IETF-110 Hackathon

IPWAVE Basic Protocols Project

March 1-5, 2021
Online
(Busan, Korea)

Champion: Jaehoon Paul Jeong
Computer Science & Engineering
Sungkyunkwan University (SKKU)
pauljeong@skku.edu

IP Wireless Access in Vehicular Environments (IPWAVE) Basic Protocols

Champion: Jaehoon (Paul) Jeong (SKKU)

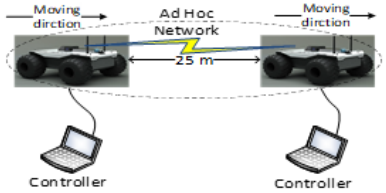
Professors:

- Jaehoon (Paul) Jeong (SKKU)
- Younghan Kim (SSU)

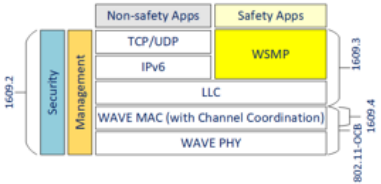
Students:

- Yiwen (Chris) Shen (SKKU)
- Bien Aime Mugabarigira (SKKU)
- Xiaohong (Dawn) Yu (SKKU)
- Kyoungjae Sun (SSU)
- Jinho Park (KNU)

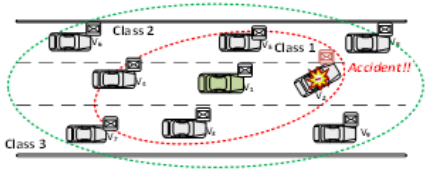
Environment Setup



WAVE Protocol Stack



IPv6 ND Option



CCM Cooperative Context Messages

ECM Emergency Context Message

Objectives:

- Demonstrate IPWAVE Basic Protocols
- IPv6 packet transmission by two aion robots
- Discover technology gaps for IPWAVE

Where to get source code:

- Git-hub open source: https://github.com/ipwave-hackathon-ietf

How to set up an environment:

Hardware

- Two laptops
- Two aion robots R1 UGV

Software

- OS: Ubuntu 18.04
- ROS: melodic

Implementation Contents:

- Develop a Vehicular Communication System for safe and secure driving using IETF IPWAVE protocols
- Transmission of IPv6 over IEEE 802.11-OCB
- Vehicular Mobility Information (VMI) option in IP-based vehicular network
- IPv6 packet transmission by two OCB-enabled WiFi modules
  - CCM and ECM Packets transmission by Python script

성균관대학교 SUNGKYUNKWAN UNIVERSITY

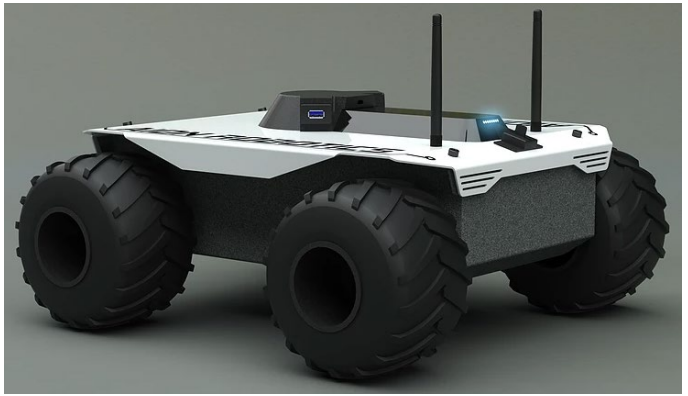
Previous Hackathon Work

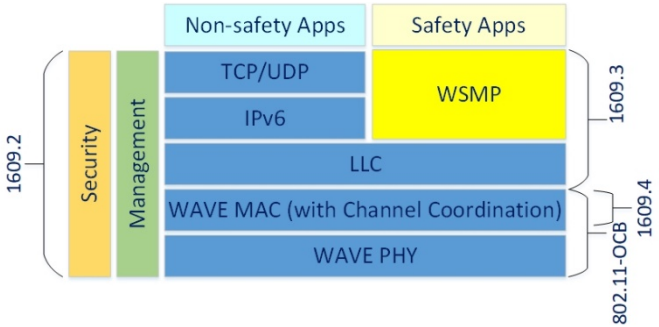
- IETF-106 Hackathon Project
  - IPv6 Packet Transmission over two OCB-enabled WiFi modules in vehicular networks
- IETF-108 Hackathon Project
  - Simulation of Context-Aware Navigation Protocol for road crash avoidance
- IETF-109 Hackathon Project
  - IP-Based Context-Aware Navigator

Hackathon Plan

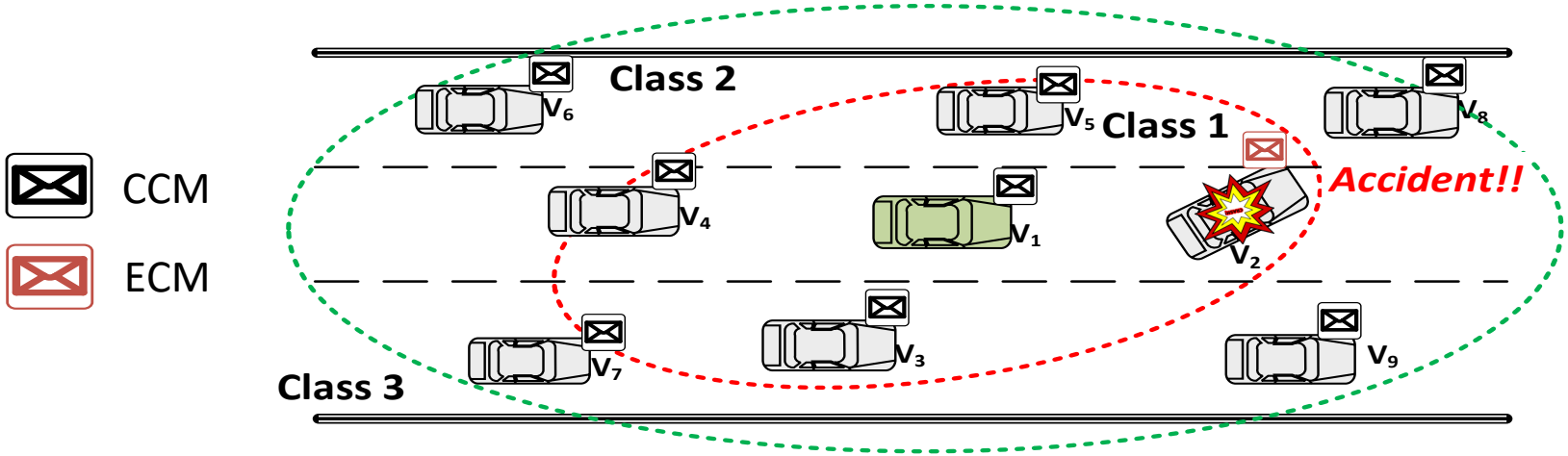
- A new ND option implementation in the Linux Kernel.
- Test of the modified Linux Kernel for a robot car for crash avoidance.
- Related Internet Draft:
  - draft-jeong-ipwave-context-aware-navigator-03

R1 of Aion Robotics





Context-Aware Navigation Protocol over IPWAVE



❖ **Road-Context Awareness through Light-weight Message Exchange**

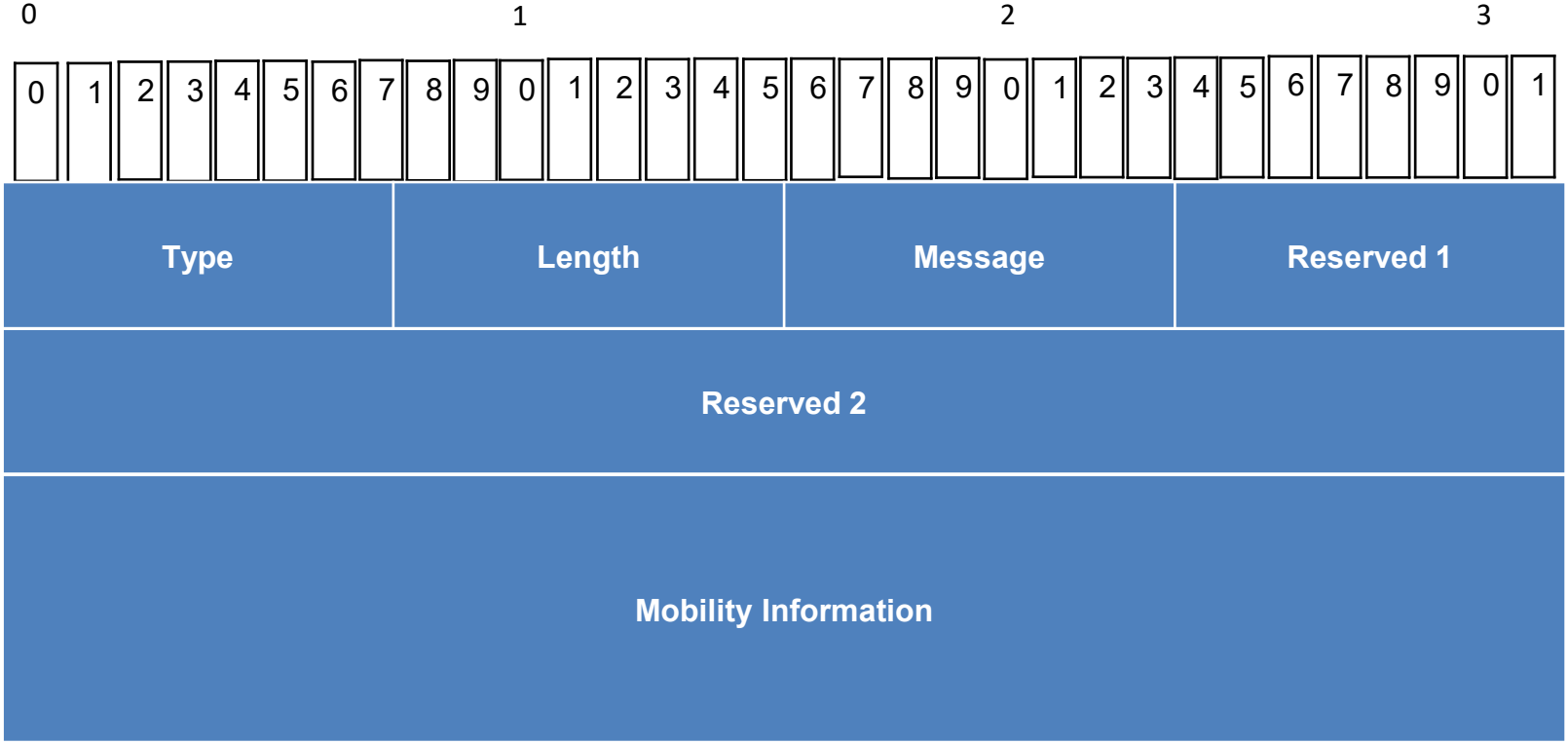
- **Cooperation Context Message (CCM) and Emergency Context Message (ECM) Options**

Reference 1: "Context-Aware Navigation Protocol for IP-Based Vehicular Networks", https://tools.ietf.org/html/draft-jeong-ipwave-context-aware-navigator-03

Reference 2: "Context-Aware Navigator for Road Safety in Vehicular Cyber-Physical Systems", ICCE-Asia, June 2018. http://iotlab.skku.edu/publications/international-conference/ICCE-ASIA-CAN.pdf

Vehicle Mobility Information (VMI)

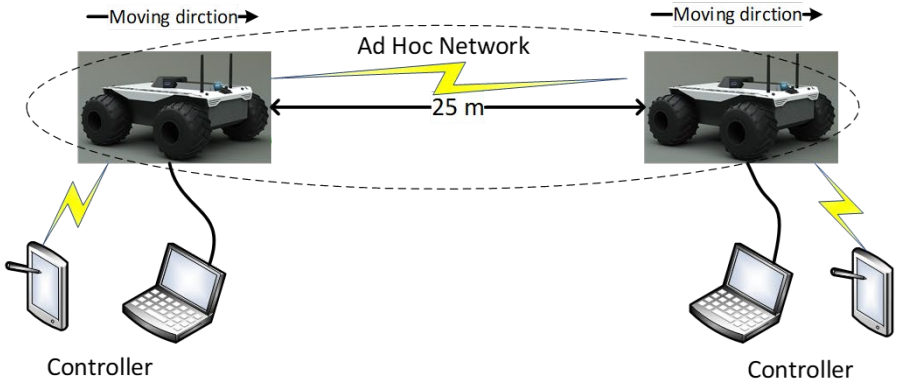
VMI Option as an ND Option:



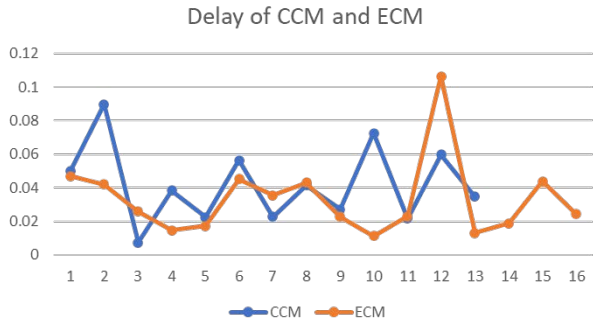
Type: CCM or ECM Options

What got done (1/2)

Feasibility by UDP-based Testing

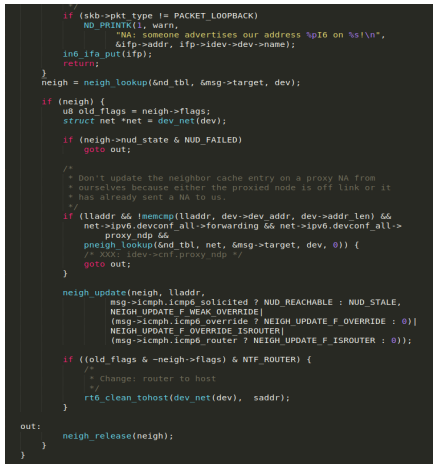


Msg Type (VMI Option)	Delay
CCM	0.0419s
ECM	0.0333s

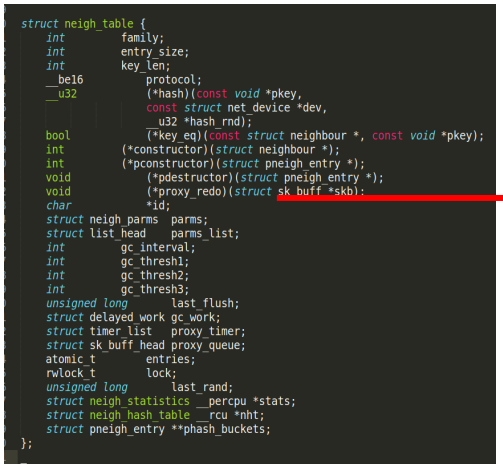


What got done (2/2)

- Files of Ubuntu Linux for ND: ndisc.c, .h, neighbour.h, icmp.c, etc.
- Restructuring of Neighbor Table and Neighbor Structure
- Redesigning of Neighbor Cache for VANET (Vehicular Ad Hoc Networks)



```
        if (skb->pkt_type != PACKET_LOOPBACK)
            ND_PRINTK(1, warn,
                  "NA: someone advertises our address %pI6 on %s!\n",
                  &ifp->addr, ifp->idev->dev->name);
        in6_ifa_put(ifp);
        return;
    }
    neigh = neigh_lookup(&nd_tbl, &msg->target, dev);

    if (neigh) {
        u8 old_flags = neigh->flags;
        struct net *net = dev_net(dev);

        if (neigh->nud_state & NUD_FAILED)
            goto out;

        /*
         * Don't update the neighbor cache entry on a proxy NA from
         * ourselves because either the proxied node is off link or it
         * has already sent a NA to us.
         */
        if (lladdr && !memcmp(lladdr, dev->dev_addr, dev->addr_len) &&
            net->ipv6.devconf_all->forwarding && net->ipv6.devconf_all->
                proxy_ndp &&
            pneigh_lookup(&nd_tbl, net, &msg->target, dev, 0)) {
            /* XXX: idev->cnf.proxy_ndp */
            goto out;
        }

        neigh_update(neigh, lladdr,
                 msg->icmph.icmp6_solicited ? NUD_REACHABLE : NUD_STALE,
                 NEIGH_UPDATE_F_WEAK_OVERRIDE|
                 (msg->icmph.icmp6_override ? NEIGH_UPDATE_F_OVERRIDE : 0)|
                 NEIGH_UPDATE_F_OVERRIDE_ISROUTER|
                 (msg->icmph.icmp6_router ? NEIGH_UPDATE_F_ISROUTER : 0));

        if ((old_flags & ~neigh->flags) & NTF_ROUTER) {
            /*
             * Change: router to host
             */
            rt6_clean_tohost(dev_net(dev),  saddr);
        }

out:
        neigh_release(neigh);
    }
}
```



```
struct neigh_table {
    int         family;
    int         entry_size;
    int         key_len;
    __be16          protocol;
    __u32           (*hash)(const void *pkey,
                const struct net_device *dev,
                __u32 *hash_rnd);
    bool            (*key_eq)(const struct neighbour *, const void *pkey);
    int         (*constructor)(struct neighbour *);
    int         (*pconstructor)(struct pneigh_entry *);
    void            (*pdestructor)(struct pneigh_entry *);
    void            (*proxy_redo)(struct sk_buff *skb);
    char            *id;
    struct neigh_parms  parms;
    struct list_head    parms_list;
    int         gc_interval;
    int         gc_thresh1;
    int         gc_thresh2;
    int         gc_thresh3;
    unsigned long       last_flush;
    struct delayed_work gc_work;
    struct timer_list   proxy_timer;
    struct sk_buff_head proxy_queue;
    atomic_t        entries;
    rwlock_t        lock;
    unsigned long       last_rand;
    struct neigh_statistics __percpu *stats;
    struct neigh_hash_table __rcu *nht;
    struct pneigh_entry **phash_buckets;
};
```

```
struct neighbour {
    struct neighbour __rcu  *next;
    struct neigh_table  *tbl;
    struct neigh_parms  *parms;
    unsigned long       confirmed;
    unsigned long       updated;
    rwlock_t        lock;
    atomic_t        refcnt;
    struct sk_buff_head arp_queue;
    unsigned int        arp_queue_len_bytes;
    struct timer_list   timer;
    unsigned long       used;
    atomic_t        probes;
    __u8            flags;
    __u8            nud_state;
    __u8            type;
    __u8            dead;
    seqlock_t       ha_lock;
    unsigned char       ha[ALIGN(MAX_ADDR_LEN, sizeof(unsigned long))];
    struct hh_cache     hh;
    int         (*output)(struct neighbour *, struct sk_buff *);
    const struct neigh_ops  *ops;
    struct rcu_head     rcu;
    struct net_device   *dev;
    u8          primary_key[0];
};
```

- Extending the ND option by adding the VMI option to ND
- Debugging the kernel to support the delivery of VMI option (CCM and ECM)

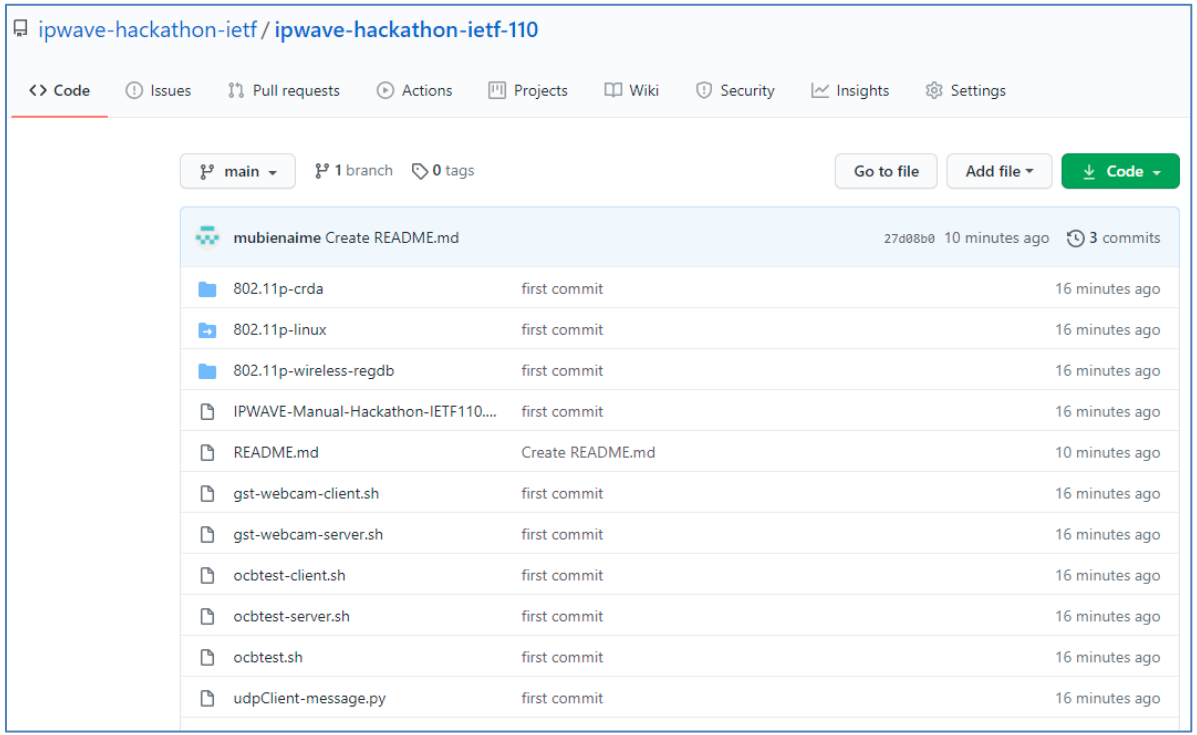
Next Step

- Continue developing and debugging the new ND option (called VMI) in the Linux kernel for vehicular environments for IETF-110 Hackathon.

- Context-Aware Navigation Protocol (CNP) over IPWAVE
  - Testing of the IPv6 ND to support VMI option for driving safety on robot cars.
  - Autopilot of robot cars using CNP message exchange.

IPWAVE Open-Source Project at Github

https://github.com/ipwave-hackathon-ietf/ipwave-hackathon-ietf-110



Wrap Up

IPWAVE Hackathon Team

Champion:

- Jaehoon Paul Jeong (SKKU)

Professor:

- Younghan Kim (SSU)

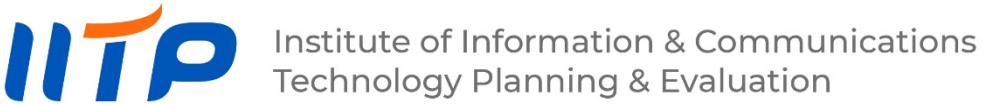
Students:

- Bien Aime Mugabarigira (SKKU)
- Yiwen Chris Shen (SKKU)
- Xiaohong (Dawn) Yu (SKKU)
- Kyoungjae Sun (SSU)
- Jinho Park (KNU)

IPWAVE hackathon team worked in collaboration with I2NSF and BMWG teams.

Sponsors


NRF
National Research
Foundation of Korea


IITP
Institute of Information & Communications
Technology Planning & Evaluation


TTA
한국정보통신기술협회
Telecommunications Technology Association


SUNG KYUN KWAN UNIVERSITY
1398
성균관대학교


숭실대학교
Soongsil University


ETRI
한국전자통신연구원
Electronics and Telecommunications Research Institute


K-IETF
KOREA IETF FORUM