**The Beacon OpenFlow Controller**

**Beacon** – מבוסס JAVA, נוצר ב-2010, מתפקד כמנהל OpenFlow בגישת open source.

נעשה בו שימוש רחב למטרת לימוד, מחקר ובסיס למוצרי תאורה Floodlight.

**המטרות של Beacon:**

1. **שיפור היצרנות (הפריון) של המפתח.**
2. **לספק את היכולת לפתוח אפליקציות חדשות ולעצור אפליקציות רצות, בזמן ריצה.**
3. **להיות בעל ביצועים טובים.**

**מנהל התקשורת הראשון בגישת open source שקדם ל-Beacon הוא NOX.  
NOX מאפשרת פיתוח בשפת Python שהיא נוחה וידידותית יותר עבור פיתוח אפליקציות רשת או בשפת C++ שהיא בעלת ביצועים טובים יותר.**

**NOX מציבה עבור המפתחים בה את ההפרדה בין נוחות לאיכות.**

**שאלות שנתעסק בהן:**

1. במנהל ה- OpenFlow NOX ישנה הפרדה בין כתיבת אפליקציות רשת בשפה ידידותית, נוחה וקלה כמו Python לבין כתיבה בשפה יעילה יותר מבחינת ביצועים (C, C++) – כלומר אפליקציות באיכות גבוהה.

**האם מנהל OpenFlow יכול לקבל את שניהם?**

**להיכתב בשפה ידידותית, נוחה וקלה מחד ולהיות בעל ביצועים טוב מאידך?**

1. **אם מנהל OpenFlow דומה במהותו למערכת הפעלת תקשורת (Network Operation System) האם הוא צריך להיות בעל יכולת לפתיחת וסגירת אפליקציות בזמן ריצה?**

שפת הפיתוח וסביבת הפיתוח חשובות מאוד ומהוות **חלק ניכר** מרמת היצרנות של המפתח ובאיכות המוצר שייצר.

ישנן המון שפות פיתוח ידידותיות אך טיב הביצוע שלהן בכל מה קשור לניהול OpenFlow לא היה ידוע בתקופת NOX.

**NOX – קצת רקע**

פלטפורמת מנהל ה-OpenFlow הראשון של NOX יצא ב-2008 עבד בשיטת ביצוע שיתופי בין תהליכונים.

**חסרונות של C ו- C++**

* זמן קומפילציה **מלאה** ארוך מאוד (יותר מ-10 דקות)
* שגיאות קומפילציה לא מובהקות, לא קל להבין מה יוצר את הבעיה.
* הקצאות ושחרור ידני של זכרון, יוצר קריסות עקב עומס על זכרון שמנוהל לא טוב או דליפות זכרון (זכרון שלא שוחרר כראוי ותופס ללא שימוש).