



Μετάβαση (porting) εφαρμογών στο IPv6

Κ. Στάμος

Μηχανικός Η/Υ και Πληροφορικής, Πανεπιστήμιο Πατρών

Μηχανικός Έρευνας και Ανάπτυξης, ΕΜ6/ΕΑΙΤΥ



Περιεχόμενα

- Αναφορά στις αλλαγές που επιφέρει το IPv6 στην ανάπτυξη δικτυακών εφαρμογών
- Αλλαγές στο socket API
- Ανάπτυξη εφαρμογών ανεξάρτητων του πρωτοκόλλου επιπέδου δικτύου
- Ενέργειες για τη μετάβαση των εφαρμογών στο IPv6
- Παραδείγματα – case study (OpenH323)



Interoperability μεταξύ IPv4 και IPv6 εκδόσεων σε dual stack μηχανήματα

	IPv4 server	IPv6 server
IPv4 client	Επικοινωνούν με IPv4	Επικοινωνούν με IPv4, ο server βλέπει την IPv4-mapped IPv6 διεύθυνση
IPv6 client	Μπορούν να επικοινωνήσουν εάν ο IPv6 client χρησιμοποιήσει μία IPv4-mapped IPv6 διεύθυνση	Επικοινωνούν με IPv6

Μετάβαση (porting) εφαρμογών στο IPv6



Βασικές διαφορές στις διευθύνσεις

- IPv4:
 - Μήκος 32 bits
 - Γραμμένες σε δεκαδικό σύστημα
 - Αποτελούνται από 4 αριθμούς μεταξύ 0 – 255
 - Δεν συντομεύονται
 - Η τελευταία IP του υποδικτύου είναι η broadcast
 - Loopback 127.0.0.1
- IPv6:
 - Μήκος 128 bits
 - Γραμμένες σε δεκαεξαδικό σύστημα
 - Συντομεύονται (π.χ. 3FFE:B00::1)
 - Δεν υπάρχουν broadcast διευθύνσεις
 - Loopback ::1
 - Διακρίνονται ανάλογα με scope (link-local, site-local, global).....

Μετάβαση (porting) εφαρμογών στο IPv6

Βασική διαδικασία δικτυακής επικοινωνίας



- Server:
 - socket
 - bind
 - listen
 - accept
 - read/write/recvfrom/sendto...
- Client
 - socket
 - connect
 - read/write/recvfrom/sendto...
- Ίδια διαδικασία και για τα 2 IP πρωτόκολλα



Αλλαγές στο socket API – IPv4

```
typedef uint32_t in_addr_t;
struct in_addr {
    in_addr_t s_addr;          /* IPv4 διεύθυνση */
};

struct sockaddr_in {
    sa_family_t sin_family;   /* AF_INET */
    in_port_t sin_port;       /* Port. */
    struct in_addr sin_addr;   /* Internet διεύθυνση */

    /* Pad για το μέγεθος του `struct sockaddr'. */
    unsigned char sin_zero[sizeof (struct sockaddr) -
                             sizeof (sa_family_t) -
                             sizeof (in_port_t) -
                             sizeof (struct in_addr)];
};
```



Αλλαγές στο socket API – IPv6

```
struct in6_addr {
    union {
        uint8_t u6_addr8[16];
        uint16_t u6_addr16[8];
        uint32_t u6_addr32[4];
    } in6_u;

    #define s6_addr          in6_u.u6_addr8
    #define s6_addr16      in6_u.u6_addr16
    #define s6_addr32      in6_u.u6_addr32
};

struct sockaddr_in6 {
    sa_family_t sin6_family; /* AF_INET6 */
    in_port_t sin6_port;     /* Transport layer port # */
    uint32_t sin6_flowinfo; /* IPv6 πληροφορίες ροής */
    struct in6_addr sin6_addr; /* IPv6 διεύθυνση */
    uint32_t sin6_scope_id; /* IPv6 scope-id */
};
```

Μετάβαση (porting) εφαρμογών στο IPv6



Δομή ανεξάρτητη IP πρωτοκόλλου

/ Αρκετά μεγάλη δομή για να αποθηκεύει οποιονδήποτε τύπο διεύθυνσης (με την ιστορική εξαίρεση του AF_UNIX). Δεσμεύονται 128 bytes */*

```
#if ULONG_MAX > 0xffffffff
# define __ss_aligntype __uint64_t
#else
# define __ss_aligntype __uint32_t
#endif
#define _SS_SIZE      128
#define _SS_PADSIZE  (_SS_SIZE - (2 * sizeof(__ss_aligntype)))

struct sockaddr_storage
{
    sa_family_t ss_family;    /* Οικογένεια πρωτοκόλλων */
    __ss_aligntype __ss_align; /* Επιβολή της επιθυμητής στοίχισης */
    char __ss_padding[_SS_PADSIZE];
};
```



Αλλαγές στον κώδικα

— Δημιουργία socket

```
socket(PF_INET6, SOCK_STREAM, 0); /* TCP socket */  
socket(PF_INET6, SOCK_DGRAM, 0); /* UDP socket */
```

— Πέρασμα socket στον kernel

```
struct sockaddr_in6 addr;  
socklen_t  addr_len = sizeof(addr);  
/* γέμισμα της δομής addr χρησιμοποιώντας μία IPv6 διεύθυνση  
πριν την κλήση της συνάρτησης */  
bind(sockfd, (struct sockaddr *)&addr, addr_len);
```

— Πέρασμα socket από τον kernel στην εφαρμογή

```
struct sockaddr_in6 addr;  
socklen_t  addr_len = sizeof(addr);  
accept(sockfd, (struct sockaddr *)&addr, &addr_len);  
/* η δομή addr περιέχει μία IPv6 διεύθυνση */
```



Συναρτήσεις μετατροπής διευθύνσεων

IPv4

```
/* Από μορφή κειμένου σε δυαδική αναπαράσταση της IPv4 διεύθυνσης */  
int      inet_aton (const char *cp, struct in_addr *inp);  
in_addr_t inet_addr( const char *cp);
```

```
/* Από δυαδική μορφή σε μορφή κειμένου της IPv4 διεύθυνσης */  
char     *inet_ntoa(struct in_addr in);
```

IPv6

```
/* Από μορφή κειμένου σε δυαδική αναπαράσταση της IPv4/IPv6 διεύθυνσης */  
int inet_pton(int family, const char *src, void *dst);
```

```
/* Από δυαδική μορφή σε μορφή κειμένου της IPv4/IPv6 διεύθυνσης */  
const char *inet_ntop(int family, const void *src, char *dst, size_t cnt);
```



Χρήση υπηρεσίας DNS

```
struct addrinfo {  
    int    ai_flags;        /* AI_PASSIVE, AI_CANONNAME */  
    int    ai_family;      /* AF_UNSPEC, AF_INET, AF_INET6 */  
    int    ai_socktype;    /* SOCK_STREAM, SOCK_DGRAM ... */  
    int    ai_protocol;    /* IPPROTO_IP, IPPROTO_IPV6 */  
    size_t ai_addrlen;     /* μήκος του ai_addr */  
    struct sockaddr ai_addr; /* δομή socket για τη διεύθυνση */  
    char    ai_canonname;   /* canonical name */  
    struct addrinfo ai_next; /* η επόμενη addrinfo δομή */  
};
```

```
int getaddrinfo(const char *node, const char *service,  
               const struct addrinfo *hints,  
               struct addrinfo **res);
```

```
void freeaddrinfo(struct addrinfo *res);
```



Προς τα πίσω συμβατότητα με IPv4

- Δύο εκδόσεις του πηγαίου κώδικα, δύο εκτελέσιμα (συμβατό μόνο με IPv4 εκτελέσιμο και συμβατό μόνο με IPv6 εκτελέσιμο)
- Μία έκδοση του πηγαίου κώδικα, δύο εκτελέσιμα (συμβατό μόνο με IPv4 εκτελέσιμο και συμβατό μόνο με IPv6 εκτελέσιμο)
- Μία έκδοση του πηγαίου κώδικα, ένα εκτελέσιμο (ανεξάρτητο από την έκδοση του πρωτοκόλλου δικτύου)



Μεθοδολογία

1. Μελέτη – εξοικείωση με κώδικα
2. Διάτρεξη του κώδικα με αυτοματοποιημένο εργαλείο
3. Εκτέλεση προφανών τροποποιήσεων
4. Εκτέλεση μη τετριμμένων τροποποιήσεων
5. Έλεγχος – debugging κώδικα και δοκιμές
6. Ολοκλήρωση μετατροπής



Σημαντικότερες μετατροπές

- Αλλαγή δομών δεδομένων `sockaddr_in`
- Αλλαγή σταθερών όπως `INADDR_ANY`
- Αλλαγή συναρτήσεων, π.χ. `inet_ntoa`
- Εξάλειψη `hard-coded` διευθύνσεων
- Αντικατάσταση ή διαγραφή `IPv4-specific` επιλογών, π.χ. `TOS`
- `RFC 2732` (διευθύνσεις στα `URL` με αγκύλες `[]`)



Αυτοματοποιημένα εργαλεία

- Διαθέσιμα για πολλές πλατφόρμες:
 - Checkv4 for Windows (Microsoft)
 - Socket Scrubber for Solaris (Sun)
 - IPv6 Porting Assistant for Tru64 Unix (Compaq)
- Κατάλληλα για κώδικα γραμμένο στις γλώσσες C/C++, ενώ για άλλες γλώσσες υπάρχει ακόμα μία έλλειψη σχετικών εργαλείων
- Χρήσιμα για project με μεγάλη βάση source code
- Στις περισσότερες μη τετριμμένες περιπτώσεις πάντως η χρήση μόνο του εργαλείου δεν είναι αρκετή και ο προγραμματιστής πρέπει να επέμβει



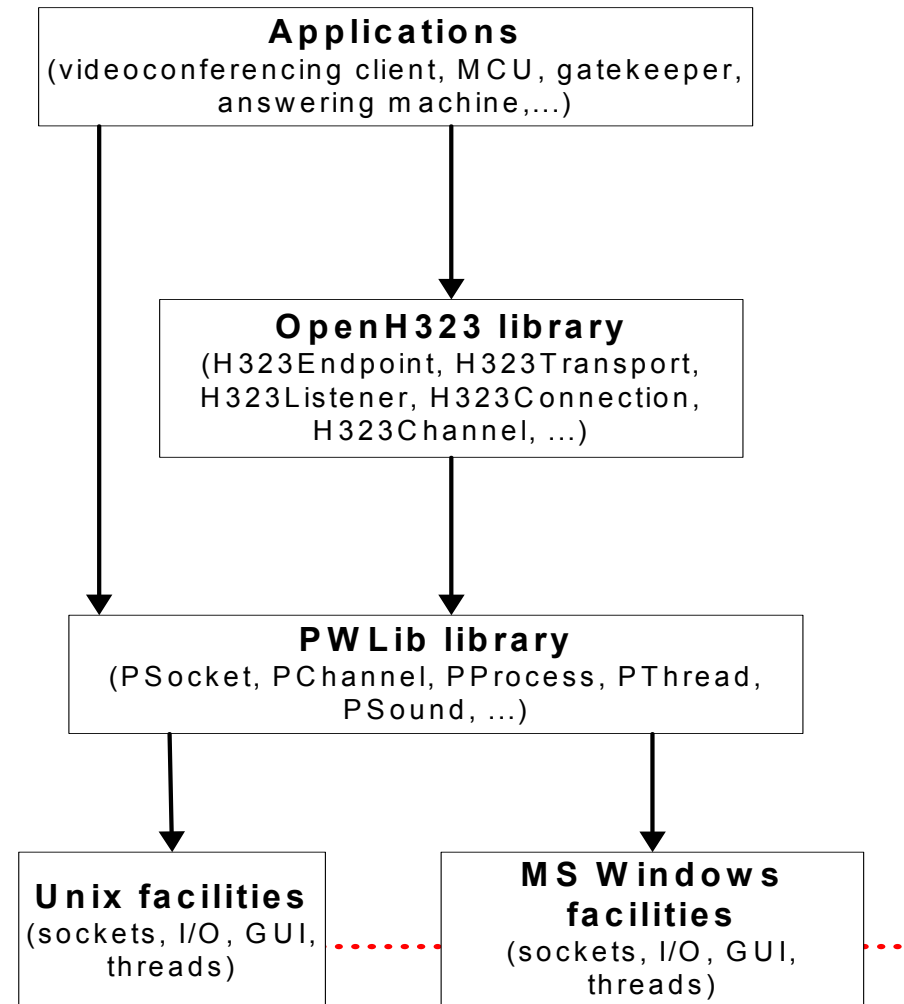
Case Study – OpenH323

- Open-source project με σκοπό την παροχή μιας βιβλιοθήκης για την υλοποίηση H.323 εφαρμογών
- OpenH323 library: παρέχει τη δυνατότητα για πολύ γρήγορη ανάπτυξη εφαρμογών που χρησιμοποιούν το H.323 πρωτόκολλο, καθώς αναλαμβάνει να υλοποιήσει τις χαμηλού επιπέδου λεπτομέρειες και αφήνει για τους προγραμματιστές των εφαρμογών τη λογική υψηλού επιπέδου.
- PWLib: open-source βιβλιοθήκη, ασχολείται με την παροχή των βασικών λειτουργικοτήτων που υποστηρίζει ένα λειτουργικό σύστημα (sockets, threads, I/O, GUI κ.λ.π.). Υποστηρίζει περισσότερα του ενός λειτουργικά συστήματα (Windows και Unix).



Δομή OpenH323

- Συνολικά περίπου 400 κλάσεις και ~1/2M γραμμές κώδικα σε C++
- Το porting έγινε στα πλαίσια του έργου 6NET



Μετάβαση (porting) εφαρμογών στο IPv6



Κυριότερα Προβλήματα

- Έμμεσες αναφορές σε IPv4 διευθύνσεις (π.χ. loops)
- Μη πλήρης συμβατότητα Linux-Windows IPv6 υλοποιήσεων
- Μη πλήρης υποστήριξη RFCs από Windows experimental stack
- Μέγεθος κώδικα



Πιστοποίηση ολοκλήρωσης

- High-level testing
- Low-level testing
- Comparative (Συγκριτικό ή back-to-back) testing



Συμπεράσματα

- Η εμπειρία μας έδειξε ότι η διαδικασία μετατροπής στο IPv6 θα είναι καθοριστική για την περαιτέρω αποδοχή του νέου πρωτοκόλλου. Ενώ για πολλές εφαρμογές αυτό θα είναι κάτι απλό, για εφαρμογές όπως το OpenH323 (μεγάλη βάση πηγαίου κώδικα με πολλές low-level λειτουργίες) θα χρειαστεί αυξημένη προσπάθεια.
- Η περαιτέρω ανάπτυξη και ωρίμανση αυτόματων εργαλείων που βοηθούν στην εργασία αυτή είναι απαραίτητη, και πιθανότατα θα πρέπει να εστιαστεί:
 - στην υποστήριξη περισσότερων γλωσσών πέρα από τις C/C++
 - στην βελτίωση του τρόπου λειτουργίας τους ώστε να μπορούν να βοηθήσουν τον προγραμματιστή και σε πιο λεπτά θέματα που ανακύπτουν κατά τη διάρκεια της μετατροπής του κώδικα μιας εφαρμογής.

Βιβλιογραφία

Δημοσιεύσεις (<http://ru6.cti.gr/bouras>)



- "The deployment of IPv6 in an IPv4 world and Transition Mechanisms", C. Bouras, P. Ganos, A. Karaliotas, Internet Research: Electronic Networking, Applications and Policy, Emerald, Volume 13, Number 2, 2003, pp. 86 – 93
- "Issues for the Performance Monitoring of an Open Source H.323 Implementation Ported to IPv6 – Enabled Networks with QoS Characteristics", C. Bouras, A. Gkamas, A. Karaliotas, D. Primpas, K. Stamos, The 2003 International Conference in Internet Computing, Las Vegas, Nevada, USA, June 23 - 26 2003, pp. 765 – 771
- "From IPv4 to IPv6: The Case of OpenH323 Library" , C. Bouras, A. Gkamas, K. Stamos, SAINT-2003, The 2003 International Symposium on Applications and the Internet, Workshop in IPv6 and Applications, Orlando, Florida, USA, January 27-31 2003, pp. 196-199
- "Transition Strategies from IPv4 to IPv6: The case of GRNET" , C. Bouras, P. Ganos, A. Karaliotas, 3rd International Network Conference-INC 2002, Plymouth, UK, July 16-18 2002, pp. 89-96

Βιβλιογραφία

Εργασίες – Παρουσιάσεις



- Από το IPv4 στο IPv6, Χρήστος Μπούρας, Κώστας Στάμος, http://ru6.cti.gr/ru6/ipv6/IPv6_Transition.doc
- Μετατροπή (porting) του OpenH323 στο IPv6, Χρήστος Μπούρας, Κώστας Στάμος, http://ru6.cti.gr/ru6/ipv6/OpenH323_porting.doc
- Μελέτη και προτάσεις βελτιστοποίησης απόδοσης των μηχανισμών που προτείνει το IPv6 για την εξάλειψη των περιορισμών του IPv4, Σιάχος Γιάννης, Διπλωματική Εργασία Μεταπτυχιακού Διπλώματος Ειδίκευσης, 2000, <http://ru6.cti.gr/bouras/lessons/ergasies.html>
- IPv6 και Εφαρμογές Πραγματικού Χρόνου, Στάμος Κωνσταντίνος, Διπλωματική Εργασία Μεταπτυχιακού Διπλώματος Ειδίκευσης, 2003, <http://ru6.cti.gr/bouras/lessons/ergasies.html>
- Εισαγωγή στην IPv6 τεχνολογία, Χρήστος Μπούρας, http://ru6.cti.gr/ru6/ipv6/Introduction_IPv6.ppt
- Μετάβαση (porting) εφαρμογών στο IPv6, Κώστας Στάμος, http://ru6.cti.gr/ru6/ipv6/IPv6_Porting.ppt

Ευχαριστώ πολύ



— Ερωτήσεις;

Μετάβαση (porting) εφαρμογών στο IPv6