukan dengan modul struct yang disediakan oleh python. Sebagai contoh, untuk menyusun suatu paket data yang nantinya akan menjadi data payload pada suatu AVLC frame dapat digunakan struct.pack(), dan hal serupa dapat dilakukan di sisi penerima dengan menggunakan struct.unpack() untuk membongkar kembali isi paket.

Fungsi struct.pack() telah menyediakan opsi untuk byte order yang sesuai, misalnya contoh di bawah untuk network byte order dan menyertakan informasi nomor paket:

```
payload = struct.pack('!H',pktno & 0xffff) +
data
```

Untuk menyertakan VDL frame, maka idealnya dapat dilakukan enkapsulasi frame sebelum paket dikirim oleh fungsi send_pkt() dan membongkar frame saat paket diterima di fungsi recv_pkt(). Contoh-contoh yang disediakan pada benchmark_tx.py dan benchmark_rx.py langsung mengirimkan/menerima data payload ke blok modulator yang melakukan pengiriman/penerimaan paket, yaitu fungsi mod_pkts() atau demod_pkts() yang bertugas untuk melakukan konversi paket ke atau dari complex modulated baseband signal, yang didasarkan atas jumlah samples per symbol, jumlah bits per symbol, access code (synchronization vector), dan sebagainya. Contoh:

```
pkt= packet_utils.make_packet(
    payload,
    self._modulator.samples_per_symbol(),
    self._modulator.bits_per_symbol(),
    ...
```

Dari paparan di atas, terlihat bahwa AVLC dapat dilakukan dengan langsung memodifikasi frame melalui manipulasi paket dengan struct.pack(), untuk kemudian langsung dilewatkan oleh send_pkt() ke mod_pkts(), atau dibuatkan fungsi pembuatan frame tersendiri atas paket-paket payload yang akan dikirim oleh send_pkt(). Tahapan implementasi ini masih dilakukan kajian, dan tahapan pemrograman dan pengujian masih berjalan terutama dalam kaitan integrasi dengan layer-layer yang lain.

IV. PENUTUP

Makalah ini telah menjelaskan suatu skema enkapsulasi AVLC frame untuk transmisi data pada sistem VDL mode 2. Hasil kajian dari format AVLC frame menunjukkan suatu kelayakan implementasi enkapsulasi ini pada contoh-contoh modul yang disediakan oleh Gnu Radio dalam mengaplikasikan suatu software-defined radio untuk VDL mode 2 menggunakan perangkat USRP. Walaupun secara konsep, tahapan implementasi sudah cukup jelas, namun masih diperlukan pemrograman lanjutan dan pengujian untuk memastikan enkapsulasi berjalan dengan baik, dan dapat terintegrasi dengan fungsi-fungsi komunikasi lainnya.

V. UCAPAN TERIMA KASIH

Penelitian ini didanai oleh Program Insentif Peningkatan Kemampuan Peneliti dan Perekayasa 2011, Kementerian Negara Riset dan Teknologi

DAFTAR PUSTAKA

- [1] PTIK-BPPT, *Program Manual Kegiatan Sistem Informasi untuk Transportasi Udara (CNS/ATM)*, 2010
- [2] A A N Ananda Kusuma, C. Sujana, I. Turyana, *Software-Defined Radio Untuk Pengembangan Testbed VHF Data Link (VDL)*, dikirim ke Teknoin, 2011
- [3] Ettus Research, The USRP: www.ettus.com
- [4] Gnu Radio: www.gnuradio.org
- [5] ISO/IEC, *Information technology – Telecommunications and Information exchange between system – High-level Data Link Control (HDLC) procedures*, 2002
- [6] ICAO, *Manual on VHF Digital Link (VDL) Mode 2*, 2001
- [7] Wideman, Deitel, *PYTHON How to Program*, 2001