IMPLEMENTASI ATN TUNNELING PADA TESTBED VHF DATALINK (VDL) BERBASIS SOFTWARE-DEFINED RADIO

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

Abstraksi

CNS/ATM (Communications Navigation Surveillance / Air Traffic Management) adalah program pengaturan lalu lintas udara yang sedang dikaji di PTIK (Pusat Teknologi Informasi dan Komunikasi), yang mana merupakan salah satu program prioritas BPPT (Badan Pengkajian dan Penerapan Teknologi). Pada aspek komunikasi telah tersedia standard VDL (VHF Data Link) pada layer 2 dan ATN (Aeronautical Telecommunication Network) pada layer 3 yang mengacu pada OSI model. Dengan standard ini pertukaran informasi dari ATC (Air Traffic Controller) ke pesawat yang semula menggunakan komunikasi suara diubah menjadi komunikasi data.

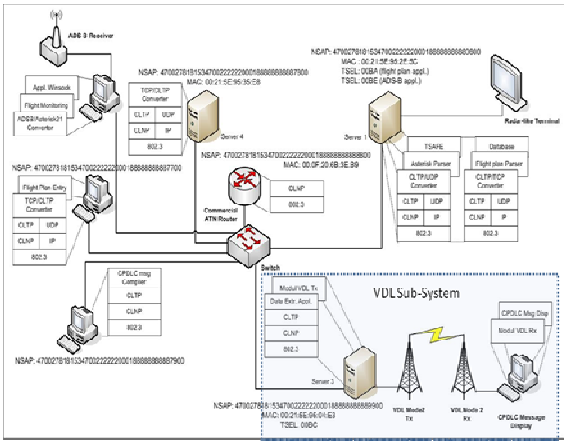
Salah satu penelitian yang sedang dilakukan adalah pembuatan suatu program jaringan menggunakan konsep tunneling untuk melewatkan datagram ATN pada testbed VDL yang dikembangkan menggunakan teknologi software-defined radio. Untuk keperluan tersebut dibutuhkan suatu virtual network device yang dibentuk oleh tun/tap driver. Telah banyak yang mengimplementasikan konsep tunneling dengan tun/tap driver, tetapi selama ini masih terbatas pada protokol TCP/IP. Makalah ini membahas pengembangan sebuah prototipe program jaringan yang mengimplementasikan ATN tunneling pada jalur komunikasi VDL-Mode 2 yang menggunakan perangkat keras Universal Software Radio Peripheral (USRP) dan perangkat lunak GNU radio.

Kata kunci : CNS/ATM, VDL-Mode 2, ATN, Tunneling, TUN/TAP driver

I. PENDAHULUAN

Peningkatan keselamatan penerbangan selalu menjadi perhatian penting pengguna, pemerintah, pengelola, dan industri. Selain pengaruh sumber daya manusia dan faktor alam, peningkatan keselamatan didapatkan melalui kemajuan teknologi-teknologi yang terkait dengan penerbangan, yang mana memotivasi munculnya teknologi CNS/ATM. Teknologi ini menekankan pada integrasi dari suatu manajemen transportasi udara dengan fungsi-fungsi komunikasi, navigasi, dan surveilance pada sistem penerbangan. Dengan demikian optimalisasi penjadwalan, kendali waktu, dan separasi trafik dapat tercapai, yang pada akhirnya meningkatkan keselamatan dan pengelolaan transportasi udara.

Keperluan akan penguasaan teknologi CNS/ATM ini ditindaklanjuti oleh PTIK-BPPT dengan merancang suatu model prototype CNS/ATM yang disebut Stand-Alone Mobile Air-routes Radar Terminal (SMART)-Build [9] ,sebagaimana ditunjukkan pada Gambar 1.



Gambar 1. Model uji coba SMART Build CNS/ATM: VDL Mode 2 sub-system.

Aspek komunikasi pada testbed ini diwakili oleh keberadaan protokol komunikasi ATN, dan suatu sub-system VDL yang memodelkan suatu sistem komunikasi

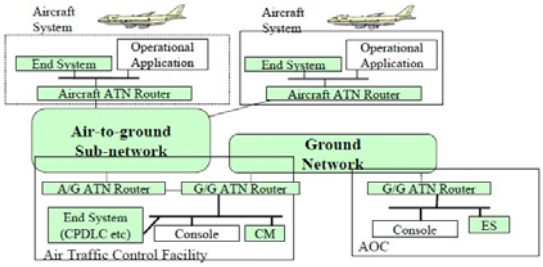
multiple-access air-to-ground dan ground-to-air. Sistem VDL ini nantinya dapat menggantikan komunikasi suara dengan komunikasi data, misalnya untuk melewatkan pesan-pesan dari aplikasi CPDLC (Controller Pilot Data Link Communication) [3][5][10]. Dari beberapa proposal VDL yang ada, VDL mode 2 adalah standar versi utama dan yang paling banyak dirujuk untuk diimplementasikan. Untuk itu, kegiatan penelitian ini hanya difokuskan pada **VDL mode 2**.

Suatu platform Software-Defined Radio (SDR) menggunakan perangkat-perangkat USRP [1] dan software Gnu Radio [2] telah dikembangkan untuk implementasi sub-system VDL Mode 2 pada SMART-Build [8]. Secara umum VDL dapat direalisasikan secara fleksibel dengan SDR, namun diperlukan berbagai modul-modul tambahan sehingga sistem VDL dengan SDR tersebut dapat terintegrasi dengan suatu jaringan komunikasi yang menggunakan protokol ATN. Sebagai contoh, VDL mode 2 yang didefinisikan pada physical layer dan datalink layer nantinya diharapkan terhubung dengan protokol CLNP yang merupakan salah satu protokol ATN di network layer. Hal ini memberikan keunikan pada penelitian ini karena umumnya sistem SDR berbasis USRP dan Gnu Radio diintegrasikan ke suatu jaringan yang menggunakan protokol TCP/IP.

Pada makalah ini akan dipaparkan gambaran umum protokol ATN dan protipe SDR yang telah dikembangkan, konsep pemrograman kernel module untuk mendukung protokol ATN, konsep tunneling menggunakan TUN/TAP, dan kontribusi penelitian dalam hal integrasi suatu sistem SDR ke jaringan berbasis ATN menggunakan modifikasi dari TUN/TAP driver yang tersedia pada sistem operasi Linux.

II. Konsep Umum Jaringan Berbasis Protokol ATN

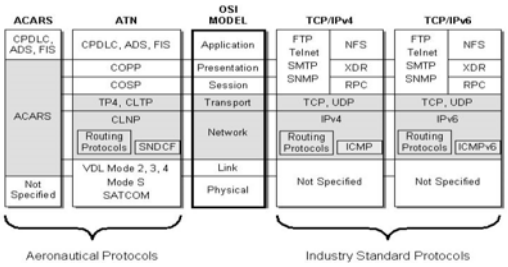
ATN (Aeronautical Telecommunication Network) adalah infrastruktur komunikasi global yang mengatur transfer data digital antara pesawat terbang dan fasilitas pengendalian penerbangan. ATN dibuat secara khusus untuk melayani berbagai layanan komunikasi aeronautika yang melibatkan pemerintah, industri penerbangan, penyedia jasa maupun pengguna layanan lalu lintas udara. ATN dirancang untuk dapat memenuhi keandalan dan ketersediaan (*availability*) yang tinggi dengan memastikan bahwa tidak ada satu kesalahan pun berdasarkan kriteria yang ditentukan dan memungkinkan pemilihan rute komunikasi data secara dinamis. Secara umum, pemanfaatan ATN diperlihatkan oleh Gambar 2, di mana ATN terdiri atas aplikasi dan layanan komunikasi yang memungkinkan komunikasi *ground-to-ground router* dan *air-to-ground router*. ATN tersebut harus memenuhi kriteria standar yang merujuk pada dokumen teknis *9705-AN/956* dan berbasis pada OSI yang merujuk dokumen *ISO/IEC 7498*.



Gambar 2. Gambaran Umum Sistem Komunikasi dengan Protokol ATN [4].

Protokol ATN dispesifikasikan menggunakan model OSI (Open Systems Interconnection). Gambar 3 memberikan komparasi perbedaan lapisan protokol yang digunakan oleh ATN dengan standar industri seperti TCP/IP dan proprietary di dunia penerbangan, misalnya ACARS. Terlihat bahwa ATN mengimplementasikan model OSI yang lebih ketat. Pada physical layer yang dikombinasikan dengan data link layer sebagai satu kesatuan, ATN menyediakan VDL mode 2,3,4, Mode S dan Satcom. Pada network layer, ATN menyediakan CLNP, sedangkan pada transport layer ATN menyediakan CLTP dan COTP. Pada session layer, ATN mengimplementasikan COSP (Connection Oriented Session Protocol), suatu protokol yang mengelola session service dan sinkronisasi data yang dipertukarkan antar koneksi. Pada presentation layer, ATN mengimplementasikan COPP (Connection Oriented Presentation Protocol). Pada application layer, ATN menyediakan berbagai aplikasi terkait penerbangan, diantaranya: CPDLC (Controller Pilot-Data Link Communications) untuk komunikasi data antara pilot dan air traffic controller, ADSB (Automatic Dependence Surveilance Broadcast) agar trafik penerbangan dapat dilihat dengan presisi tinggi, FIS (Flight Information Systems) suatu sistem penampilan data-data penerbangan.

Sebagai catatan, ATN memiliki beberapa kelebihan dibandingkan TCP/IP, misalnya dari sisi address space yang lebih besar di mana ATN menggunakan 20-22 octets atau 160-176 bits dibandingkan dengan IPv4 dan IPv6 yang masing-masing menggunakan 32 bits dan 128 bits. Selain itu, ATN juga memiliki manajemen congestion yang lebih baik.

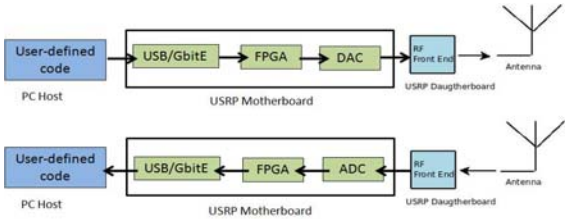


Gambar 3. Komparasi Lapisan Protokol ATN dengan Protokol Komunikasi Standar Lainnya.

Pengembangan perangkat yang mendukung ATN di Indonesia dapat dikatakan relatif baru, yaitu sejak tahun 2007 saat penelitian tentang ATN dimulai dilakukan oleh BPPT bekerjasama dengan SGU (Swiss German University), yang kemudian dilanjutkan dengan melibatkan UI (Universitas Indonesia) pada tahun 2009 [6][7]. Beberapa hasil penelitian diantaranya pembuatan modul pengiriman dan penerimaan CLTP mengacu pada *ISO/IEC 8602:1995*. Selanjutnya, modul tersebut diperbaiki dengan mengintegrasikan CLTP dan CLNP. Perbaikan modul pengiriman dan penerimaan tersebut mengacu pada *ISO/IEC 8602:1995* dan *ISO/IEC 8473:1994(E)*. Hasil penelitian lainnya adalah mengenai *fragmentasi* dan *reassembly*, dan berhasil memperbaiki kemampuan CLNP yang merujuk pada *ISO 8473: 1994(E)*.

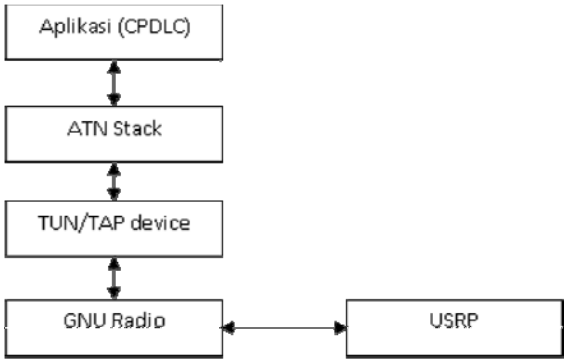
III. SDR Menggunakan USRP dan Gnu Radio

Sistem SDR untuk mengimplementasikan sub-sistem VDL seperti pada Gambar 1, memerlukan suatu perangkat keras dengan dukungan modul-modul pemrograman yang memadai. Filosofi yang diadopsi untuk kegiatan penelitian ini adalah *open source* (bersifat terbuka). Dari beberapa solusi SDR berbasis *open-source* yang tersedia, solusi yang menggunakan perangkat keras USRP [1] dan perangkat lunak Gnu Radio [2] yang dipilih. Informasi tahapan-tahapan pengembangan sistem SDR untuk VDL mode 2 ini dijelaskan di [8].



Gambar 4. Diagram blok untuk jalur transmit-receive antar USRP.

Tahapan pemrosesan sinyal pada suatu jalur transmit dan receive dari suatu sistem SDR ditunjukkan pada Gambar 4. Terlihat bahwa secara umum sistem tersebut terbagi menjadi 3 blok. Di bagian kanan adalah blok RF frontend, yang terdiri atas daughterboard dan antena. Blok ini bertindak sebagai antarmuka ke domain analog RF. Blok yang di tengah adalah USRP board yang berfungsi mempersiapkan digital samples untuk antarmuka antara domain analog dan pemrosesan sinyal digital pada PC host. Pada USRP, sinyal analog di-sampled dan didegradasikan ke baseband pada FPGA, untuk kemudian dikirim ke PC host melalui usb atau gigabit Ethernet, untuk selanjutnya diolah dengan fungsi-fungsi pemrosesan sinyal seperti modulasi, demodulasi, coding, dan lain-lain dengan bantuan perangkat lunak Gnu Radio.



Gambar 5. Hubungan Gnu Radio dan Modul-Modul lainnya pada PC Host.

Dari paparan di atas, terlihat bahwa teknik-teknik pemrosesan yang biasanya dilakukan secara hardware dapat dipindahkan ke skema software, sehingga diharapkan akan lebih fleksible, serta menghemat waktu dan biaya untuk pengembangan. Dari sisi pemrograman sistem, terjadi perubahan sudut pandang di mana fungsi-fungsi yang semestinya diprogram pada hardware atau

kernel OS, sekarang diprogram sebagai aplikasi pada user space. Gambar 5 menggambarkan sekilas hubungan antara Gnu Radio dengan beberapa aplikasi lainnya, dan yang terpenting adalah bagaimana Gnu Radio dapat berkomunikasi dengan protocol stack ATN.

IV. ATN Kernel Module

Membuat modul kernel (*kernel module*) tidak sama dengan membuat program di *user space*, karena kernel module terletak di dalam *kernel space*. *Kernel space* adalah ruang pengalamatan di dalam kernel Linux, sebagai inti dari sistem operasi yang menyediakan *services* bagi aplikasi pengguna. Sedangkan *user space* adalah ruang pengalamatan di dalam memori tempat program aplikasi berada. Perbedaan kernel module di *kernel space* dengan program aplikasi di *user space* adalah pada titik awal ketika program dieksekusi. Kernel module dieksekusi mulai dari *module_init()* sedangkan program aplikasi diawali dengan fungsi *main()*.

Implementasi protokol CLNP, CLTP dan ES-IS dimulai dengan membuat kernel module ATN di dalam kernel Linux [6][7]. Kernel module ATN yang dibuat menggunakan kernel Linux 2.6.x.x. Kernel module ATN memiliki dua fungsi utama yaitu *module_init(init_atn)* dan *module_exit (cleanup_atn)* yang terletak di dalam file *af_atn.c*. Fungsi *module_init(init_atn)* ini bertindak sebagai fungsi utama di dalam pembuatan kernel module ATN. Fungsi *module_init(init_atn)* berisi beberapa fungsi inisialisasi dan registrasi fungsi-fungsi handler di dalam kernel Linux. Fungsi handler ini dapat dibagi menjadi dua bagian yaitu fungsi handler yang menangani penerimaan paket data dan fungsi handler yang menangani bagian pengiriman paket data. Sedangkan *module_exit(cleanup_atn)* berisi fungsi untuk menghapus registrasi fungsi-fungsi handler di dalam kernel Linux.

V. Konsep Tunneling Pada ATN

Protokol CLNP sebagai salah satu protokol ATN yang berada di Network Layer akan dihubungkan dengan VDL Mode 2 yang berada di Datalink Layer. Konsep yang digunakan adalah membuat program jaringan menggunakan konsep tunneling untuk melewatkan datagram ATN pada testbed VDL yang dikembangkan menggunakan teknologi software-defined radio. Untuk keperluan tersebut dibutuhkan suatu virtual network device yang dibentuk oleh tun/tap driver. Telah banyak yang mengimplementasikan konsep tunneling dengan tun/tap driver, tetapi selama ini masih terbatas pada protokol TCP/IP. Hubungan antara modul-modul terkait dengan integrasi komponen software-defined radio dengan jaringan ATN ditunjukkan pada Gambar 5.

Salah satu bentuk komunikasi yang digunakan oleh pesawat dengan fasilitas pengendali udara (ATC) adalah pertukaran data melalui jaringan telekomunikasi udara (ATN). Fungsi komunikasi pada VDL Mode 2 yang melalui ATN mendukung komunikasi data digital. Agar komunikasi data digital antara pesawat dengan fasilitas pengendali udara (ATC) aman maka dapat dibuat suatu tunnel agar terbentuk suatu jalur komunikasi data yang *secure*, merupakan aspek tambahan yang perlu dikaji dalam implementasi ATN tunneling.

TUN/TAP Driver

TUP/TAP adalah program driver yang melayani pengiriman dan penerimaan paket untuk program aplikasi [11][12]. TUN/TAP berfungsi pada jaringan point to point dan ethernet, menerima paket dari media fisik, menerima paket dari program aplikasi dan mengirim paket melalui media fisik dan menyampaikan paket ke program aplikasi.

Untuk menggunakan program driver TUN/TAP adalah dengan menggunakan file device yang berada di */dev/net/tun*, driver tun akan berhubungan dengan network device dan kernel. Network device akan muncul sebagai tunXX atau tapXX.

Implementasi

Implementasi konsep tunneling pada protokol ATN adalah dengan membuat program aplikasi jaringan sederhana yang memungkinkan program ini dapat berkomunikasi dengan module ATN di dalam Kernel Linux dan device tun/tap yang telah dibuat.

Program dibuat menggunakan bahasa pemogramman C pada Sistem Operasi Ubuntu 10.04 dengan menggunakan compiler gcc versi 4.4.3. Mekanisme pada program adalah mengalokasikan beberapa memori pada sistem lalu membuka file device tun yang berada di path */dev/net/tun* , kemudian membuat mekanisme pemogramman socket networking dengan memanfaatkan API socket pada Library C di GNU/Linux. Hal terpenting dari program ini adalah pada pembentukan socket descriptor dimana syscall socket memanfaatkan komunikasi domain PF_ATN dan protokol ATN didalamnya. Kemudian memanfaatkan beberapa syscall yang sudah didukung oleh module ATN

seperti syscall *sendto()* untuk pengiriman paket data dan *recvfrom()* untuk penerimaan paket data. Hal ini bisa di lihat pada potongan code dibawah ini.

```
/**************************************
tun_alloc: allocates or reconnects to a
tun/tap device. The caller must reserve
enough        space        in        *dev.
**************************************/

int tun_alloc(char *dev, int flags) {

struct ifreq ifr;

int fd, err;

char *clonedev = "/dev/net/tun";

if((fd = open(clonedev, O_RDWR)) < 0) {

perror("Opening /dev/net/tun");

return fd;

}

memset(&ifr, 0, sizeof(ifr));

ifr.ifr_flags = flags;

if(*dev) {

strncpy(ifr.ifr_name, dev, IFNAMSIZ);

}

if((err   =   ioctl(fd,   TUNSETIFF,   (void
*)&ifr)) < 0) {

perror("ioctl(TUNSETIFF)");

close(fd);

return err;

}

strcpy(dev, ifr.ifr_name);

return fd;

}
```

Pada pembuatan socket ATN untuk mendukung keluarga protokol ATN (PF_ATN) di dalam Kernel Linux dapat dilihat pada code dibawah ini

```
/* get a socket descriptor */

if((sd = socket(AF_ATN, SOCK_DGRAM, 0)) <
0)     {

perror("CLTP server - socket() error");

exit(-1);

}

else

printf("CLTP server - socket() is OK\n");
```

Hasil-hasil pengujian

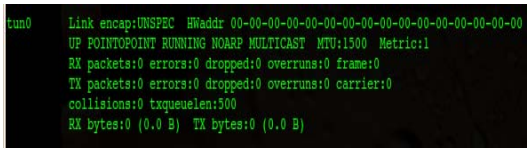
Pengujian dilakukan dengan melihat kompatibilitas atau kecocokan antara device driver tun/tap yang ada dengan module ATN di dalam Kernel Linux itu sendiri.

Beberapa hasil pengujian diantaranya adalah :

Gambar 6. Device tun/tap tanpa module ATN

Pada gambar 6, dapat dilihat bahwa program server belum mendukung komunikasi data ATN tetapi sudah terkoneksi dengan device tun/tap. Sebelum menggunakan device tun/tap untuk ATN, diperlukan teknik untuk memasang module ATN ke dalam Kernel Linux. Dengan perintah *insmod atn.ko* dalam mode administrator, maka module ATN akan terintegrasi ke dalam Kernel Linux. Dengan menjalankan kembali program server maka device tun/tap sudah terkoneksi dengan system dan sudah terintegrasi dengan module ATN. Hal ini dapat dilihat pada gambar 7 dan gambar 8 dibawah ini.

Gambar 7. Device tun/tap dengan module ATN.



Gambar 8. Device tun0 yang berhasil dibuat.

VI. Integrasi ATN pada Gnu Radio

Protokol CLNP sebagai salah satu protokol ATN yang berada di Network Layer akan dihubungkan dengan VDL Mode 2 yang berada di Datalink Layer. Konsep yang digunakan adalah membuat program jaringan menggunakan konsep tunneling untuk melewatkan datagram ATN pada testbed VDL berbasis software-defined radio yang dikembangkan menggunakan USRP

dan Gnu Radio. Untuk keperluan tersebut dibutuhkan suatu virtual network device yang dibentuk oleh tun/tap driver. Telah banyak yang mengimplementasikan konsep tunneling dengan tun/tap driver, tetapi selama ini masih terbatas pada protokol TCP/IP.

Karena Gnu Radio pada dasarnya suatu aplikasi yang bekerja di user-space, maka untuk mengintegrasikan SDR berbasis USRP/Gnu Radio ke suatu jaringan, diperlukan suatu mekanisme supaya paket-paket yang diterima ataupun dikirim ke suatu media fisik jaringan dapat dilewatkan terlebih dahulu ke user-space. Tambahan context-swicth kernel ke user space ini menyertakan tambahan overhead yang dapat mempengaruhi kinerja, namun untuk tahap awal target kegiatan ini adalah mengintegrasikan SDR berbasis USRP/Gnu Radio ke SMART-Build. Peningkatan kinerja akan dilakukan sebagai riset lanjutan.

Dalam modul-modul Gnu Radio, disertakan contoh penggunaan tun/tap driver untuk integrasikan ke jaringan menggunakan IP protocol. Dengan menggunakan script tunnel.py, tun/tap driver berhasil membuat dan mengaktifkan virtual interface gr0. Virtual interface ini bisa dianggap seolah-olah suatu network device yang mana paket-paket yang diterima dibawa ke user-space program, dan tidak ke media fisik. Tugas selanjutnya adalah memodifikasi tun/tap driver module supaya dapat mengenal dan melewatkan (tunneling) ATN datagram.

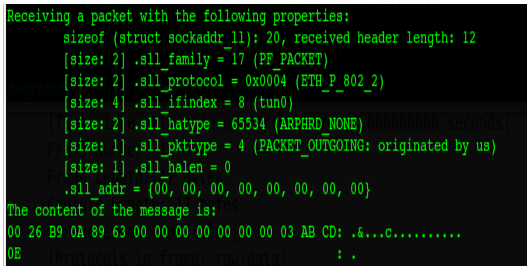
Pengujian

Pengujian dilakukan dengan mengaktifkan script tunnel.py di dua host yang sudah terinstall modul-modul GNU Radio dan module ATN sampai virtual device dikenal seperti ditunjukan pada gambar 9. Pertama kali kami melakukan pengujian dengan memanfaatkan protokol TCP/IP di dalam GNU Radio yang sudah terhubung dengan USRP melalui media USB. Tunnel.py bisa mengenali dan membentuk device tun/tap dengan nama gr0.

Hal yang serupa dilakukan untuk ATN, di mana device tun/tap berhasil dibuat dan diintegrasikan dengan module ATN. Namun sebagai catatan, walaupun virtual device tun0 untuk ATN berhasil dibuat namun dalam pengujian belum tercapai target pengiriman data melalui virtual interface tersebut, yang mana masih memerlukan penyesuaian dengan modul-modul datalink dan physical layer yang digunakan untuk VDL mode 2.

Module ATN yang sudah terintegrasi ke dalam Kernel Linux sudah mendukung beberapa protokol ATN diantaranya ES-IS, CLNP dan CLTP, dan dari ketiga protokol ini ada satu buah protokol yang bekerja dengan membroadcast paket data di layer datalink seperti protokol ARP. Karena module ATN dan device tun/tap belum terintegrasi dengan sempurna maka kami mencoba menganalisa proses pengiriman paket data melalui device tun0.

Pada gambar 9 dapat dilihat bahwa device tun0 bisa digunakan dengan baik dengan protokol family PF_PACKET tetapi belum sempurna dengan PF_ATN.



Gambar 9. Analisa device tun0 yang digunakan pada pengiriman paket data.

VII. Penutup

Makalah ini telah menggambarkan skema integrasi suatu sistem VDL Mode 2 berbasis perangkat USRP dan software Gnu Radio ke suatu testbed jaringan yang menggunakan protokol ATN. Hasil kajian menunjukkan bahwa diperlukan suatu teknik ATN tunneling dengan cara memodifikasi driver tun/tap dengan target virtual device ini dikenal oleh Gnu Radio yang bekerja di *user space*. Kegiatan penelitian telah mengimplementasikan virtual device tun/tap pada module ATN, di mana virtual device dapat muncul dan dikenal, namun masih diperlukan hal-hal sbb.:

- Pemahaman mekanisme pengiriman dan penerimaan datagram ATN melalui suatu virtual device.
- Penambahan syscall baru (seperti : write, read) pada module ATN agar bisa berkomunikasi dengan device, karena selama ini aktivitas penelitian ATN berfokus hanya pada pembuatan protokol jaringan saja.

- Kajian dan pemahaman yang mendalam untuk bisa berkomunikasi dengan device, karena konsep layer datalink pada module ATN berbeda dengan TCP/IP
- Kajian dan pemahaman yang mendalam untuk bisa mengintegrasikan module ATN ke dalam module GNU Radio.

Ucapan Terima Kasih

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

Daftar Pustaka

[1] Ettus Research, The USRP: www.ettus.com
[2] Gnu Radio: www.gnuradio.org
[3] ICAO Doc 9776 AN/970 Manual on VHF Digital Link (VDL) Mode 2.
[4] M. J. Erickson, "OKI's ATN Router", *ATN Seminar*, Chiang Mai, Thailand, 2011
[5] R.D. Grappel, "Internet Over the VDL-2 Subnetwork: the VDL-2/IP Aviation Datalink System, *NASA Project Report*, 2000.
[6] H. Fahmi, H. Faidah, T. Prastowo, and C. Lim, "Implementation of CLNP for the ATN with BSD Socket," International Conference on Telecommunications, 2008.
[7] D. Isyanuar, P. Dana, W. Yulianto, D. Irawan, H. Fahmi, H. Faidah, K. I. Eng, C. Lim, "Integration of Transport Layer to Connectionless Mode Network Service in Aeronautical Telecommunication Network," *International Conference on Information & Communication Technology and Systems* (*ICTS*), 2009.
[8] A. A. N. Ananda Kusuma, C. Sujana, I. Turyana, "Software-Defined Radio untuk Pengembangan Testbed VHF Data Link (VDL), in *Proc. Teknoin*, 2011
[9] PTIK-BPPT, *Program Manual Kegiatan Sistem Informasi untuk Transportasi Udara (CNS/ATM)*, 2010.
[10] RTCA document, *Signal-In-Space Minimum Aviation System Performance Standards (Masps) for Advanced VHF Digital Data Communications Including Compatibility with Digital Voice Techniques*, 2000.
[11] Universal TUN/TAP Driver: http://vtun.sourceforge.net/tun
[12] D, Wenliang "Virtual Private Network (VPN) Lab", Syaracuse University 2010.